



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Research Commons

<http://researchcommons.waikato.ac.nz/>

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.

**CatchIt: Capturing Cues of Bookmarked Moment to
Feed Digital Parrot**

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
REQUIREMENT FOR THE DEGREE

Of

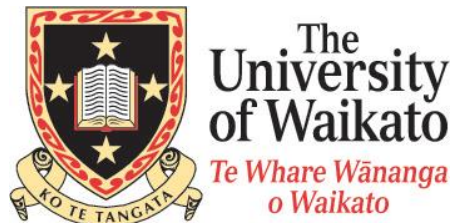
Master of Science in Computer Science

at

The University of Waikato

by

Hasan Jamal H. Alyamani



Department of Computer Science

Hamilton, New Zealand

July, 2012

© 2012 Hasan Alyamani

Abstract

CatchIt is a mobile application which aims to capture information about moments in a user's life in a semi-automatic way. The captured moments are then ready to be fed into an augmented memory system for later retrieval.

In *CatchIt*, the time and location contexts of the captured moment will be saved automatically from the system and GPS respectively once the user bookmarks the moment with the ability of modifying them later; whereas the desired other context, people, will be saved manually. The user of *CatchIt* also can capture further details (contents) of the moments by different ways: taking notes (textual information), taking photo, recording video, and recording audio. Additionally, the user can revise an earlier moment if the user wishes to. Later on, the user can transfer the selective bookmarked moments into the augmented memory system called the *Digital Parrot*.

The implementation of *CatchIt* is the central focus of this study. To do so, the requirements of *CatchIt* are specified based on results of a previous study and from a scenario. The conceptual architecture and the user interface of *CatchIt* are designed according to the *CatchIt* requirements. The user interface is implemented in addition to the database where is the captured information from the user interface will be stored in and retrieved from.

Evaluating the usability of *CatchIt* will come next. In the evaluation, the study will involve three sessions: (1) initial questionnaire to know more about the

preferred capturing ways of different scenarios, (2) testing the applications where an existing mobile application, *Hansel*, will be tested for the same length of testing CatchIt (one week for each) and (3) a guided interview which will extract the usability of CatchIt comparing with Hansel on one hand and adding a new contact to the phone using the application *Contacts*. An additional goal of the last session of the study will be to extract the feeling and the interests of using CatchIt.

The results of this study indicated the user interface of CatchIt needs to be even easier to use. The findings of the study form the foundation for further work to improve the user interface of CatchIt and to understand more of the user needs of such mobile capturing application.

Acknowledgments

First of all, all thanks, gratitude and praises are for Allah who gave me health, wellness and strength to complete this thesis and who illuminated my way to find the right ideas and solutions.

As Islam taught me that who does not thank people does not thank Allah, I give my thanks to some of people without whom this thesis could not have been completed. Firstly, I would like to thank my supervisor, Annika Hinze, for all her support, interests, encouragement, and precise advices throughout the writing of this thesis. Thank you Annika for helping me and assisting me to improve myself and to overcome the difficulties that I had.

Deep thankfulness and appreciation to my mother Elham Raboee and my father Jamal Alyamani for all their prayers to do well in my study and their patience for being away from them; may Allah give me anything to make both of you proud of me. Also, I wish to thank my sister Danya and my brother Baraa for their support and patience to make my dreams come true. I am so lucky to have such a nice sister and brother here in New Zealand and special regards are for the rest of my family for keeping asking about me and about my study.

Many thanks and gratitude for my friends Mumraiz Kasi and Mohammed Adeel for their help, advice and support. Also, special regards, to all my friends who did not give up encouraging me and praying for me to do my best and who ask about me and about my study. All your words affected on me and made motivation up.

Also my grateful thanks must go to everyone who participated in my study which helped me to carry out my study.

My thanks must also reach the King Abdullah Bin Abdulaziz to give me this great opportunity to study in New Zealand and continue my study which disappeared lots of the difficulties that I had.

Contents

Abstract	i
Acknowledgments.....	iii
Contents	v
1. Introduction	1
1.1 Information Overflows.....	1
1.2 Human Memory	1
1.3 Approaches and Applications to Support Human Memory	3
1.3.1 Approaches to Support Human Memory	3
1.3.2 Application to Support Human Memory	4
1.4 Capturing Cues.....	5
1.5 Focus and Structure of this Thesis	7
2. Background	8
2.1 Persona and Scenario	8
2.1.1 Persona	8
2.1.2 Scenario.....	9
2.2 Memory Types	11
2.2.1 Sensory Memory	12
2.2.2 Short-term Memory.....	13
2.2.3 Long-term Memory.....	16
2.3 Information Flow in Human Memory.....	17

2.4	Memory Cues	19
2.4.1	Types of Cues in the Scenario.....	20
2.4.2	Types of Capturing Cues.....	22
2.4.3	Ways of Capturing Cues	23
2.5	Summary	25
3.	<i>CatchIt</i> Requirements.....	28
3.1	Review the Scenario.....	28
3.1.1	Capture a Moment.....	28
3.1.2	Revise a Moment.....	29
3.1.3	Retrieve a Moment.....	31
3.2	Related Work	31
3.2.1	<i>Hansel</i>	32
3.2.2	<i>Contacts</i>	34
3.2.3	<i>Bookmarking a Moment</i>	35
3.3	Software Requirements	39
3.3.1	Functional requirements.....	39
3.3.2	Non-functional Requirements	44
3.4	Summary	49
4.	Conceptual System Design	54
4.1	Conceptual Architecture.....	54
4.1.1	Bookmark a Moment.....	56

4.1.2	Revise.....	57
4.1.3	Synchronize CatchIt with Digital Parrot.....	57
4.2	Overview Prototype Interface.....	58
4.3	Initial User Interface Prototype.....	61
4.3.1	Bookmark a Moment.....	62
4.3.2	Revise an Earlier.....	63
4.4	Expert Walk-through.....	89
4.4.1	Methodology.....	90
4.4.2	Result.....	91
4.5	The Final Prototype.....	107
4.6	Summary.....	111
5.	Implementation.....	113
5.1	Portable Device.....	114
5.2	Target User.....	115
5.3	Local Storage of CatchIt.....	116
5.3.1	Entity relationship diagram (ERD).....	116
5.3.2	Enhanced Entity Relationship Diagram (EERD).....	127
5.3.3	Relational Database Schema.....	132
5.4	The User Interface.....	134
5.4.1	Review the Scenario.....	134
5.4.2	CatchIt User Interface.....	135

5.5	Transferring the Data of the Bookmarked Moments	157
5.6	Summary	159
6.	Evaluating of the Usability of <i>CatchIt</i>	161
6.1	Goals of the Study	161
6.1.1	Method	162
6.1.2	The Applications of the Study.....	162
6.1.3	Participants.....	164
6.1.4	Procedure.....	164
6.1.5	Pilot Study.....	170
6.2	Results	170
6.2.1	Demographic information of the Participants	171
6.2.2	Preferred Ways of Capturing	171
6.2.3	Usability of CatchIt.....	176
6.2.4	Feelings and Suggestions	201
6.3	Discussion	208
6.3.1	Understanding the User Needs.....	209
6.3.2	A Leap Forward in Best Feeding	218
6.4	Summary	222
7.	Conclusion and Further Work.....	224
7.1	Summary	224
7.2	Further Work.....	226

Reference.....	230
Appendix A (Material for the Usability Study)	234
Appendix B Participants' Expressions.....	250
Participants Expressions of the Initial Questionnaire	250
The Participants Expressions of the Guided Interview	252
The Ease of Using CatchIt	252
The Ease of Using Hansel.....	254
CatchIt Vs Hansel	255
CatchIt Vs Contacts.....	257
CatchIt Feelings	258

List of Figures

Figure 1.1: Stages of <i>bookmarking a moment</i> approach. Reprinted from <i>Capturing Situational Context in an Augmented Memory System</i> (p. 43),by Alallah, J. Copyright 2011 by Jehan Alallah. Reprinted with permission.	7
Figure 2.1: Categories of human memory.....	11
Figure 2.2: Sensory memory captures the surrounding information by using the five senses.	12
Figure 2.3: Information transferred into short-term memory from sensory memory.	13
Figure 2.4: Functions and sub-functions of short-term memory.	15
Figure 2.5: Correlations between short-term memory and long-term memory. ...	16
Figure 2.6: Organizing structure of the stored information of long term memory.	17
Figure 3.1: Main window of Hansel. The user types the label of the place as an option and presses the <i>Drop Tag</i> button to drop a tag on the current location.	33
Figure 3.2: Contacts list in Contacts.	35
Figure 4.1: Conceptual architecture of CatchIt. The user interacts with the user interface to bookmark the target moment, revise it, and synchronize CatchIt with Digital Parrot to send the data of the bookmarked moment from CatchIt to Digital Parrot. All data of the target moments (contexts: location, time captured automatically and people captured manually; contents: such as the action and clothes of the user) will be stored in a local storage.....	55

Figure 4.2: Data flow diagram of CatchIt. The portable device feeds CatchIt with the time and location contexts in the first step while the user adds the people context and contents of the bookmarked moment with the ability of changing the time and location contexts in the second process. The third process transfers the stored contexts and contents of the bookmarked moments into another application, Digital Parrot. 56

Figure 4.3: Overview prototype of CatchIt. The key map is shown the upright of the figure. 59

Figure 4.4: The main view of CatchIt, *Page 1: The Main Menu*, at the initial running. Bookmarking a moment is the only main function is available. At the right side, *Message Box A* is an associated message box with *Page 1: The Main Menu*. 62

Figure 4.5: All buttons of *Page 1: The Main Menu* are available as long as there is at least one bookmarked moment..... 63

Figure 4.6: All bookmarked moment are listed in *Page 2: Bookmarked Moments*. Each item of the list has the time and the name of the bookmarked moment. 64

Figure 4.7: The values of the components of the bookmarked moment are changeable and can be deleted from *Page 3: Editing Moment*. The default values of time and location are the current time and location respectively and they are provided by CatchIt. The value of the rest of components is null. At the left side of the figure the message box *Message Box C* pops up in *Page 3: Editing Moment* when the user is going to delete a component of the bookmarked moment..... 65

Figure 4.8: *Page 3: Editing Moment's Name* before adding a moment's name. The *Save* button is not available until the user adds a moment's name. At the left side, *Message Box F* and *Message Box G* are the associated message boxes of *Page 4: Editing Moment's Name*..... 67

Figure 4.9: Saving a moment's name is available in *Page 4: Editing Moment's Name* if there is a new value of it..... 68

Figure 4.10: Editing the time of the bookmarked moment is the function of *Page 5: Editing Time*. The user can change the captured time and add a description of it. It is not allowed to save the data unless changes have happened to the data of the page. 69

Figure 4.11: Associated message boxes of *Page 5: Editing Time*..... 70

Figure 4.12: A valid inserted time in *Page 5: Editing Time* to the bookmarked moment..... 71

Figure 4.13: Editing location is available in *Page 6: Editing Location*. The associated message boxes are shown at the right side of the figure..... 72

Figure 4.14: *Page 6: Editing Location* after adding a description of the bookmarked moment..... 73

Figure 4.15: Editing text is available in *Page 7: Editing Text*. The associated message boxes are shown at the right side of the figure. 74

Figure 4.16: *Page 7: Editing Text* after adding textual notes to be attached to the bookmarked moment..... 75

Figure 4.17: *Page 8: Editing Photo* allows the user editing the photo of the bookmarked moment. The right side of the figure shows the message boxes that will pop up for the user according to their action on *Page 8: Editing Photo*. 76

Figure 4.18: <i>Page9: Capturing Photo</i> has an access to the built-in camera to capture a photo of the bookmarked moment.....	77
Figure 4.19: <i>Page8: Editing Photo</i> after adding a photo with a caption.....	78
Figure 4.20: Editing video is allowed in <i>Page10: Editing Video</i> . The right side of the figure shows both message boxes <i>Message Box F</i> and <i>Message Box G</i> that will pop up for the user according to their action toward <i>Page10: Editing Video</i>	79
Figure 4.21: <i>Page11: Capturing Video</i> has an access to the built-in camera to capture a piece of the bookmarked moment in a video format.	80
Figure 4.22: <i>Page10: Editing Video</i> after adding a video with a caption.	81
Figure 4.23: Editing an audio is allowed in <i>Page12: Editing Audio</i> . The right side of the figure shows the associated message boxes of <i>Page12: Editing Audio</i>	82
Figure 4.24: Recording audio is allowed in <i>Page13: Capturing Audio</i> and the audio is ready to be attached to the bookmarked moment.	83
Figure 4.25: <i>Page12: Editing Audio</i> after adding a recorded audio.	84
Figure 4.26: On the left side of the figure the initial view of <i>Page 14: Edit People</i> is shown. The check box control and the button <i>Delete</i> are not available as long as there is no name in the list. On the right side of the figure the associated message box <i>Message</i>	85
Figure 4.27: Contacts list of the user's phone is fetched to <i>Page 15: Contacts</i> . The user can select more than one name to be added to the bookmarked moment.	86
Figure 4.28: <i>Page 14: Editing People</i> after adding people's names to the people list.....	87

Figure 4.29: The appearance of *Page3: Editing Moment* after adding or modifying a collection of capturing types. 88

Figure 4.30: The moment's name is added to the first bookmarked moment in *Page2: Bookmarked Moments*. 88

Figure 4.31: *Message Box I* informs the user that the transferring data from CatchIt to Digital Parrot has been done. 89

Figure 4.32: Page 3-6: Location on Map is a new page of CatchIt. It is placed between Page 3: Editing Moment and Page 6: Editing Location. The aim of this page is to show the location of the bookmarked moment on a map. 92

Figure 4.33: New page, *Page 3-8: Photos* is placed between *Page 3: Editing Moment* and *Page 8: Editing Photo* with a slight change of the order of viewing pages. In this page all related photos of the bookmarked moment are listed in a table. Each cell has a small photo and its caption. If the user wishes to show the photo in a big size or edit it, they select the target photo then press *Edit*. The user can add more photos by pressing *Add* or delete the target photo by selecting it then press on *Delete*..... 93

Figure 4.34: *Page 3-10: Videos* is a new page which lists the captions of the all related videos of the bookmarked moment in a table. This new page is reached from *Page 3: Editing Moment* and can move the user to *Page 11: Capturing Video* if the user wishes to add video or can move to the user to *Page 10: Editing Video* if the user wishes to edit the selected photo. 95

Figure 4.35: Watching the recorded video is available in *Page 10.5: Play Video*. 96

Figure 4.36: The list of recorded audios with their captions is shown in *Page3-12: Audios*..... 97

Figure 4.37: All tagged people in all previous bookmarked moments are listed in <i>Page15: People</i>	98
Figure 4.38: A new person can be added from <i>Page 16: Person</i> so then this name will be shown in the list of people's name in <i>Page 15: People</i>	99
Figure 4.39: The new view of <i>Page 1: The Main Menu</i> . The brief information about the system is removed from the page.	100
Figure 4.40: <i>Page 10: Editing Video</i> after changing its contents. The remaining contents are the text box for inserting caption of the recorded video and play, save and back buttons.....	101
Figure 4.41: New text box to contain the caption of the attached text is added in <i>Page 7: Editing Text</i>	102
Figure 4.42: User can insert a caption for the recorded audio from <i>Page 13: Capturing Audio</i>	103
Figure 4.43: <i>Message Box J</i> pops up when the user wishes to delete a person from the list in <i>Page 15: People</i>	107
Figure 4.44: Adapted overview prototype according to the changes that have been completed after taking the recommendation of the expert walk-through into our account.	110
Figure 5.1: A diagrammatic view of a part of the conceptual architecture of CatchIt.	113
Figure 5.2: The first version of ERD of CatchIt.	117
Figure 5.3: The second version of ERD. One more entity, PEOPLE_IN_MOMENT, with two key attributes, <i>p_ID</i> and <i>m_ID</i> and two <i>one-to-many</i> relationships, <i>has</i> and <i>is_tagged</i> are placed after breaking the	

<i>many-to-many</i> relationship between the entities MOMENT and PEOPLE of the first version of the ERD.	123
Figure 5.4: The enhanced entity relationship diagram of CatchIt. The set of sub-classes {PEOPLE_IN_MOMENT, LOCATION, TIME} is a specialization of the super-class MOMENT while the set of sub-classes {NOTE, PHOTO, VIDEO, AUDIO} is a specialization of the super-class CAPTURE. The relation between PEOPLE and PEOPLE entities is a in a specific relationship type.....	127
Figure 5.5: The relational database schema.	133
Figure 5.6: (a) is the icon of CatchIt while (b) is the main window of CatchIt, which has the main functions of the applicaiton.....	136
Figure 5.7: (a) Message box pops up when the target moment is bookmarked. (b) The <i>Moment Details</i> window has a table of functions of the bookmarked moment.....	137
Figure 5.8: The <i>Moment Label</i> window holds the label of the bookmarked moment.....	138
Figure 5.9: In the <i>Moment Time</i> window the user can reset the time of the bookmarked moment and insert a description of that time.....	139
Figure 5.10: (a) The <i>Location</i> window shows the target location on the map while (b) shows the <i>Moment Location</i> which allows the user to change the location of the bookmarked moment by inserting the address of the target location and also this window also allows the use to insert a description of the target location.....	140
Figure 5.11: Taking notes of the bookmarked moment and inserting a label for the note are available in the <i>Note</i> window.	141

Figure 5.12: There is no photo in the *Photos* window as long as the user has not taken a single photo of the bookmarked moment. To add photo the user should press on the camera button at the bottom left corner of the window. 142

Figure 5.13: (a) The user takes a photo and then he can use it by pressing the *Use* button. (b) After using a photo, the user moves to the *Photo* window where the user can insert a caption of the photo. 143

Figure 5.14: *Photos* window after taking photos of the bookmarked moment. Each photo is shown in a small size with its caption. The caption *No label* means there is no associated caption for the photo. 144

Figure 5.15: (a) The first step of deleting a photo from The *Photo* window is to choose the target photo. (b) The second step is choosing a particular photo. In this step the deletion symbol rotates and the *Delete* button is made visible. 145

Figure 5.16: An empty video list in the *Videos* window because of there is no recording video of the bookmarked moment. 146

Figure 5.17: (a) The built-in camera works in a video mode. (b) The video recording stops if the recording time reaches one minute and a message box pops up to inform the user. 147

Figure 5.18: (a) The *Video* window when the user adds a caption to the recorded video. (b) The *Videos* window shows the list of captured video. 148

Figure 5.19: The audio list in the *Audios* window is empty because of there is no recorded audio of the bookmarked moment. 149

Figure 5.20: (a) The *Audio* window allows the user to record the target moment and it includes an instruction of the maximum length of the recorded audio.

(b) When the user stops recording, the user is allowed inserting a caption of the recorded audio.	150
Figure 5.21: The <i>Moment Details</i> window after attaching notes, photos, videos and audios to the bookmarked moment. The values of the fourth, fifth, sixth and seventh cells in the table have changed.	151
Figure 5.22: All bookmarked moments are listed in the <i>Moments</i> window including the data and time of the bookmared moment as well as the label of the bookmarked moment in the case of adding a label to that moemnt.	152
Figure 5.23: Playing the recorded video in CatchIt.	153
Figure 5.24: Both lists in (a) the <i>Tag People</i> window and (b) the <i>Select People</i> window are empty because there are no people tagged in the current moment, and there are no people tagged in any previous moments.	154
Figure 5.25: The <i>Person</i> window allows the user to insert a new person's name as a first step to tag this person in the bookmareked moment.	155
Figure 5.26: (a) Selected names in the <i>Select People</i> window will go to (b) the <i>Tag People</i> window.	156
Figure 5.27: The XML file has the required data of the bookmarked moments to be sent to Digital Parrot.	158
Figure 5.28: XML file (plist file) which includes the data of the bookmarked moments is attached with an email to be sent.	159
Figure 6.1: Age distribution of participants.	171
Figure 6.2: Frequency of the preferred capturing ways, to capture the events at friend's birthday party.	172
Figure 6.3: Frequency of the preferred capturing ways to capture the important notes in a lecture.	173

Figure 6.4: Preferred capturing ways to capture the shopping list.....	174
Figure 6.5: Frequency of the preferred capturing ways to capture the meeting with a supervisor/ advisor.	175
Figure 6.6: Frequency of the preferred ways to capture the information about a product.....	175
Figure 6.7: Frequency of the preferred capturing ways to record moments in participants' lives.	176
Figure 6.8: Ease of bookmarking a moment using CatchIt.....	178
Figure 6.9: Ease of changing the required data of CatchIt, time and location; 4 participates changed the values of the required data.....	179
Figure 6.10: Number of participants who added the optional data of CatchIt. ...	180
Figure 6.11: Ease of adding the optional data, label.	183
Figure 6.12: Ease of adding the optional data, notes.	183
Figure 6.13: Ease of adding the optional data, photos.	184
Figure 6.14: Ease of adding the optional data, video.	185
Figure 6.15: Ease of adding the optional data, audio.	185
Figure 6.16: Ease of adding the optional data, people.	186
Figure 6.17: Ease of moving to the previous window in CatchIt.....	187
Figure 6.18: Ease of tagging a place using Hansel.	188
Figure 6.19: Ease of adding the optional data of Hansel.	189
Figure 6.20: Ease of moving back to the previous window in Hansel.....	190
Figure 6.21: Frequency of choosing the preferred application (CatchIt or Contacts) to add the optional data.	199
Figure 6.22: Usefulness of supporting multi-languages to insert the textual information in CatchIt.	202

Figure 6.23: Frequency of the reasons behind the usefulness of supporting different languages to insert the textual information in CatchIt.	203
Figure 6.24: Frequency of the missed capturing ways in CatchIt (3 out of 12 participants asked for new capturing ways).	203
Figure 6.25: Frequency of choosing the capturing ways of CatchIt to be kept. .	204
Figure 6.26: Frequency of choosing the capturing ways of CatchIt to be left out.	205
Figure 6.27: Frequency of the participants' suggestions of CatchIt for each category.	206

List of Tables

Table 3-1: Comparison between <i>bookmarking web page</i> and <i>bookmarking a moment</i> approach. The symbol ‘●’ means that the feature is supported in the approach while ‘[●]’ means that the feature is supported by the use of a cooperative system.	37
Table 4-1: Heuristics of the walk-through that the experts will test the system by. The study involves pages, events and message boxes of the initial user interface by testing the area itself, content, controls, and title. The white cell means that the item of heuristics is covered because it was existing in the initial user interface prototype. Each of covered item will be identified inside the white cell. Otherwise, the item will not be covered in this study (the gray cells).	90
Table 5-1: The entities of the first version of ERD of the system.	118
Table 5-2: The attributes of the entities of the first version of the ERD of the system.	119
Table 5-3: The relationships between entities of the first version of the ERD of the system.	121
Table 5-4: The attributes of the new entity of the second version of ERD, PEOPLE_IN_MOMENT, of the system.	125
Table 5-5: All the relationships between the entities of the second version of the ERD of the system.	126
Table 6-1: Common aspects between CatchIt and Hansel and between CatchIt and Contacts. The common aspects with CatchIt are symbolized by ✓ whereas aspects that are not common are symbolized by ×.	163
Table 6-2: Questions of the initial questionnaire.	165

Table 6-3: Two groups of participants used each application, CatchIt and Hansel, for a week. Group A started using CatchIt then used Hansel whereas Group B started using Hansel then used CatchIt.....	166
Table 6-4: Five categories of questions in the guided interview.	167
Table 6-5: Number of participants who did not change the required data of CatchIt and the frequency of the reasons that prevented the participants to do so..	179
Table 6-6: Number of participants who did not add the optional data of CatchIt and the frequent of the reasons that prevented them to do so.	181
Table 6-7: Result of the compression of the ease of dealing with regarding the main function between CatchIt and Hansel. The result involves the Frequency of reasons that made participants prefer the application.	191
Table 6-8: The result of the compression of the ease of distinguishing the target between CatchIt and Hansel. The result involves the frequently of the reasons that made the participants preferred to choose the application.	193
Table 6-9: Result of the compression of the ease of adding the required data between CatchIt and Hansel. The result involves the frequency of the reasons that made the participants prefer the application.	194
Table 6-10: Result of the compression of the ease of dealing with regarding the main functions of CatchIt and Contacts. The result involves the frequency of the reasons that made the participants prefer the application.	196
Table 6-11: Result of the compression of the ease of distinguishing the target between CatchIt and Contacts. The result involves the frequency of the reasons that made the participants prefer the application.	197

Table 6-12: Result of the compression of the ease of adding the required data between CatchIt and Contacts. The result involves the frequency of the reasons that made the participants prefer the application.	198
Table 6-13: Proportion of reason of the ease of inserting the optional data of the applications CatchIt and Contacts.....	200
Table 6-14: Frequency of each missed feature in CatchIt.....	207

1. Introduction

An overview of the importance of supportive systems of human memory due to the considerable amount of surrounded information is discussed in this chapter. In addition, this chapter attempts to demonstrate how these systems might match the working-way of human memory. Also, this chapter presents one of the applications, which supports the human memory, and the complementary functions of that application to make it much similar to human memory. Finally, the central focus of this thesis will be emphasized.

1.1 Information Overflow

With massive amount of information that surrounds us, and with which we must deal, the ability of keeping all this information under the control of human brain is becoming increasingly difficult. As a result, the human memory needs more effort or aid to manage and cope with those data. Therefore, individuals in society as whole are likely to forget information, which they already have, and face difficulty in retrieving the information effortlessly (Lalanne & Hoven, 2007).

1.2 Human Memory

Human memory has sequence steps to deal with information in order to recall that information. It receives, stores, organizes and retrieves information. Because the aim of this process is to remember information after a period of time, psychologists classify the human memory in accordance with the time of storing the information. In fact, they tend to categories the human memory into three

inter-dependent categories: *sensory memory*, *short-term memory*, and *long-term memory* (Zhang, 2004).

Sensor memory is the memory that is responsible for receiving information. It senses the surrounding environment of the human by the five human senses and sends the information that the human paid attention to the short-term memory. Short-term memory then stores information temporarily and feeds the long-term memory by several methods such as repeating specific information (Atkinson & Shiffrin, 1971). After that, information, which is located in the long-term memory, will be organized, stored permanently and will be ready to be retrieved.

However, retrieving the stored information from the long-term memory is not that simple. Long-term memory needs assistance from the complemented memory which is short-term memory.

The short-term memory sends a piece of information, which is called *memory cue*, such as time or location of a particular stored event which is in long-term memory. Once cue triggers the long-term memory, the stored information will be retrieved from long-term memory to short-term memory. For instance, if a person had an accident in their childhood, it would not be difficult to remember that accident when they are within that location. So the location in the previous scenario is a cue to remembering a personal experience.

1.3 Approaches and Applications to Support Human Memory

Whilst increasing human needs to remember information, technology has also become more significant in improving our lives. In remembering matter, technology plays a great role to enhance and aid human memory so that individuals can remember with less effort. Many applications with different approaches, which aid the human memory, attempt to simulate the working-way of human memory. These applications manage to receive, store, organize and retrieve information (Lamming & Flynn, 1994).

1.3.1 Approaches to Support Human Memory

Personal information management (PIM) and continuous archival and retrieval of personal experiences (CARPE) are two common approaches that attempt to do the same functions as human memory.

Personal Information Management (PIM)

PIM is “the methods and procedures by which we handle, categorize and retrieve information on a day-to-day basis” (Lansdale, 1988, p. 55). This approach tends to copy all original information of the person into a digital format to facilitate the retrieval of information in the same format at the recalling time. Because this approach places extra emphasis on the organizing stage, many organizing methods are implemented such as grouping the files by activity, temporal organization, and organizing information semantically. However, copying all personal information to be retrieved does not necessarily represent the personal

experience in a direct way. An e-mail folder has a copy of all e-mails the individual received and sent. The search engine of that folder retrieves a particular group of e-mails. That does not mean those e-mails retrieve the personal experiences when the user received the emails although the search engine still manages to retrieve those emails according to the user-desires.

Continuous Archival and Retrieval of Personal Experiences (CARPE)

The CARPE approach, on the other hand aims to retrieve personal experiences by capturing almost the entire experience (Gemmell & Sundaram, 2006). It uses multimedia (video and audio record) to capture particular types of data (cues) of the whole user-moments, both interesting and not so interesting moments, automatically. In other words, CARPE systems represent the whole experience of the user or most of it. Nevertheless, that leads to capture huge amount of data, which needs enormous storage capacity. As a result, the retrieval stage becomes difficult even though some of CARPE systems tend to extract the interesting moments of the user and label them by using machine-learning techniques.

1.3.2 Application to Support Human Memory

In order to support human memory and augment it, some applications apply the previous approaches, PIM and CARPE. PIM approach supplies the application in the retrieval stage while CARPE approach supplies the application in capturing the cues of person's moments or experiences. One of these applications is the Digital Parrot.

Digital Parrot

Digital Parrot is an aid system for human memory. This system combines the approaches of CARPE and PIM. Digital Parrot captures the personal experiences (CAPRPE) then organizes the captured experiences semantically in order to retrieve these experiences (PIM) (Schweer, 2011).

In point of fact, this application combines context-awareness with semantic information. Firstly, the application understands an individual's situation when the individual receives new information firstly (context-awareness). Then, this system associates the new information with related stored information to organize it semantically (semantic information). Finally, in the retrieving stage, Digital Parrots shows the stored information in interactive visualization.

Nevertheless, Digital Parrot currently lacks a capturing method to feed itself with appropriate information (the function of sense memory) from temporary storage (the function of short-term memory). It needs to receive information from an external resource to complete the whole process of simulating the working way of human memory. Ways to feed Digital Parrot have been explored by Alallah (2011); the implementation of such a feeding system is the focus of this thesis.

1.4 Capturing Cues

Capturing information or cues is the first step of augmenting memory (Allalah & Hinze, 2011). Capturing cues works as the sensory memory does. It records moments of the user by several ways such as taking note and photos. Moreover, it feeds the next step, storage stage (both temporarily and permanently) in order to

retrieve the required information from the retrieval stage at a later time. As a result, the augment memory process will be completed. Despite the importance of the capturing stage, Digital Parrot lacks the ability to capture information (storing the information permanently and retrieving it later are the functions of Digital Parrot).

To support the sensory and short-term memory in Digital Parrot, the method of “marking moments” aims to bookmark the individuals’ selective moments as they do with bookmarking web pages in order to retrieve the bookmarked moments later by connecting with Digital Parrot (Alallah & Hinze, 2011). To implement this method, Alallah and Hinze (2011) did a user study to firstly explore the typical cues (time, location, and people) of the target moment so that retrieving of information later will be more efficient (Shiffrin, 1993). This step aims to imitate the sensory memory. In the second step, this study investigates what the appropriate capturing ways for the participants to capture the required moment (taking note, taking photo(s), video recording, audio recording) in addition to the best way of capturing them (automatically, manually, semi-automatically capturing). The second step aims to imitate the short-term memory. As the outcome of this study, Alallah and Hinze (2011) suggested some ideas to design a capturing moment system, which is called CatchIt (Figure 1.1, CatchIt will capture the interesting moment of the individual, then it transfers the information into Digital Parrot that retrieves the related information of the recorded moment). Nevertheless, there is no implementation has yet tested these ideas of CatchIt.

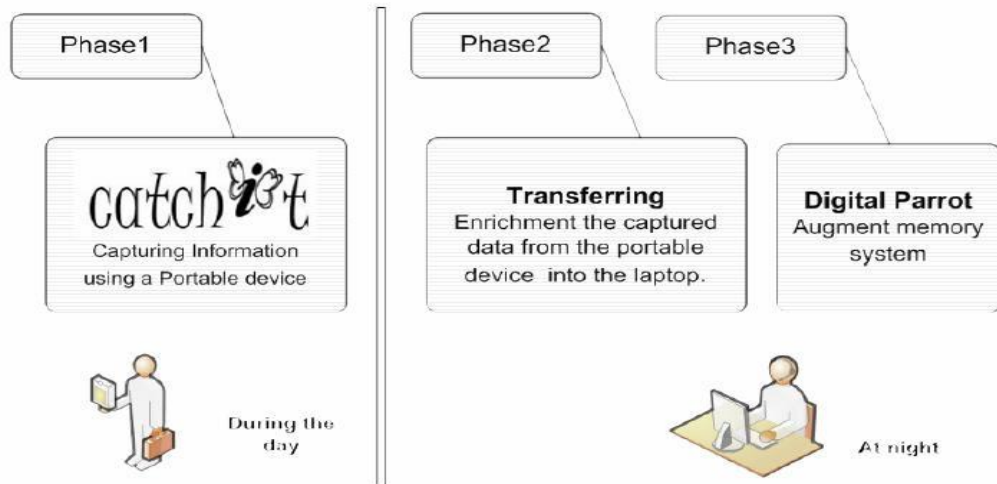


Figure 1.1: Stages of *bookmarking a moment* approach. Reprinted from *Capturing Situational Context in an Augmented Memory System* (p. 43), by Alallah, J. Copyright 2011 by Jehan Alallah. Reprinted with permission.

1.5 Focus and Structure of this Thesis

In this thesis, we will develop the design and implementation of CatchIt and then evaluate CatchIt with human participants and suggest improvements to the design. This thesis consists of seven chapters. The first chapter has introduced the information overflows in the human memory, the human memory and some approaches to support the human memory. In addition, capturing cues to support the information retrieval of the human memory and the focus and the structure of this thesis has been included in the first chapter. The rest chapters of the thesis are: Chapter 2 will inform the background of the thesis. Chapter 3 will specify the requirements to implement CatchIt. Chapter 4 will present the design of the conceptual architecture of CatchIt as well as the design of the user interface of CatchIt. The implementation of CatchIt will be explained in Chapter 5 while Chapter 6 will address the evaluation of the usability of CatchIt. At the end, the thesis will be concluded.

2. Background

This chapter introduces the area of this thesis, augmented memory. To simplify this area, we will introduce a scenario and we will refer to it as an example of each part of the area. The scenario will assist to illustrate the types of the human memory, information flow in human memory and the memory cues.

2.1 Persona and Scenario

Baraa is a main character in the scenario. He has great times with his friends and he wishes to capture these times to recall them. This scenario will be detailed in this section and it will be used to explain the other sections of this chapter.

2.1.1 Persona



Baraa is 24. He studies at Victoria University. He is a social person and he is outgoing. In addition, Baraa's life is very busy. If he does not spend his day studying, he joins his friends. Actually, Baraa has lots of friends from various countries and he goes with his friends to many places.

Because Baraa believes that he has to remember his friends' names (particularly the new ones and those who have 'difficult' names), and where, when, how he met them. Also, Baraa is interested in telling his friends about his previous experiences and about some other events he joined as well. Because of the busy life and the large number of friends he has, and the many interesting events he has joined, Baraa spends a lot of time and effort to remember his great moments, which sometimes includes new friends and new experiences.

Baraa intends to spend less time and effort, and wishes to establish an easy way to bookmark the interesting moments he had in order to review them when he wants.

2.1.2 Scenario

The scenario is divided into seven scenes. Each scene identifies uniquely to simplify referring the scenario.

Scene 1 (S1): Planning to Go

After attending the lecture, Baraa meets some of his friends at university. They are discussing options to fill the schedule for the coming weekends. Baraa makes the suggestion to go and watch the rugby match between All Blacks and South Africa in Westpac Stadium. They agree and he goes immediately to pay for the tickets.

Scene 2 (S2): New Experiences for Baraa

On the day of the match, 30th of July in 2011, after Baraa and his friends gather at university, they go to the stadium, *Westpac Stadium*. Baraa sees the All Blacks haka for the first time. He likes it because it reminds him of a traditional dance of his home country. He does not know what the All Blacks dance is called so he asked a person behind him, Jodi.

Scene 3 (S3): Baraa and Jodi

Baraa and Jodi have the first conversation between them. They talk a lot after they introduce themselves. In mean time, one of his friends offers hotdogs to Baraa and Jodi.

Scene 4 (S4): Unforgettable Events

Baraa has some funny events at that moment such as when Jodi thinks that he is from Brazil and one of his friend's glasses are broken. Also, Baraa receives a phone call from his mother. He cannot hear her very well so he tells her that he will call her back later. Indeed, Bara is excited by all that is happing to him right now.

Scene 5 (S5): Capturing the Moment

Because this is the first time for Baraa to have seen a rugby match, a haka, and to have met Jodi, he wants to remember this moment so he can tell his friends and family about it later. So he decides to use CatchIt to capture this collective moment.

Scene 6 (S6): Revise the Marked Moment

Later on that day, Baraa has nothing to do. He decides to manage the marked moment of this day. He reviews the marked moment that he marked today in the stadium. He turns his computer on and runs the desktop application of CatchIt. He inserts new information on the target moment as well as changes few stored information. He prefers doing that by using his computer rather than using his mobile because the size of the screen offers better reviewing on a bigger window.

Scene 7 (S7): Synchronize the Marked Moments between CatchIt and Digital Parrot

When Baraa comes back home, he transfers the marked moment from CatchIt to Digital Parrot. Thus, this moment will be saved for a long time. Baraa can also retrieve this moment after a long period of time.

2.2 Memory Types

Psychologists classify the human memory according to the period of time, for which the human memory keeps information into three types: sensory memory, short-term memory and long-term memory (Zhang, 2004) (see Figure 3.1).

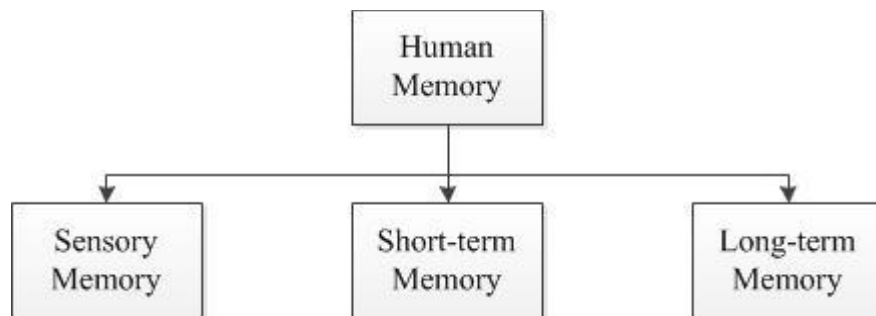


Figure 2.1: Categories of human memory.

2.2.1 Sensory Memory

As Zhang (2004) said the sensory memory is the shortest memory in terms of keeping information. In addition, it is responsible of capturing all surrounding information from the five senses (Figure 3.2). In our scenario: the sensory memory captures the cheering of fans (hearing), all surrounding people and objects such as the players and the rugby ball (sight), the smell of everything such as the aroma of food (smell), food taste (taste) such as hotdog, and all surrounding tangible objects such as seats (touch).

Moreover, sensory memory holds the captured information for milliseconds (Zhang, 2004) so that it is difficult to take advantage of the massive amount of captured information. However, sensory memory passes information to the short-term memory, where the information is stored temporarily, if we pay attention to what the sensory memory captured.

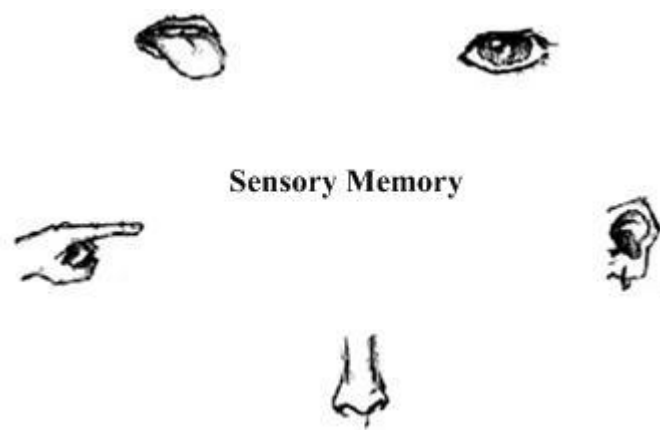


Figure 2.2: Sensory memory captures the surrounding information by using the five senses.

2.2.2 Short-term Memory

As it is clear from its name, short-term memory is used to store information temporarily and to retrieve it for the next few seconds (Kintsch, Patel & Ericsson, 1999) or minutes (Zhang, 2004). In other words, the information in short-term memory is able to decay (Kintsch, Patel & Ericsson, 1999). Short-term memory captures the information that the person paid attention to from the sensory memory (Figure 3.3). In our scenario, if Baraa paid attention to what happened to his friend's glasses (S4), this event is likely to be stored in the short-term memory for a while. So Baraa can remember that event for short time.



Figure 2.3: Information transferred into short-term memory from sensory memory.

So we can infer that the short-term memory is not useful if the information is required to be stored for long time and then to have it remembered. Nevertheless, short-term memory plays a big role to make saving the required information permanently possible. Short-term memory is an active memory and it also has multifunction. In addition to storing information on a temporary basis, it processes the stored information in different ways such as rehearsal and coding the information (Figure 3.4, these two processes of short-term memory, coding the information and rehearsal it are achieved by either maintenance or by elaborative rehearsal).

Rehearsal information is divided to two parts: maintenance rehearsal and elaborative rehearsal (DeWitt, 2007). Maintenance rehearsal is repeating the whole information, or a piece of it so as, to store the target information in the short-term memory (DeWitt, 2007). In some circumstances short-term memory manages to translate the repeated information to the long-term memory particularly if the repetition happens after long lags (Smith, Glenberg & Bjork, 1978). In contrast, elaborative rehearsal links the information in short-term memory with the previous experiences or knowledge and in this way is more efficient in remembering information after a long time since capturing it (DeWitt, 2007). In our scenario (S2): if Baraa repeats the body movements of haka many times, he might manage to keep those movements in his memory. This is an example of maintenance rehearsal, whereas associating those movements with the same movements of traditional dance that Baraa knows very well is an example of elaborative rehearsal. Another process that the short-term memory does in addition to rehearsing information is coding (Figure 3.4, coding the information is one of the sort-term information processes). It selects which way of rehearsing that the short-term memory uses to store information temporarily (Atkinson & Shiffrin, 1971).

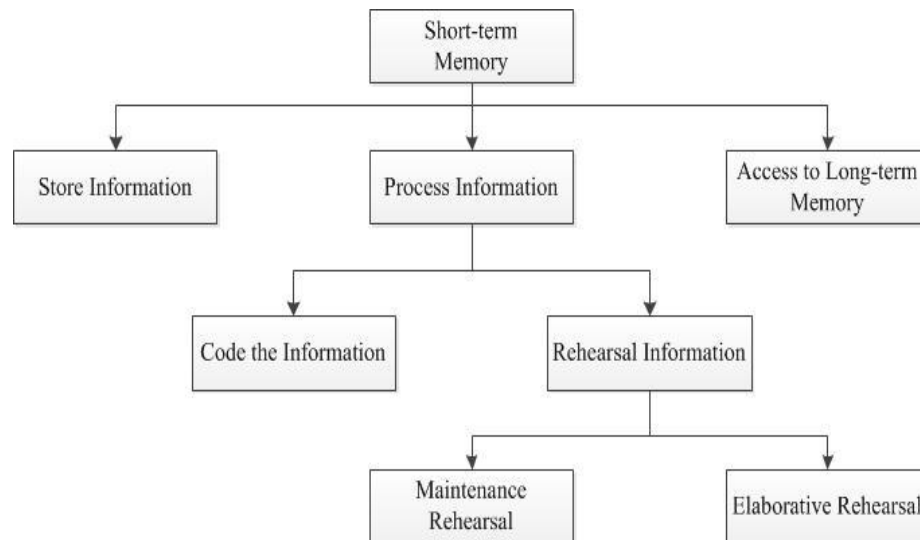


Figure 2.4: Functions and sub-functions of short-term memory.

Moreover, short-term memory works as a search tool for the stored information in long-term memory when short-term manages to get access to long-term memory (see Figure 3.4, short-term memory gets an access to long-term memory). This process enables the individual to remember a previous moment in the distant past in an indirect way (Atkinson & Shiffrin, 1971). After sending enormous amount of information to long-term memory to be stored there (the first step of Figure 3.5), short-term memory works as a search tool by using the memory cues as the query terms (the second step of Figure 3.5). When the search tool matches the cue with target information, long-term memory retrieves that information as a result of that search operation (the first step of Figure 3.5). In other words, the cue could be a small piece of the context of the target information that is used to collect the whole picture of that context (Jaimes, Omura, Nagamine & Kirata, 2004).

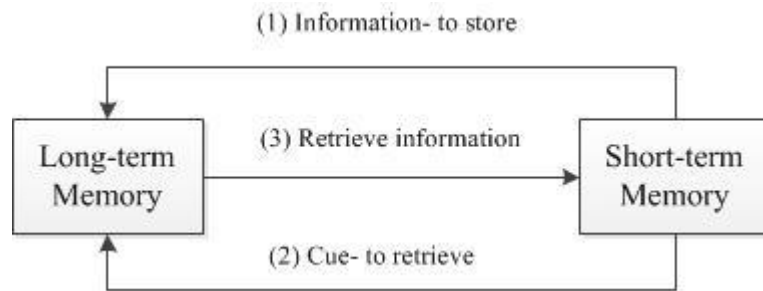


Figure 2.5: Correlations between short-term memory and long-term memory.

2.2.3 Long-term Memory

In contrast to the short-term memory, long-term memory is the place to store information for a long time or permanently (Kintsch, Patel & Ericsson, 1999). When long-term memory receives information from short-term memory, long-term memory organizes it according to a specific structure to enable it recall the information later. There are two organizing structures that information is formed into in the long-term memory: episodically and semantically (Figure 2.6).

Episodic way of organizing information (episodic memory) means that memory groups the new stored information with one of the past experiences; so an episodic memory has one context. In contrast, the semantic way (semantic memory) makes a relation between the new information with a global fact or knowledge, so semantic memory could have multi-contexts. To make the meaning clearer, we will refer to our scenario. As mentioned, for Baraa and Jodi (S3) the stadium was the first place they met. In the case of using episodic organizing information for organizing the new information, Baraa matches his new friend's name "with a name of one of his old friends". In a different way, in semantic

organizing information, Baraa matches the name “Jodi” with a famous author’s name, “Jodi Picoult” and an English actor “Jodi Albert”.

From the perspective of retrieving from long-term memory, after organizing, the stored information will be recalled.

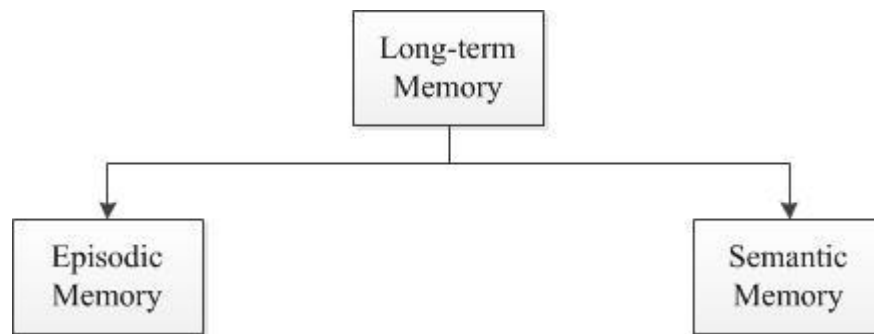


Figure 2.6: Organizing structure of the stored information of long term memory.

The stored information is triggered by cues, which come from the short-term memory – as it was explained previously.

In this thesis, we will focus on sensory memory that captures moments and short-term memory, which stores the captured moments in short time.

2.3 Information Flow in Human Memory

There are two types of information the human memory deals with: new information and previous information. New information is captured firstly by sensory memory. Secondly, by paying attention to that information the captured information is sent to short-term memory to be stored temporarily. The third step is transferring the new information from short-term memory to long-term memory after applying certain processes on it in order to store the new information

permanently. After storing the new information into long-term memory, it becomes one of the previous information or moment that individuals want to remember at a later point in time.

In order to retrieve the previous information, short-term memory accesses long-term memory by using memory cues (memory cues will be explained extensively in the next section) such as location cue. Memory cues triggers long-term memory and assists to retrieve it.

From our scenario, Baraa has precious moment of interesting events in the stadium. All surrounding information such as time, location, people and objects are captured by sensory memory. Information, which Baraa pays attention to, is placed in Baraa's short-term memory so that he can remember it for short time. The new moment of Baraa is translated to his long-term memory a after certain process from his brain, such as repeating some points of that moment. Later, the short-term memory of Baraa uses the location cue, Westpac Stadium, to trigger that moment, which is located in long-term memory. Now, Baraa remembers the haka and his friend's broken glasses.

In this thesis, we are interested in information flow between sensory and short-term memory on the one hand. On the other hand, we are partly interested in the relation between short-term and long-term memories (transferring information from short-term to long-term memory).

2.4 Memory Cues

Cue is a related context to the moment, which the individual had experienced. It works as a memory trigger to recall that moment or another related moment. Therefore, cue is similar to the query term of search operation. The search operation matches the cue with the related moments, which are located in long-term memory, to retrieve some details of the target moment. In our scenario, Baraa could use “Jodi”, his a new friend’s name, as a cue to remember some events happened in the stadium, his target moment. That indicates that using more than one cue increases the opportunity to reach more details of the target moment (Jaimes, et al., 2004) (Tan, Stefanucci, Proffitt, & Randy, 2001; Polyn, Natu, Cohen, Norman, 2005). In our scenario, Baraa uses more than one cue such as *Jodi* (people cue) and *Westpac Stadium* (location cue) to remember more information of his target moment.

Furthermore, the outcome of the search operation for the target moment may be another cue thereby the searching operation will use another cue to retrieve other details of the target moment (Jaimes, et al., 2004). As a result of retrieving another cue and so forth, the individual is very likely to remember more details of the target moment (Lee & Dey, 2007). In our scenario, Baraa could use *Westpac Stadium* as a location cue to retrieve another cue *Jodi*, which is a people cue, and then *hotdog*, which is an object cue. This way leads Baraa to get more details of his target moment.

2.4.1 Types of Cues in the Scenario

There are various types of cues for the moment in our scenario (related context to the moment): location, place, time, emotion, object, people, and phone call.

Location Cue

Location cue indicates the location context of the required moment. Location cue is one of the best types of cues in terms of retrieving the stored moment in particular, in the effort perspective (Tan, et al., 2001). The individual needs less effort when they use the location cue so as to remember their moments. In our scenario (S2), the location cue is Westpac Stadium.

Time Cue

Time cue is the time-context of the target moment. It is a powerful cue to retrieve the stored information efficiently (Lansdale, 1988). It has a unique value for each moment or for the information the individuals want to remember even if it is described semantically. In our scenario (S2), time cue is *30.July.2011 08:00 PM* (the first time Baraa saw a rugby match).

Emotion Cue

Emotion cue describes the emotional situation or event the individual had during the moment they want to remember. It is changeable particularly for a moody person. In addition, it has limited values such as laughing, happiness, which make getting unique emotion difficult. As a result, emotion cue does not help to retrieve the stored moments (Burt, Mitchell, Raggatt, Jones, & Cowan, 1995). However, this cue makes the retrieval operation easier in a situation of correlating the

moment with strong emotion (Lalanne, Hoven, 2007). In our scenario (S4), emotion cue is Baraa's excitement.

People Cue

People cue describes the people surrounding an individual during the target moments. It might present the peoples' names, faces, or professions. Nevertheless, not all people help retrieve the stored moments. People, who always surround the individual are not a good cue, that because they become familiar to the individual in contrast to those who rarely meet that person (Alallah & Hinze, 2011). In our scenario (S3), Jodi is the people cue.

Phone Call Cue

One of the most common activities that individuals partake in during the day is making phone calls. The phone call cue was used in some studies to investigate the use of different memory cues to retrieve certain information, such as the study that Jaimes, et al. undertook (2004). If we related the start and end time of the phone call with the location of the individuals, we would retrieve some information about the bookmarked moment. Example of retrieving information is if the individual was driving, walking or sitting in a fixed place. In our scenario (S4), call from Baraa's mother to him is a phone call cue.

Object Cue

Object cue has variety of attributes such as location, shape and color to present a specific object, which takes part in our moments. In our scenario (S3), object cue is hotdog.

In this thesis, we will focus on using location and time cues as the main cues. However, the user may apply some capturing ways to capture emotion, people, phone call, and object cues (see Section 2.4.3).

2.4.2 Types of Capturing Cues

There are two types of capturing cues that aid memory applications that may use to capture the users' experiences: total capture and situation-specific capture (Sellen and Whittaker, 2010).

Total Capture

Total capture method aims to capture the whole user's life including both desired and undesired moments. This type of capturing captures massive data so that it seeks enormous amount of storage space in addition to capturing trivial moments. For example, the total captures will capture all moments of our persona Baraa, including the not interesting ones, such as when he met up with his friend after the lecture (S1) and also the interesting moment, such as the moment in the stadium (S5).

Situation-specific Capture

In situation-specific capture method, the application tends to capture the interesting moments automatically, manually, or semi automatically. The automatic capture is recording the moment without any interaction from the user. Therefore, it records almost the whole interesting moment. For instance, this way of capturing captures automatically Baraa's interesting moment (S5), in the stadium, which actually means the total of that moment.

In contrast, in the manual capture, the user records any piece of information they want to remember and recall later. For instance, from our scenario, Baraa should record every single thing of his moment in the stadium. However, there are some sorts of information that are commonly forgotten to be recorded such as time and location although their importance is significant. In Allalah and Hinze (2011) 14 participants out of 30 recorded the time while no participant recorded location.

Semi-automatic capture method is a more efficient capturing way (Schweer & Hinze, 2007). It activates the role of user during recording the target moment by allowing them to annotate the captured context such as adding a caption to a taken photo. In addition, the user selects the interesting moment and the proper capturing ways to record the target moment. For instance, in our scenario (S5), Baraa can capture his interesting moment (in the stadium) by using a suitable way, such as taking a photo and typing a label for that photo. Whilst capturing Baraa's moment, the system also records the time and location automatically.

In this thesis, we will concentrate on the semi-automatic capture by both ways, annotating the contexts as well as selecting the target moment and capturing ways.

2.4.3 Ways of Capturing Cues

There are various ways of capturing cues or the contexts of the target moment. Taking note (textual format) and taking photo/s, recording video, and recording audio (media format) are examples of capturing ways. The user can use one capturing way to capture their moment as well as a combination of different capturing ways.

Taking Notes

Note-taking is recording the information that individuals want to remember in a textual form. In our scenario, for example, Baraa can write down the time and location of his target moment (S2). By using this capturing way, individuals think about suitable words that introduce the required information before writing it down. In this way, note taking improves the stage of information retrieval (DeWitt, 2007).

Taking Photo/s

Photos can describe the situational-context of the user with more details. It might contain the place, people and the colors of objects as well as the shapes of them. For example, in our scenario, Baraa can take a photo of the stadium (S2) and hotdog (S3) as well. Also, as some of the participants of Allalah and Hinze (2011) did, taking photo captures the textual information without the need for writing that information down on page.

Video Recording

Although using video recording to capture the target moment leads to the difficulty of finding out the particular needs of it, it ensures recording more details of the events that the target moment includes (Jaimes, et al., 2004). In our scenario (S2), Baraa can record the haka so that he can remember the details.

Audio Recording

Audio recording is a capturing way that has the ability to record people's sounds and objects' voices of surrounding environment. For instance, in our scenario,

Baraa can use the audio recording to record the conversation between himself and Jodi (S3), or record the words of the haka (S2). In addition, this capturing way works as a taking-note but in a different format. In our scenario, Baraa can record Jodi's name (S2).

In this thesis, we will cover the previous capturing ways (taking note, taking photo/s, and video recording). Additionally, it will be ability of combining two different capturing ways or more.

2.5 Summary

In this chapter, in order to make our work easier to follow and more understandable, we firstly introduced a persona and sequential scenes of a scenario as examples in the sections. Then we illustrated three types of memory: sensory, short-term and long-term memories as well as the functions of each one and the correlation among them. We are more interested in the relation between sensory and short-term memories (to capture the surrounded information and save it temporarily).

Because of the application we are going to implement aims to capture the users' moments (the same function of the sensory memory), this chapter emphasized the sensory memory, by explaining its function (capturing the surrounded information); what types of information (cues) we are interested in, how can we capture them, and which capturing ways are available to capture the requested cues (Figure 2.7, the interesting points of our thesis is shown by the orange boxes and lines. The left side of the figure shows the types of human memory as well as

the functions of each one of them whereas the right side of it extends the function of sensory memory, capturing information, with more details). Focusing on these areas will help us to find out the appropriate requirements of the system we want to implement. The requirement analysis is discussed in the next chapter.

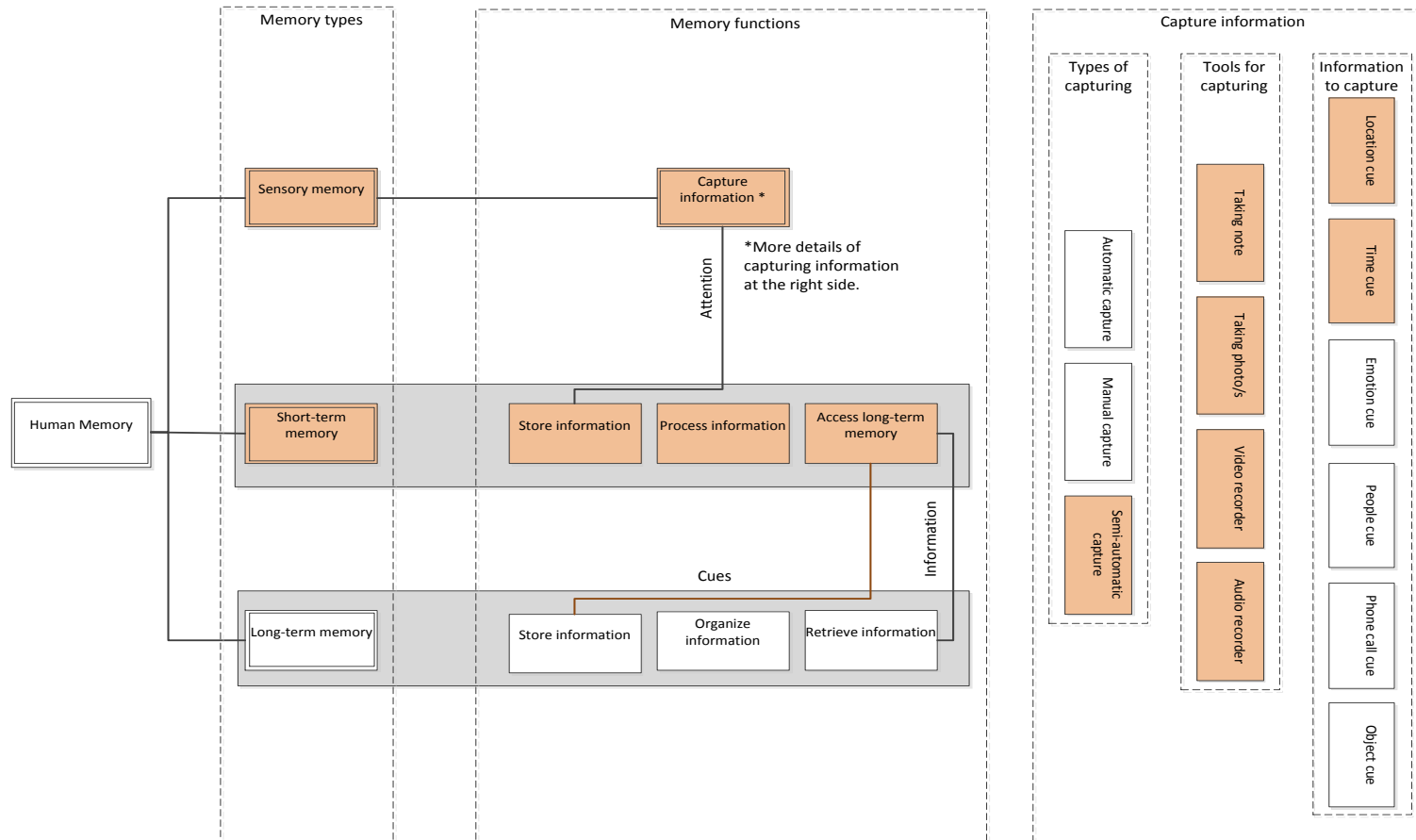


Figure 2.7: All highlighting areas of this thesis including types of memory with their functions in addition to capturing information.

3. *CatchIt* Requirements

The focal point of this chapter is deciding the requirements our system needs. The requirements of our software will be extracted from two resources: (1) the scenario and (2) the previous work on bookmarking a moment by Allalah (2011).

3.1 Review the Scenario

Baraa, who was introduced with a scenario in Section 2.1 will be used in this chapter to identify the requirements for *CatchIt*. Baraa, who does not lose his interest in collecting some of his moments to remember them later and to tell his friends about them, uses *CatchIt*. He bookmarks his interesting moments to retrieve them later when he requires them. We now walk through the scenario again to identify system features for each step.

3.1.1 Capture a Moment

Baraa is walking from the city center of Wellington to the Westpac Stadium with his friends. He has in his possession the wearable capturing device, which captures the moments. The device, being fairly small and light, Baraa does not feel that he is carrying an additional device.

Once Baraa enters the stadium and takes a seat, he feels that this moment is full of excitement. He intends to use the capturing device. Because this is the first time that Baraa is using the system, it means there has been no previous moment that was captured. Because the capturing option is the only option that is available, he cannot

use the system to revise an earlier captured moment, translate it to other applications, delete it, or edit any related information to the captured moment.

Baraa is very interested in capturing this moment so he captures it confidently; he captures the moment; it is not captured by the system. Baraa captures the moment himself because he knows that the captured moment will not be accessed by anyone else. Once he bookmarks the moment, the system then records the location after asking Baraa permission for using his current location and time of the moment. Baraa then labels the moment as *the stadium*.

3.1.2 Revise a Moment

In the stadium, Baraa later watches the haka of the All Blacks team for the first time. He wants to add this event into the bookmarked moment *the stadium*. He runs our system again, however on this occasion all options of our system, such as deleting and revising the previous moments are available to Baraa. He chooses revising bookmarked moment and selects *the stadium* moment from the chronological order list of recent bookmarked moments. Then he uses the video recorder which is supported by our system and then records the haka. In fact, even though the capturing device has a big capacity, it allows Baraa to record up to only one minute of the haka. The system places this limit so that less storage capacity is used.

In the stadium as well Baraa meets a new friend, Jodi, and has a conversation with her. Baraa asks Jodi if he may record part of it and if he could take a photograph of her; she agrees. Baraa uses our system, which supports recording audio and recording image. The system allows Baraa to record only a short part of their conversation so as

to save space in the storage capacity of the capturing device. Baraa also records Jodi in the image form- i.e., taken a photograph of her.

After the conversation, Baraa has some events that he wants to remember later: his friend's glasses broke and Baraa receives a phone call from his mother. Consequently, Baraa writes notes of these events in his own language, Arabic, and also in English.

Later, Baraa returns to home. He wants to review the captured moment that happened in the stadium and to edit it. He intends to caption the recorded video and audio and also the photograph that was taken because he believes by adding captions he would be able to remember the captured moment more easily. He labels them *the haka, first conversation with Jodi*", and "*Jodi*" respectively. In addition, Baraa wants to label the location and the time of the captured moment semantically. This method will help him remember the captured moment more easily compared to trying to remember the moment by the coordinates of the location and by the time/date format. Although our system represents the location of the captured moment in an address format, Baraa labels the location as *Westpac Stadium* and as *the first day I saw a rugby match* a time format of that moment. Each time Baraa edits the moment, he receives feedback from the system, which tells him that all changes are saved.

Mistakenly, Baraa changes the time of the moment by putting it forward. Nonetheless, because our system does not accept either modifying the time of moment by making it 'in the future' or by matching time of another previous moment, the system lets him know that this new date is unacceptable. Thus, Baraa is happy because the system

helps him save his moment, that is, saves changing one of the most important aspects of his moment, the time.

Baraa now is sure that he does not need to add new related information to the stadium moment. He synchronizes the capturing device with another application, Digital Parrot (explained in Chapter 1) which is located in his computer. He then translates the information from our system to Digital Parrot so he could retrieve it later. On completion of these processes, all captured moments are deleted from the capturing device.

3.1.3 Retrieve a Moment

Four months later, Baraa is planning to meet some of his friends. He intends to tell his friends about his interesting moment at the stadium. Hence, he runs Digital Parrot, searches for that moment, retrieves the pieces of information, and then remembers it.

3.2 Related Work

There are three related work to ours: *Hansel*, *Contacts* and *bookmarking a moment*. The first two are mobile applications. They will be used in the study to compare them with our application CatchIt to evaluate the usability of CatchIt. On the other hand, *bookmarking a moment* is the approach that the implementation of CatchIt will be based on.

3.2.1 *Hansel*

Hansel is a mobile application (Markus Luczak-Rösch, 2012). The aim of this application is to tag the target places to be revisited later. The application captures the time of the selective tagged place as well as the coordinates of that place automatically on behalf of the user. As an option, the user enables to add a label of the place before the user is going to tag the target place.

The usage of *Hansel* is very simple. The required action and the optional data, a label, are located in a same window, *Go and Tag* (see Figure 3.1). The user of *Hansel* just needs to press one button, *Drop Tag*, once the user reaches the target location. Subsequently, *Hansel* will capture the current time and the current location of the user. The users can tag as much places, as they want. In addition, at the end of whole windows of *Hansel* there is a tab bar, which has four buttons to move between the main windows of the application easily. For instance, simply, if the user wishes to see the list of all tagged place, the user presses the *Taglist* button, which is one of the tab bar buttons.



Figure 3.1: Main window of Hansel. The user types the label of the place as an option and presses the *Drop Tag* button to drop a tag on the current location.

However, all information of the tagged places is stored remotely in a server not in the device. As a result, Hansel needs the Internet connection to store the information of the target places and also to retrieve the stored information. In other words, the user is limited with the Internet connection if they wish to tag a certain place, to show the tagged places on the map or to show the list of the tagged places.

The significance of using Hansel in this study is to compare it with our application, CatchIt, in terms of the ease of use even though the main goal of the applications is different. While CatchIt bookmarks the target moment, Hansel tags the target place.

3.2.2 *Contacts*

Contacts is a built-in *iPhone* application (Apple, 2012a) (see the user interface of one window of *Contacts* in Figure 3.2). It allows the user to insert new contact information as well as editing any information of the target contact. Adding a new contact details in *Contacts* has a wide range of the semi-required data (we called it semi-required because one of them is enough to save the contact details) and one of the optional ones. The instances of the semi-required data are the contact first name, last name and the mobile number. On the other hand, the only optional data, which is not enough to save the contact details, is the contact picture. Furthermore, *Contacts* allows the user to share the whole information of a certain contact via e-mail or message.

Contacts is an important application to compare it with *CatchIt* in terms of the ease of using the user interface. The user of *iPhone* interacts with *Contacts* frequently- each time they wish to add a new contact. Also, *Contacts* offers to the user rich information to be inserted and to be showed.

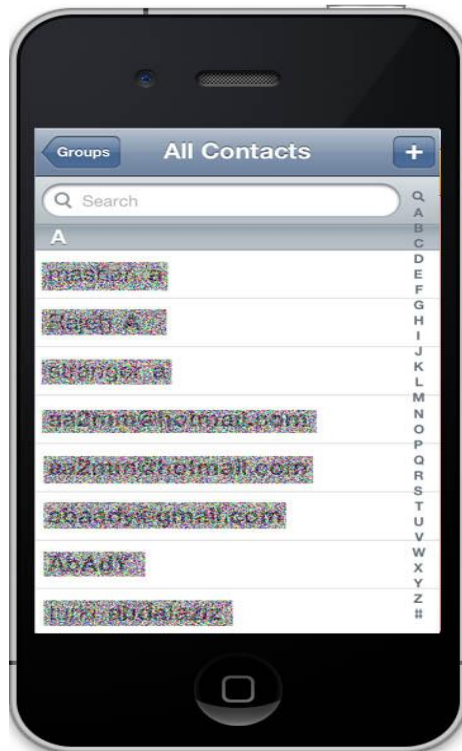


Figure 3.2: Contacts list in Contacts.

3.2.3 *Bookmarking a Moment*

Most of the system requirements have been identified in *bookmarking a moment* approach by Allalah and Hinze (2011). Therefore, we will shed light on this approach before identifying the whole requirements of our software.

3.2.3.1 Introduction

Bookmarking a moment is an approach that aims to capture some data (cues) of the selective moment, save it temporarily, and send it to Digital Parrot (discussed in Chapter 1), which assists to retrieve the required moment later (Allalah & Hinze, 2011). That means that *bookmarking a moment* approach presents both sensory memory (capturing data) and short-term memory (store information temporarily, process it, and translate it to long-term memory).

Bookmarking a Web Page Vs Bookmarking a Moment

The idea of *bookmarking a moment* approach comes from the same idea of bookmarking the web pages in the Internet, for the purpose of revisiting it later. While bookmarking web page allows the user to select their interesting web page, bookmarking moment allows the user to capture their interesting moment (see Table 3-1). In addition, both bookmarking a web page and moment save some information automatically. Bookmarking a web page saves the title and link of the web page automatically whereas *bookmarking a moment* saves the location and time of the moment automatically. Later, bookmarking a web page allows the user to change some stored information such as the title and add new information such as comments. In the same way, *bookmarking a moment* approach allows the user to modify the stored moment by changing some cues of the moment such as location and time as well as adding new information cues such as people and object cues. Another similarity between these two approaches is that the user is able to remove any bookmarked web page or moment.

However, because booking a web page does its functions in a fixed place (e.g. PC, laptop), it saves the bookmarked web page permanently (unless the user chooses to remove it) in a certain location of the user's computer's drive. This approach additionally, directly retrieves the bookmarked web page from the storage location. On the other hand, *bookmarking a moment* approach deals with different moments' locations- it saves the bookmarked moment temporarily in a portable device and retrieves the most recent bookmarked moments. Nevertheless, this approach overcomes problem by cooperating with another application- Digital Parrot (the symbol '[●]' in storing and retrieving cells in Table 3-1). *Bookmarking a moment*

sends the temporarily saved information to Digital Parrot for saving permanently and retrieving when required at a later time.

Table 3-1: Comparison between *bookmarking web page* and *bookmarking a moment* approach. The symbol ‘●’ means that the feature is supported in the approach while ‘[●]’ means that the feature is supported by the use of a cooperative system.

Features	Approaches	
	Bookmarking a web page	Bookmarking a moment
Fixed/ mobile	Fixed	Mobile
Capturing the required information	●	●
Storing the required information	●	[●]
Modifying the stored information	●	●
Adding new information	●	●
Removing the stored information	●	●
Retrieving the stored information	●	[●]

Bookmarking a Moment Features

Because some contexts will be captured automatically in this approach and some will be captured selectively by the user, *bookmarking a moment* offers important advantages to both sides: the system on the one hand and the user on the other.

From the system perspective, because the application allows the user to select their desired moments to be captured, the inserted information will be diminished. This process ensures that the system will consume not only less memory capacity but also storage space. In addition, when *bookmarking a moment* transfers the captured moments to Digital Parrot, Digital Parrot does not need to distinguish between the interesting and not-interesting moments of the user. As a result, this system will

dispense with some of the extra processes, which are present in other augment human memory systems.

From the user perspective, since the user captures their target moments selectively, all information is rich and valuable for the user. It is rich because it leads the user to remember more details of the target moments and reconstruct them when they collect these pieces of information (knowing, reconstructing and remembering). Also the inserted information is valuable because the information causes the user to remember their interesting moments. Furthermore, since the *bookmarking a moment* approach places the user in the loop of capturing and revising the moment, the user memory will be augmented (Jaimes, et al., 2004).

CatchIt

As a result of booking a moment approach, Allalah and Hinze (2011) did a study to investigate the appropriate moment cues, capturing types, and capturing ways. They introduced a system which called *CatchIt*. CatchIt is the application which fulfils the functions of this approach. It supports semi-automatic capturing of the required moment where the user captures the proper cues of their interesting moment (people, phone call, and object cues) in addition to capturing the time and location cues by the system itself. CatchIt allows the user to store temporarily the interesting moments, revise them, and then translate them to another application in order to store the target information permanently and retrieve the target moments at a later time.

The central idea of CatchIt is to have two interfaces. The first one is a device-based interface. In this interface the user can capture a moment and view (in a limited way)

the captured moments on small screen. Another option of this interface is sending the captured moment to the second interface, a computer-based interface. Computer-based interface has the ability of searching for the captured moment, showing the captured moments on a big screen, and also revising them. In addition, this interface can transfer the captured moment to Digital Parrot, which has powerful information retrieval ability.

Because no implementation has yet tested the application CatchIt, our software requirements will hover around the idea of CatchIt in order to implement it and, following implementation, to test it.

3.3 Software Requirements

In this section, we will give explanation of the software requirements of CatchIt, including the functional and non-functional requirements. Most of the software requirements of CatchIt have been suggested by Allalah and Hinze (2011). We will, however, not use all of their suggestions. In addition, we will adapt some, and ignore some according to our scenario in this chapter.

3.3.1 Functional requirements

The functional requirements (FR) “describe the functionality or services that the system is expected to provide” (Sommerville, 2001, p. 100). The functional requirements of our system, CatchIt, will describe the functions and services that this system offers to the user. As Allalah and Hinze (2011) observed, the functional requirements of CatchIt are essentially based on the following seven features:

portability, semi-autonomy, support capturing multimedia, synchronization, moment selection, support multi-languages text, and moment annotation. In addition, we will add our own suggestions which originate from the scenario, such as captured data annotation, revise information, and semantic representation, to the functional requirements.

Portability

Because the interesting moment does not have a specific time or location, CatchIt shall work at any time and in any location. The user, Baraa for example, used CatchIt when he was in the stadium (Section 3.1.1).

Semi-autonomy

Time and location contexts of the interested moment, being fleeting moments, are forgettable despite the importance of these contexts (Allalah & Hinze, 2011). Moreover, the user may be busy during the interesting moment. The user needs to ensure that these important contexts of their interested moment are captured. In addition, the user needs the system to capture their interested moment effortlessly. For instance, because Baraa was occupied with his friends and was busy watching the match, CatchIt captured the time and location for him once Baraa bookmarked the moment in the stadium (Section 3.1.1).

Furthermore, because CatchIt is a system that aims to augment the user memory, the user, when they are free, shall be able to capture some contexts of the moment manually, such as the surrounding people, the objects, and the emotions. In our

scenario for instance, Baraa took note of some of the events, which happened around him and his friend in the stadium (Section 3.1.2).

Support Capturing Multimedia

The diversity of capturing ways makes it possible to cover more recommended ways of capturing the interesting moment. For example, the user may choose to use *taking note* rather than *taking photos* or may choose to use *recording audio*. Thus, CatchIt shall support capturing multimedia (note-taking, photo-snapping, and audio/video recording). For instance, in our scenario, Baraa recorded the video of the haka and took a photo of Jodi and wrote her name as a caption of the photo (Section 3.1.2).

Synchronization

CatchIt is a portable system, which cooperates with another desktop system, Digital Parrot. While CatchIt captures the interesting moments, Digital Parrot organizes the captured information and also retrieves it. It can be seen therefore, that the synchronization between these two systems is a necessity. CatchIt shall translate the captured memory directly to Digital Parrot, or indirectly by using another application which is located in the same place as Digital Parrot. In our scenario, after Baraa captured his moments, he synchronized CatchIt and Digital Parrot so as to translate the captured information Digital Parrot (Section 3.1.2).

Moment Selection

CatchIt is not a lifelong capturing system. It shall capture the moments selected by the user, so that Baraa, for example, can select any moment he wants to capture (e.g. the moment in the stadium) and the moment of the stadium was captured by him (Section

3.1.1). Baraa, in addition, ignored the moment he did not want to capture (e.g. the moment in the university).

Support Multi-languages Texts

In the study by Allalah and Hinze (2011), there were some participants who used their first language to write down notes to capture the required information. This was possible because CatchIt shall support inserting textual information in various languages. Baraa, for instance, wrote notes on the captured data by using his first language, Arabic, and also his second language, English (Section 3.1.2).

Moment Annotation

Any recorded moment has to be distinct from other moments. It is impossible for example to have one moment occur twice; (in other words, the same moment cannot occur at two different times). Consequently, CatchIt shall associate in addition to the moment time, the moment location as default to the stored moments. Also, the user has an option to insert their own annotation for the stored moment (e.g. label the moment). As a result, the stage of retrieval of information will be easier when this stage accepts various query terms (e.g. time, location, the moment name). For instance, Baraa added the term *the stadium* as an annotation for his moment in the stadium (Section 3.1.1). Simultaneously, the time and location of the stadium moment were annotated by the system itself (Section 3.1.1). As a consequence, Baraa could retrieve the stadium moment by using the terms: *30th of July in 2011* (the time of moment), or *the stadium* (the label of moment) (Section 3.1.3).

Captured Data Annotation

To simplify extracting the important information from the capturing moment, CatchIt shall provide adding a caption of captured data. The user can add a description of the time, location, photo, video, or audio. For instance, Baraa added the caption *Westpac Stadium* for the location of the stadium moment (Section 3.1.2).

Revise Information

When the user is busy during capturing the target moment, they may insert inaccurate information, or miss capturing some desired cues of the moment. The user of CatchIt shall revise the recorded information. Another option for the user, suggested by Allalah and Hinze (2011) is revising the captured moment in a desktop application. However, we will offer revising the recorded moment in the portable device itself rather than revising the recorded moment in the desktop application. For instance, when Baraa wanted to support his target moment by adding extra related information for it after bookmarking the moment, he took a photo of Jodi and posted to the stadium moment (Section 3.1.2).

Semantic Representation

Because the coordinates of the location of the target moment are difficult to follow by the user, it is useful for the user to represent the location of the target moment semantically. Consequently, CatchIt shall represent automatically the location in the address format as well as in coordinate's format. Baraa can see the address of the stadium moment, *Westpac Stadium, Wellington, New Zealand*, the semantic information of the location, as well as the coordinates of it, *-41.273273, 174.787051* (Section 3.1.2).

CatchIt also shall represent the semantic description of the location and the time of the target moment if the user had described them during revising the target moment, as Allalah and Hinze (2011) suggested. The time, for instance, can be represented as seasons or birthdays. In our scenario, because Baraa described the time of the target moment as *the day I first saw a rugby match*, CatchIt will represent the time of the stadium moment by in accordance with Baraa description and also by using time/date format (Section 3.1.2).

3.3.2 Non-functional Requirements

Non-functional requirements (NFR) address the characteristics, properties, or qualities that the system must have (Westfall, 2006) in addition to the constraints on input/output devices that are used in the system and any requirements that make the system usable as a whole (Sommerville, 2001). The nine non-functional requirements provided by Allalah and Hinze (2011) are: comfort, appropriate size, storage (longevity of the system), location device, digital camera, audio recorder, ease of use, privacy, and portable device-based interface. Moreover, we will address additional non-functional requirements of CatchIt. We will add local wireless/ wireless network, the accuracy, the availability requirements, and error-handling requirements.

Comfort

Because the interesting moment is located everywhere, the user should carry the wearable device wherever they feel comfortable with it. Therefore, the device that CatchIt is placed in shall be comfortable. In our scenario for example, Baraa was comfortable with carrying our capturing device (Section 3.1.1).

Appropriate Size

The small size of the device helps it to be more portable. Also, the lightweight is another aspect that makes the device much more portable. Mainly for the reason of portability, the device shall be small and lightweight. The capturing device in our scenario was small and light (Section 3.1.1).

Storage (Longevity of the System)

Due to using different data format to save the interesting moment in addition to the possibility of holding the captured moment for a while until they are transferred to Digital Parrot, the device shall have large capacity of storage disk. In our scenario, for instance, because CatchIt allowed Baraa to capture up to one minute, Baraa could not make lengthy video recorded of his target moment although the capturing device has a big storage capability (3.1.2). So that Baraa saved more capacity of CatchIt to store new moments at a later time.

Location Device

One of the most important contexts of the moment is its location. To get the information about the location of bookmarked moment, the device shall connect with the Global Positioning System (GPS) or shall have a built-in GPS in it. By using a built-in GPS, the capturing device in our scenario could detect Baraa's exact location (Section 3.1.1).

Digital Camera

For taking photos, or for recording video to capture the cues of the target moment, the device shall have digital camera. Baraa in Section 3.1.2 could record a short video of

the haka and take a photo of Jodi with built-in digital camera in his portable device (Section 3.1.2).

Audio Recorder

CatchIt uses audio recording capturing ways. Consequently, the device shall have an audio recorder. Because the wearable device in the scenario (Section 3.1.2) supported audio recording, Baraa could have an audio record of the conversation between him and Jodi.

Ease of Use

CatchIt has several capturing ways to capture the target moments (taking note, taking photos, video recording, and audio recording). In addition, CatchIt has several services (bookmark moment, revise moment, and synchronize moment), and also it uses sensor to detect the location of the moment. The device as a result shall be easy to use and shall not require extra training. For the whole scenario Baraa had no difficulties using our system.

Privacy

The system allows the user to select their own moments to be bookmarked and all other information they want to be captured. Additionally, the system needs to use the current location of the user when the user bookmarks the target moment. The current location of the user is personal information. To use this sort of information the system shall request acceptance of using the current location of the user before using the system in first place. In our scenario, Baraa was confident when using our system because the system guaranteed his privacy.

Interactive Interface

Marking a moment works on two related applications as Allalah and Hinze (2011) suggested; one of them is located in a portable device and another one is located in the desktop of the user's computer. The portable device-based interface captures the required moments and translates those moments to the desktop application whereas the computer-based interface revises the required moments and submits those moments to Digital Parrot. However, we will add revising option to the portable device-based interface rather than using this option just on the computer-based interface. The portable device-based interface of CatchIt shall implement its functions (capturing, revising, and synchronizing) with an interactive way with the user. The system interacted with Baraa in the whole scenario, when he captured, revised, synchronized information.

Access to the Internet

The local wireless network ensures that the device is connected to the Internet. That helps the system to get the address of the target moment when it sends the coordinates and receives the address. That happened in our scenario (Section 3.1.2) when the system converted the coordinates of Baraa's location to the address format. Hence, the device shall have a port to access into the Internet.

Accuracy

The accuracy of inserted data ensures that the system will work more accurate. Consequently, the system will not deal with inaccurate values or format during inserting data. For instance, Baraa could not change the time of the bookmarked moment (Section 3.1.2) when he tried to use inaccurate value of the moment time.

Availability

By identifying the available processes and data the system provides, the number of errors the user might make while using our system will be diminished. All the required processes which are needed during all running time of the application such as bookmarking a moment shall be available whenever the user wishes (Baraa could capture whatever and whenever he wants (Section 3.1.1)). What else, there are processes shall be available under some conditions. For instance, revising an earlier moment shall not be available until there is at least one bookmarked moment in the system (Baraa could not revise a moment until he bookmarked a moment (Section 3.1.1)). In other words, the system shall make these processes partly available. Moreover, the captured data shall be available until the user sends the captured information to Digital Parrot.

Error-handling

Sometimes the user may make some mistakes of using our system. In this circumstance, the system shall avoid these mistakes in addition to treat them. The system also shall show the user the reasons behind these mistakes in appropriate way such as using pop-up message. As a result, the user will know their mistakes of using and they will try to fix them when they use the system again. For instance, when Baraa changed by mistake the time of the bookmarked moment, a message popped up to tell him why there was an error of changing the time of bookmarked moment (Section 3.1.2).

3.4 Summary

To identify the requirements of the CatchIt system, we reviewed the scenario from Chapter 1 and adapted it to the purposes of this chapter. Then we introduced the related works, *Hansel*, *Contacts* and *Bookmark a moment*. The first two related work, Hansel and Contacts will be used later in evaluating the user interface of our application when we will compare between then and our application. On the other hand, the third related work used to highlight some requirements to implement our system combining with some of the scenario-based requirements. In general, the requirements we covered in this chapter were under either the functional requirements or non-functional requirements.

We can sum up the functional requirements of CatchIt in the following points:

FR(1) Portability

- a. The system shall be used whenever the user wants and wherever the user is.

FR(2) Semi-autonomy

- a. The system shall record the time of the captured moment.
- b. The system shall record the location of the captured moment (coordinates format).
- c. The user shall record the people cue of the captured moment if the user wishes.
- d. The user shall record the object cue of the captured moment if the user wishes.
- e. The user shall record the emotion cue of the captured moment if the user wishes.

- f. The user shall record the phone call cue of the captured moment the user wishes.
- g. The user shall record any event which is happened during the captured moment if the user wishes.

FR(3) Support Capturing Multimedia

- a. The system shall support a taking-note ways to capture any required data.
- b. The system shall support a taking-photo(s) ways to capture any required data.
- c. The system shall support a video recording ways to capture any required data.
- d. The system shall support an audio recording ways to capture any required data.

FR(4) Synchronization

- a. The system should transfer the captured data from CatchIt to Digital Parrot.

FR(5) Moment Selection

- a. The user shall select their interesting moment without any interference from the system itself.

FR(6) Support Multi-languages Text

- a. The system shall provide multi-languages to enable insertion of the textual information.

FR(7) Moment Annotation

- a. The use shall annotate automatically the captured moment by using the time of the moment.

- b. The user shall annotate automatically the captured moment by using the location of the moment.
- c. The user shall annotate the captured moment by labeling the moment if they want.

FR(8) Captured Data Annotation

- a. The user shall annotate the captured data when they use a taking-photo(s) ways if they want.
- b. The user shall annotate the captured data when they use a video recording ways if they want.
- c. The user shall annotate the captured data when they use an audio recording ways if they want.

FR(9) Revise Information

- a. The user shall revise their captured moment at any point of time until they decide to translate the captured moment into Digital Parrot.

FR(10) Semantic Representation

- a. The system shall represent the address of the location of the captured moment (address format).

With regard to the non-functional requirements, we can sum up in the following points:

NFR(1). Comfort

- a. The user shall feel comfortable when they use the capturing device.

NFR(2). Appropriate Size

- a. A captured device should be small.
- b. A captured device should be lightweight.

NFR(3). Storage (Longevity of the System)

- a. A captured device shall have large storage capacity.
- b. The length of recorded video shall be one minute or less.
- c. The length of recorded audio shall be one minute or less.

NFR(4). Location Device

- a. A captured device shall have a built-in GPS.

NFR(5). Digital Camera

- a. A captured device shall have a digital camera to take a digital photo.
- b. A captured device shall have a digital camera to record the digital video.

NFR(6). Audio Recorder

- a. A captured device shall have an audio recorder.

NFR(7). Ease of Use

- a. The system shall be easy to use.
- b. The system should not require extra training to be used.

NFR(8). Privacy

- a. The system shall make the user selects the target moment on their own.
- b. The system shall make the user captures whatever data they want on their own.
- c. The system shall ask the user to use their current location the first time they use the system.

NFR(9). Interactive Interface

- a. The system shall interact with the user appropriately.

NFR(10). Access to the Internet

- a. The system shall connect to the Internet when it is needed.

NFR(11). Accuracy

- a. Date/time of moment must be not used before the current date/time.

- b. Date/time of moment must not be used for another moment.

NFR(12). Availability

- a. *Bookmark a moment* is available at all times.
- b. *Revise a moment* is available solely if there is one or more captured moment.
- c. Synchronize the application with other devices is available solely if there is one or more captured moment.
- d. Delete a moment is available solely if there is one or more captured moment.
- e. Save new changing is available solely if the user changes or modifies the captured data.
- f. All captured moments and their associated data will be available until the user transfers them to Digital Parrot.

NFR(13). Error-handling

- a. The system shall handle any unaccepted values of data.

The next chapter will put the highlighted requirements of the system, which were explained in this chapter, into our account to design CatchIt.

4. Conceptual System Design

After identifying the requirements of the system (in Chapter 3), the design of the system will be discussed in more details in this chapter. These details involve the conceptual architecture of CatchIt, overview of the user interface prototype, the initial design of the user interface prototype, and the final design of the user interface prototype.

Towards the end of this chapter the appropriate way of the interaction between the main functions of our application and the user will be found. Indeed, CatchIt bookmarks the selected moments to be revisited if the user requests. In addition, CatchIt combines with Digital Parrot in order to get the benefits of using its search engine to retrieve the bookmarked moment and to get the benefits of enabling the user to look for the target bookmarked moment by using different ways of searching what Digital Parrot has.

4.1 Conceptual Architecture

The conceptual architecture of our system for bookmarking moments (CatchIt), which builds on the results of the requirements in Chapter 3, will be discussed in this section. Figure 4.1 shows the procedures involved in interacting with CatchIt. CatchIt includes two components: the local storage and the user interface. The connection between the local storage and the user interface will allow CatchIt to perform its main steps, *bookmark a moment*, *revise* and *synchronize* CatchIt with Digital Parrot, (see Figure 4.1).

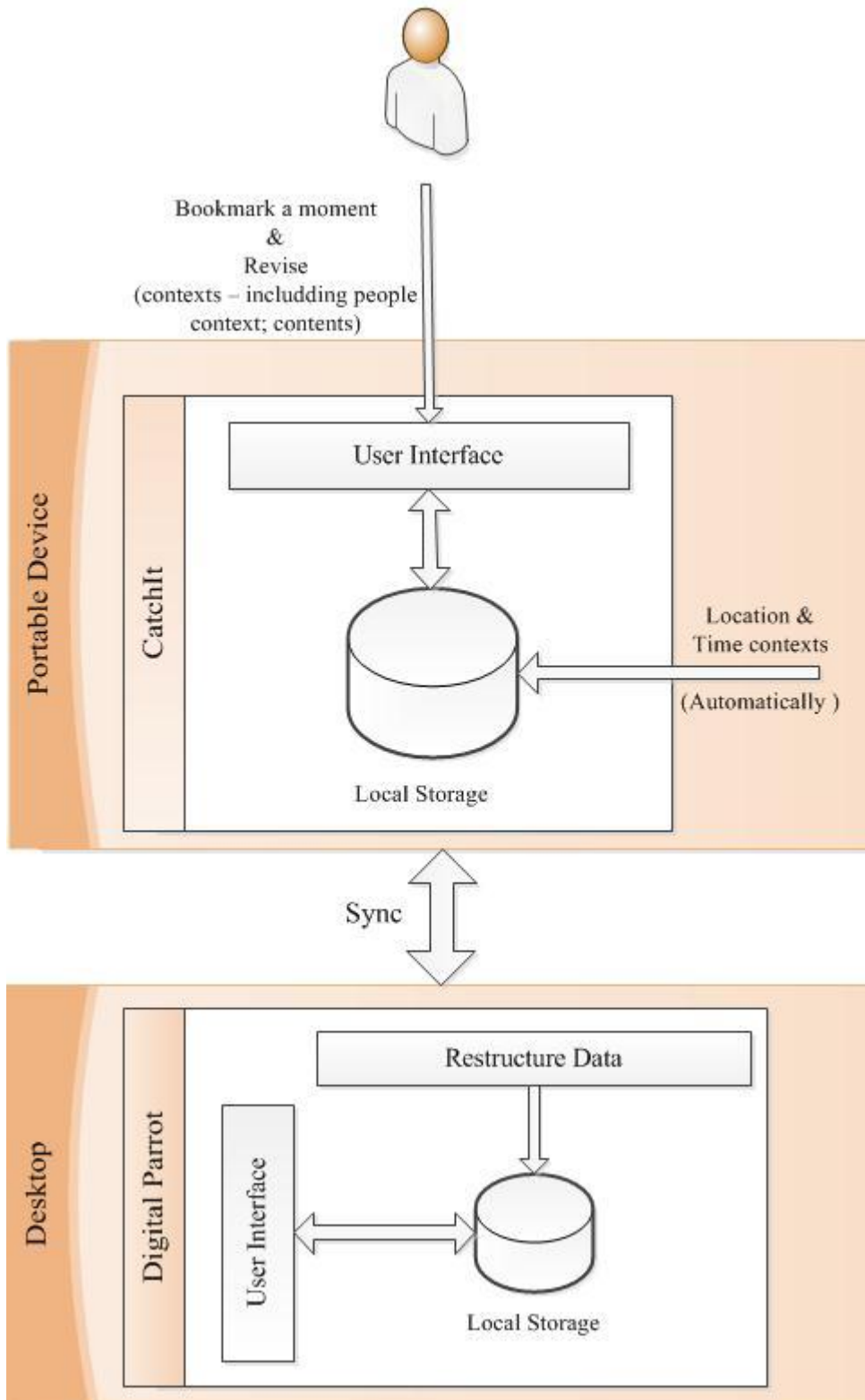


Figure 4.1: Conceptual architecture of CatchIt. The user interacts with the user interface to bookmark the target moment, revise it, and synchronize CatchIt with Digital Parrot to send the data of the bookmarked moment from CatchIt to Digital Parrot. All data of the target moments (contexts: location, time captured automatically and people captured manually; contents: such as the action and clothes of the user) will be stored in a local storage.

4.1.1 Bookmark a Moment

Bookmarking a moment is the first step of CatchIt. It allows the user to bookmark the selected moments. The user interacts with the user interface in order to bookmark the target moment (e.g., during the day as an event is happening). In this step, the contexts of location and time of the target moment will be captured automatically on behalf of the user. Then the captured values of the contexts of the location and time of the target moment will be stored in the local storage of the portable device to be used if they are required (see Figure 4.2).

The user can add further information into the bookmarked moments as well as changing and deleting information of the bookmarked moment from the next step.

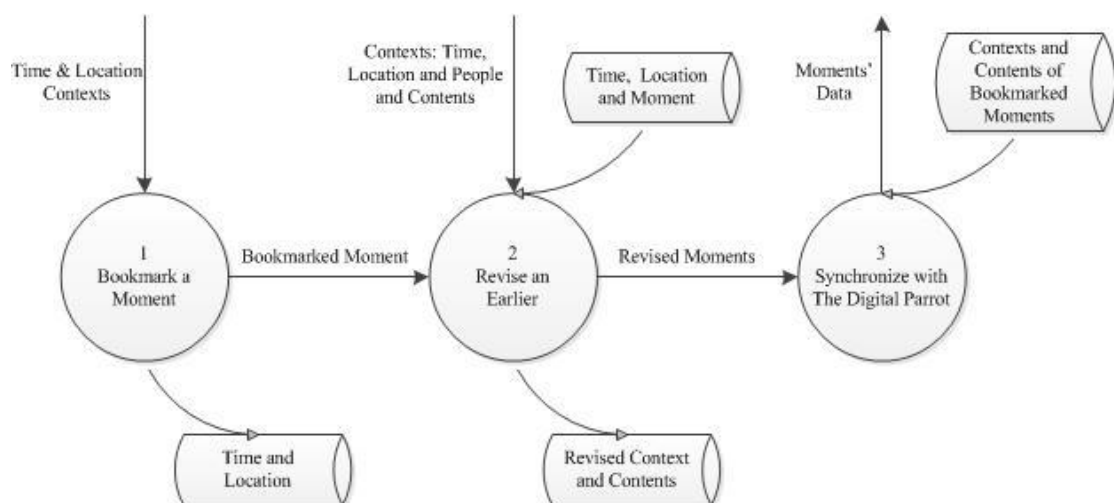


Figure 4.2: Data flow diagram of CatchIt. The portable device feeds CatchIt with the time and location contexts in the first step while the user adds the people context and contents of the bookmarked moment with the ability of changing the time and location contexts in the second process. The third process transfers the stored contexts and contents of the bookmarked moments into another application, Digital Parrot.

4.1.2 Revise

The revising step may include adding some values (context: people; contents: such as what the user was doing at that moment or what did they wear), changing values of the existing data or deleting some values of a target moment. All the components of the revising process are performed manually.

To do weather adding new values, changing the target values or deleting the target values of a particular moment (for example, at the end of the day), firstly, CatchIt will connect with the local storage. This operation will retrieve the stored time values of all bookmarked moments. All time values of the bookmarked moments will be listed in the user interface. Therefore, the user can choose whatever bookmarked moment they wish to revise. Then, both sorted time value and location value of the target bookmarked moment will be retrieved from the local storage. The updated values of any supported context (time, location and people) or any contents will be saved in the local storage (see Figure 4.2).

The revised bookmarked moments have the ability to be viewed whenever the user wishes or to be transferred to another application (Digital Parrot) from the next step.

4.1.3 Synchronize CatchIt with Digital Parrot

The user interacts with the user interface to synchronize two systems, CatchIt and Digital Parrot. In this step, CatchIt interacts with its local storage to retrieve all related data of the bookmarked moments, contexts and contents, (see Figure 4.2). Then CatchIt generates an XML file which includes some of the retrieval data from the local storage and sends the generated XML file to Digital Parrot. Digital Parrot is

required to be developed to succeed the synchronization. It should develop a way to restructure the XML file to the appropriate format for itself and then store the restructured data in its local storage (see Figure 4.1).

4.2 Overview Prototype Interface

As mentioned in the previous section, CatchIt has three basic steps that the interface should include: (1) bookmarking a moment, (2) revising the bookmarked moments, and (3) synchronizing CatchIt with Digital Parrot. After bookmarking the target moment (the first step), the user will be able to either revise the bookmarked moments (the second step) by using several capturing ways (taking notes, taking photo, recording video, and recording audio) or transferring the data of all bookmarked moments to Digital Parrot (the third step).

As a result of the main three steps of CatchIt, the user interface of CatchIt initially has 15 different pages that the user can move among and interact with. Each page has a particular function of our system. For example, there is a responsible page of adding an audio record into the bookmarked moment. In addition to pages, there are various message boxes which give the user an appropriate instruction or information about the user interaction with the functions such as deleting a bookmarked moment. For example, before deleting a bookmarked moment the system will warn the user that deleting that moment means that the bookmarked moment will not be available any more in the system.

All pages and message boxes of the system are shown in overview in Figure 4.3. The pages are colored red whereas the message boxes are colored orange. Order numbers

identify the pages in this figure while alphabetical characters identify the message boxes. The same figure also consists of arrows among the pages and the message boxes. These arrows describe the transition between pages and message boxes of CatchIt.

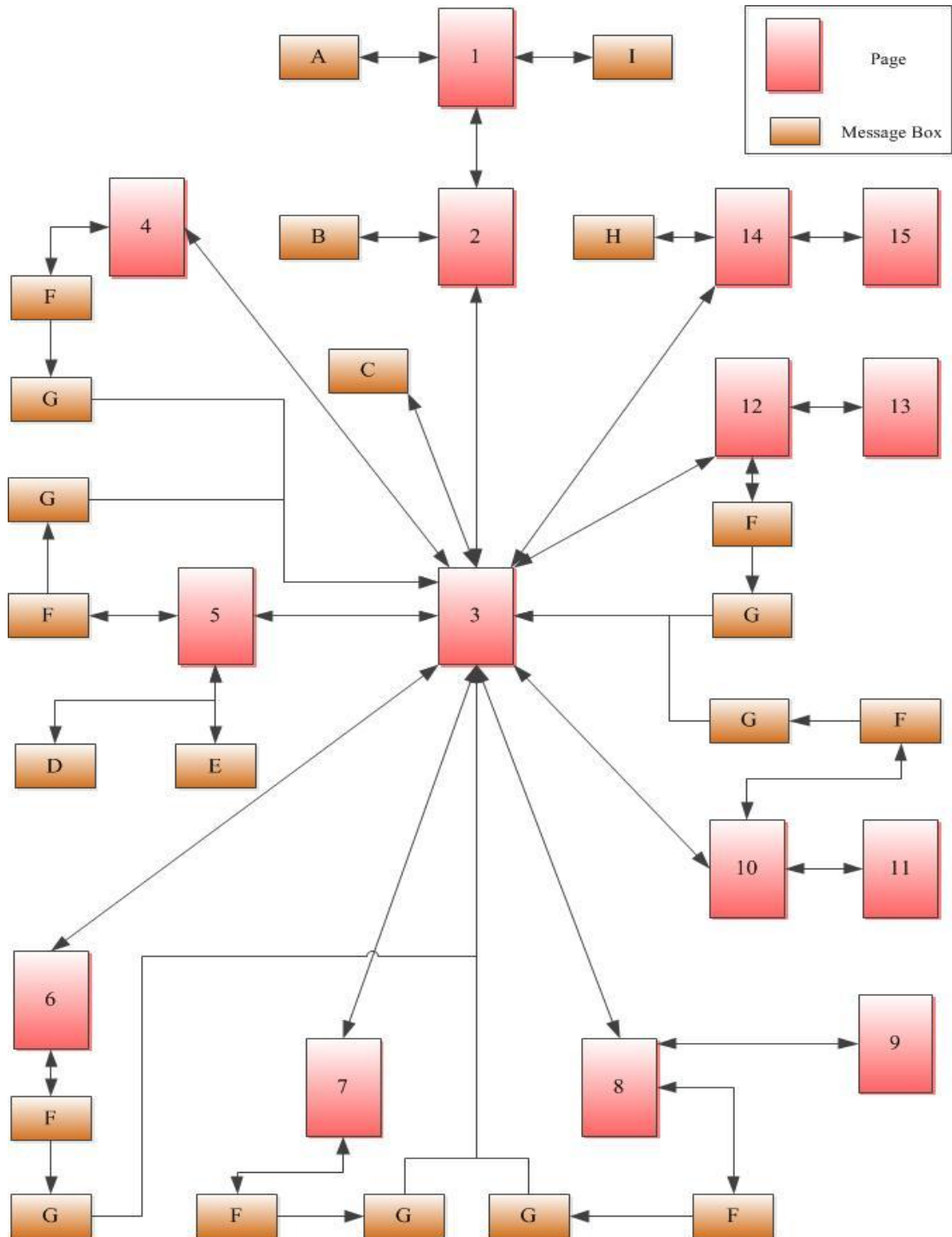


Figure 4.3: Overview prototype of CatchIt. The key map is shown the upright of the figure.

Message Box A and *Message Box I* are popped up in *Page 1: The Main Menu* which is connected to *Page 2: Bookmarked Moments*. *Page 2: Bookmarked Moments* has a *Message Box B* and from this page, it is able to move to *Page 3: Editing Moment*. From *Page 3: Editing Moment*, the user of the system can move to *Page 4: Editing Moment's Name*, *Page 5: Editing Time*, *Page 6: Editing Location*, *Page 7: Editing Text*, *Page 8: Editing Photo*, *Page 10: Editing Video*, *Page 12: Editing Audio* and *Page 14: Editing People* and then back to *Page 3: Editing Moment*.

In addition to all these connections between *Page 3: Editing Moment* and the connected pages, *Message Box C* is popped up in *Page 3: Editing Moment*. *Page 4: Editing Moment's Name*, *Page 5: Editing Time*, *Page 6: Editing Location*, *Page 7: Editing Text*, *Page 8: Editing Photo*, *Page 10: Editing Video* and *Page 12: Editing Audio* have the same message boxes, *Message Box F* and *Message Box G*. Those pages are connected directly to *Message Box F* which whether returns the user to the page that the message box came from or shows *Message Box G*. If *Message Box G* is popped up, CatchIt will back immediately to *Page 3: Editing Moment*. Other message boxes are *Message Box D* and *Message Box E* which they are related to *Page 5: Editing Time* and *Message Box H* which is related to *Page 14: Editing people*.

After moving to *Page 8: Editing Photo* via *Page 3: Editing Moment*, the user can reach *Page 9: Capturing Photo* and back again to *Page 8: Editing Photo*. Similarly, *Page 11: Capturing Video*, *Page 13: Capturing Audio* and *Page 15: Contacts* are reached from *Page 10: Editing Video*, *Page 12: Editing Audio* and *Page 14: Editing People* respectively and vice versa.

It is obvious that Figure 4.2 does not have many details on the functions that the pages have or the reason behind showing the message boxes to the user. However, it has overview connections between both pages and message boxes in our system. Therefore, walking through the prototype in the next section will be easier when each view is highlighted.

4.3 Initial User Interface Prototype

The focus of this section is on each view rather than taking an overview of the system as a whole. Each view has various kinds of controls, which help to insert data, process that data and then display the information to the user.

The three main functions of CatchIt (a. *bookmarking a moment*, b. *revising an earlier moment*, and c. *synchronize CatchIt with Digital Parrot*) are located in *Page 1: The Main Menu* in addition to the brief information about purpose of the application. Each main function is reachable by a particular button of this page. Thus, *Page 1: The Main Menu* is not just the first view, it is the main one (see Figure 4.4).

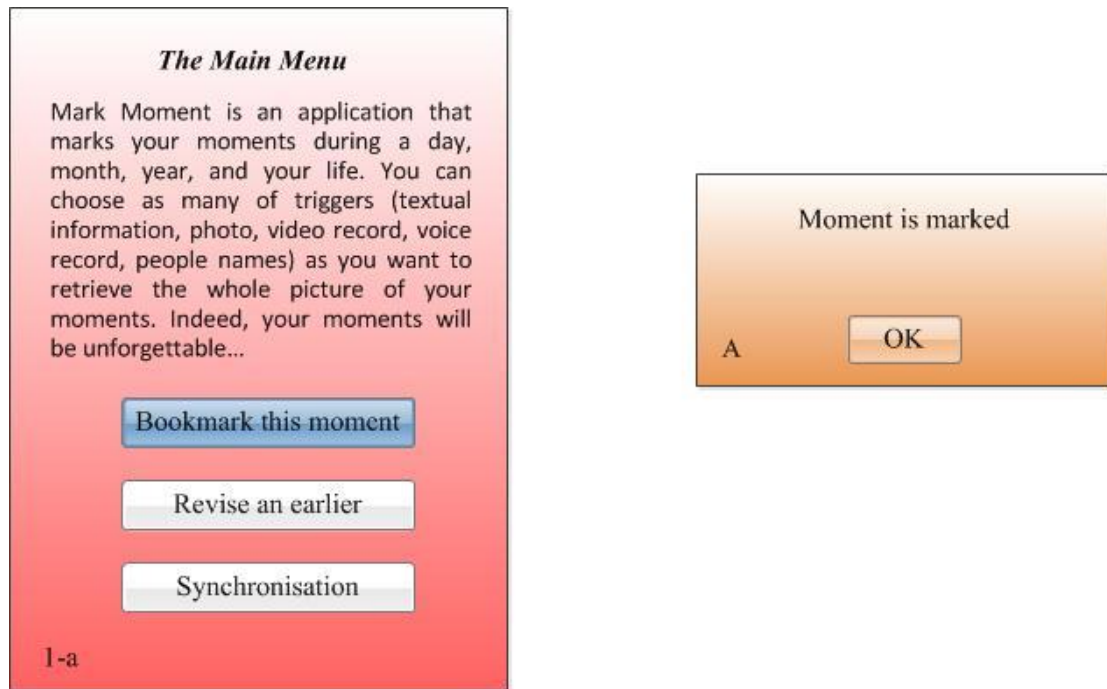


Figure 4.4: The main view of CatchIt, *Page 1: The Main Menu*, at the initial running. Bookmarking a moment is the only main function is available. At the right side, *Message Box A* is an associated message box with *Page 1: The Main Menu*.

4.3.1 Bookmark a Moment

In initial use of CatchIt, the *Bookmark this moment* button is the only active button on page (Figure 4.4). If the *Bookmark this moment* button is pressed, both the current time and the current location will be captured and saved automatically. Then, *Message box A* will pop up, which informs the user that the required moment has been bookmarked. Other related information of the bookmarked moment will be inserted manually if the user wishes. Once the first moment in the system is saved, the buttons *Revise an earlier* and *Synchronization* of *Page 1: The Main Menu* will be available to the user (Figure 4.5).

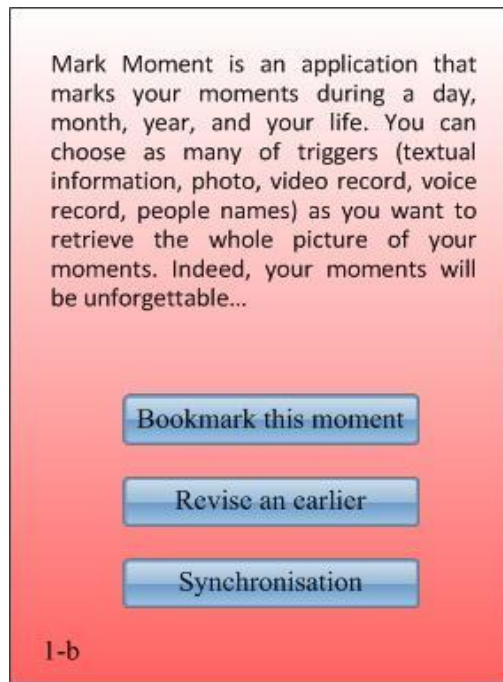


Figure 4.5: All buttons of *Page 1: The Main Menu* are available as long as there is at least one bookmarked moment.

4.3.2 Revise an Earlier

While the button *Revise an earlier* is available the user is allowed to move to *Page 2: Bookmarked Moments*. The bookmarked moments will be listed in *Page 2: Bookmarked Moments* (see Figure 4.6). The time and the name of each bookmarked moment will be shown in each item of the list. Because the name of the bookmarked moment is not provided automatically by the system the name of the bookmarked moment will be *No Name*.

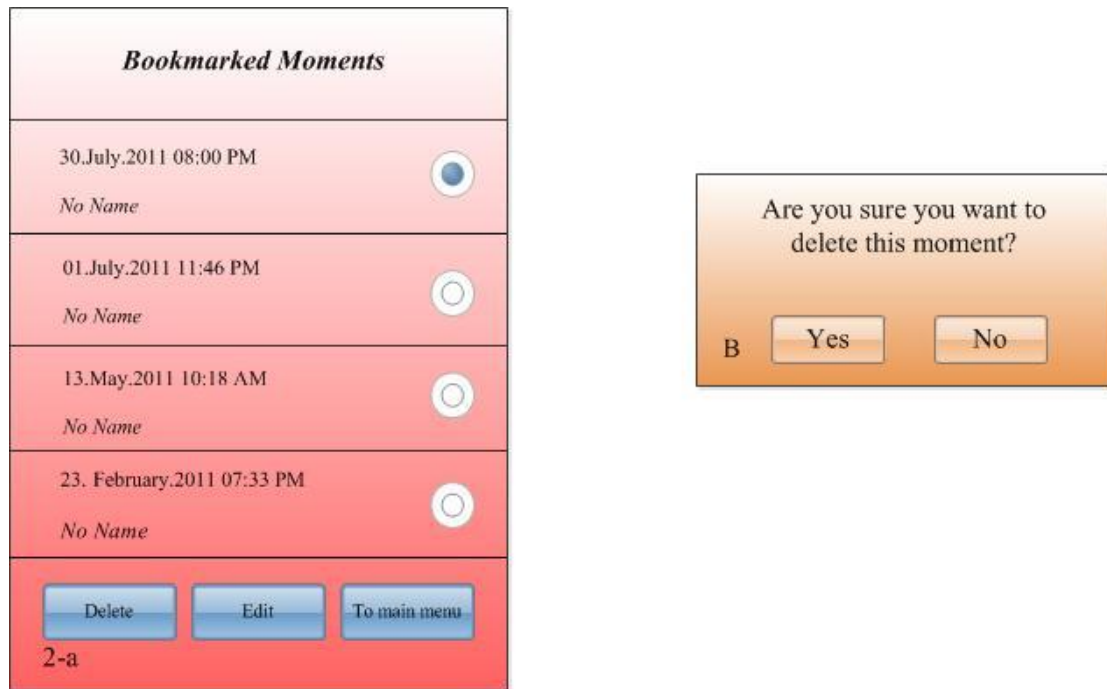


Figure 4.6: All bookmarked moment are listed in *Page 2: Bookmarked Moments*. Each item of the list has the time and the name of the bookmarked moment.

The user can edit or delete any bookmarked moment from the list of *Page 2: Bookmarked Moment* after selecting a bookmarked moment. Also the user can go back to the main page, *Page 1: The Main Menu*. *Message Box B* will pop up if the user wishes to delete a particular bookmarked moment after selecting it and pressing *Delete*. On the other hand the button *Edit* allows the user to edit any related information of the selected bookmarked moment when CatchIt transfers the user to *Page 3: Editing Moment*.

Page 3: Editing Moment has a list of the captured details of the bookmarked moments (Figure 4.7). The time and location are provided to the user by the system and the name of the bookmarked moment, any note about the moment, token photos of that moment and both recorded types video and audio are able to the user to be involved

into the bookmarked moment. Although the time and location of the bookmarked moment are captured automatically, the user has the ability to modify them.

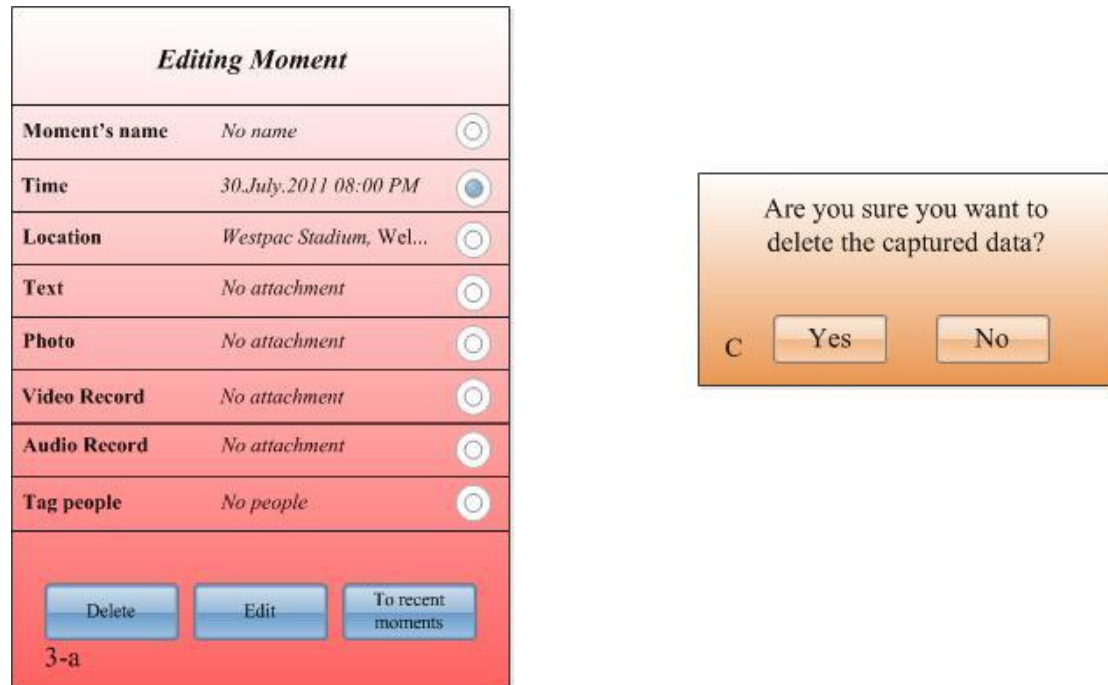


Figure 4.7: The values of the components of the bookmarked moment are changeable and can be deleted from *Page 3: Editing Moment*. The default values of time and location are the current time and location respectively and they are provided by CatchIt. The value of the rest of components is null. At the left side of the figure the message box *Message Box C* pops up in *Page 3: Editing Moment* when the user is going to delete a component of the bookmarked moment.

Deleting any captured data is allowed in *Page 3: Editing Moment* except the time and the location because they are the significant identifiers of the bookmarked moment. The user can delete the token note, all photos, all videos and all audios that are associated with the bookmarked moment by selecting the capture type that CatchIt offers; the user deletes by pressing the *Delete* button. Deleting any data means that the user cannot retrieve it later on. So, CatchIt makes sure that the user really wishes to delete that data by providing the pop-up option *Message Box C*.

The button *Edit* in *Page 3: Editing Moment* involves several functions: adding one or more captured data into a particular capturing type, updating and deleting a particular captured data of one of the capturing type. To edit a collection of captured data under one of the capturing types (moment's name, time, location, text, photo, video and audio) the user selects the capturing type from the list in *Page 3: Editing Moment* then presses *Edit* button to move to the required page.

As it can be seen in Figure 4.6, the default values of moment's name, text, photo, video record and audio record in *Page 3: Editing Moment* is null. However, firstly, the user can add any desired data into any preferred cue types and then the user can change that data or delete it.

Moment's Name

To add or modify the moment's name of the bookmarked moment, the user selects *Moment's name* from *Page 3: Editing Moment* to be moved to *Page 4: Editing Moment's Name* (see Figure 4.8). There is a text field in *Page 4: Editing Moment's Name* to type the moment's name such as *First experience with Ruby* and also there is an inactive button *Save* to save and activated button *Back* to move the user back to *Page 3: Editing Moment* without any changes. After editing the moment's name (see Figure 4.9 which shows *Page 4: Editing Moment's Name* when a new moment's name is added) the user can save the changes by pressing the *Save* button which is activated after adding a name for the moment.

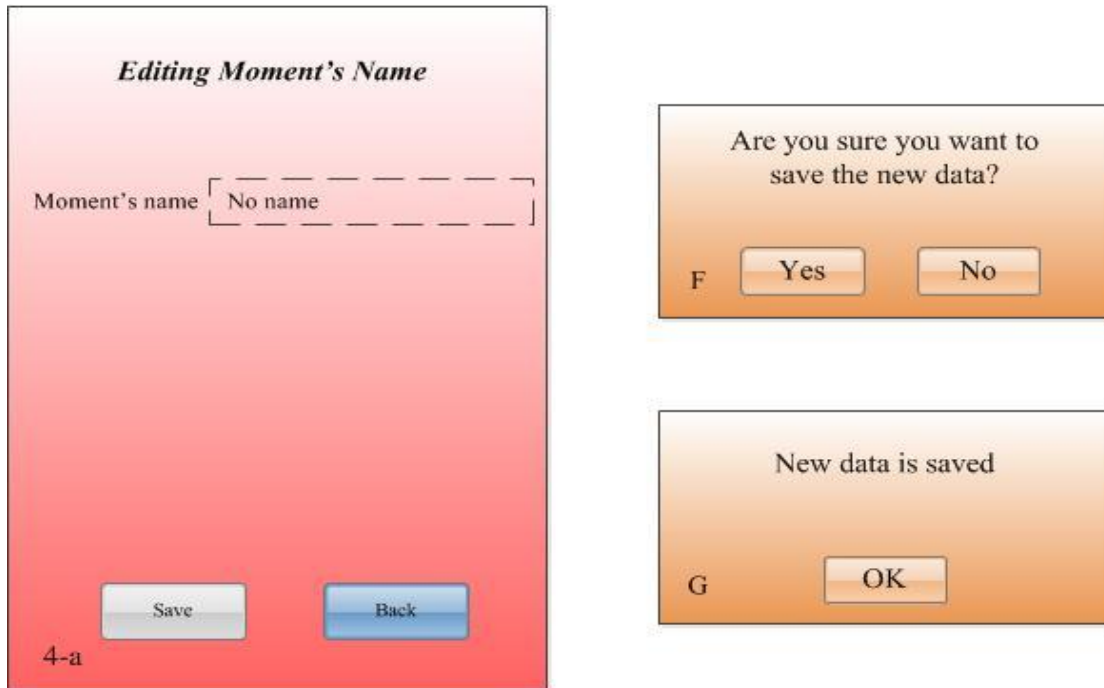


Figure 4.8: *Page 3: Editing Moment's Name* before adding a moment's name. The *Save* button is not available until the user adds a moment's name. At the left side, *Message Box F* and *Message Box G* are the associated message boxes of *Page 4: Editing Moment's Name*.

If the user presses *Save* in order to save the changes, *Message Box F* will pop up to ensure that the user wishes to save the new changes of the moment's name. If the user is sure to save the changes, CatchIt will save the changes, giving the user appropriate feedback in *Message box G* about the saving operation and then move back to *Page 3: Editing Moment*. On the other hand, if the user wishes to ignore saving the changes, the user will press the button *No* in *Message Box F* to go back to *Page 4: Editing Moment's Name*.

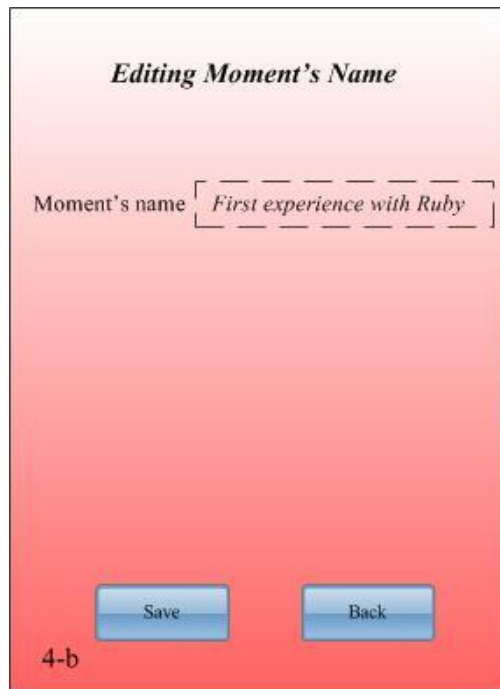


Figure 4.9: Saving a moment's name is available in *Page 4: Editing Moment's Name* if there is a new value of it.

Time

Although CatchIt captures the current time of the bookmarked moment and uses it as the default value for the time of that moment, the user is allowed to change this value from *Page 5: Editing Time*. To reach *Page 5: Editing Time*, the user selects the second item in the list of *Page 3: Editing Moment, Time*, and then the user presses *Edit*.

Page 5: Editing Time has two text fields and two buttons (see Figure 4.10). The first text field is for inserting the time of the moment whereas the second text field is for inserting a description of that time. The two buttons in *Page 5: Editing Time* are *Save* which is inactivated if there are no changes, and *Back* which is activated all the time to enable user to go back to *Page 3: Editing Moment*.

Editing Time

Time

Time description

5-a

Figure 4.10: Editing the time of the bookmarked moment is the function of *Page 5: Editing Time*. The user can change the captured time and add a description of it. It is not allowed to save the data unless changes have happened to the data of the page.

Nevertheless, there are two main conditions which are applied if the user wishes to change the time of the bookmarked moment. The user can change the time of the bookmarked moment as long as the new time is unused previously with another bookmarked moment. For instance, if the user used *30.July.2011 08:00 PM* in one of the bookmarked moments, CatchIt will not accept that time in another bookmarked moment because the times are not two different moments, but the same time. In this sort of inserting errors, CatchIt informs the user about the errors of using the new time in *Message Box D*. Therefore, the user will recognize their mistake. (see Figure 4.11, *Message Box D* and *Message Box E* inform the user about error of changing the moment time while *Message Box F* checks if the user wishes to save the new changes and *Message Box G* lets the user know about the last action that happened in *Page 5: Editing Time*).

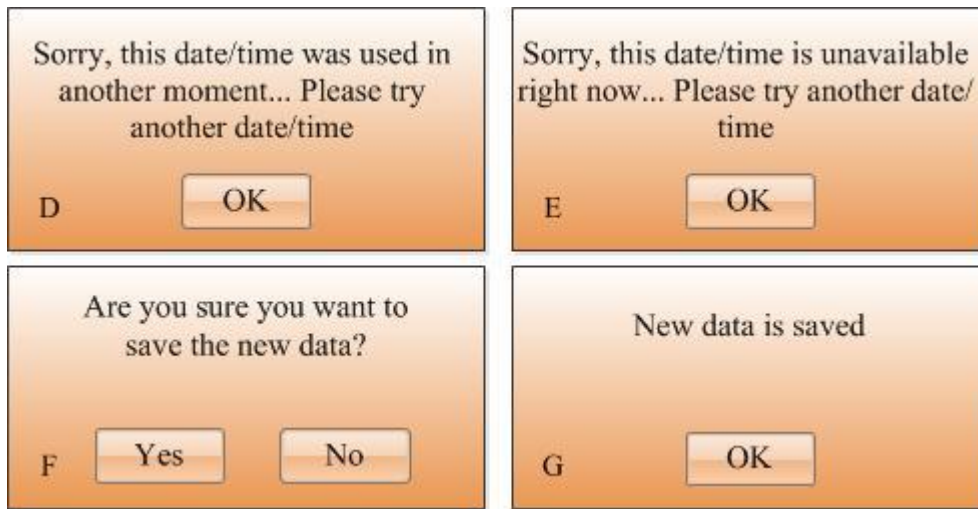


Figure 4.11: Associated message boxes of *Page 5: Editing Time*.

In addition, the new time cannot be a forward time. The future time is invalid in CatchIt. CatchIt only bookmarks the current moment or moments that happened in the past. For instance, the user cannot set the time of the bookmarking moment to be *30.July.2011 08:00 PM* if the current time is *22.July.2011 08:00 PM*. In this circumstance *Message Box E* will pop up to instruct the user that the requested time is invalid (Figure 4.11).

As long as the new time is valid (not used previously in different moment and not a forward time) CatchIt will ensure if the user wants to save the new time by showing *Message Box F*. Once the user verifies their request CatchIt will save the new time and show *Message Box G* (Figure 4.12 shows *Page 5: Editing time* with a valid time). However, if the user decides to ignore saving the new time, the user can press the button *No* in *Message Box F* to go backward to *Page 5: Editing Time*.

Editing Time

Time [28.July.2011 11:46 PM]

Time description [After Uni]

Save Back

5-b

Figure 4.12: A valid inserted time in *Page 5: Editing Time* to the bookmarked moment.

In addition to modifying the time of the bookmarked moment in *Page 5: Editing Time*, the user can describe time as they wish.

Location

The location of the bookmarked moment is provided automatically, same as the time of that moment. The provided location is the current location of the bookmarked moment. However, the user can to change that location if they wish, as in case of bookmarking a moment from the past. For example, when the user is at home and they wish to bookmark a moment they had two hours ago in the cinema.

To change the location of the bookmarked moment the user reaches to *Page 6: Editing Location* via *Page 3: Editing Moment*. The provided location is shown in *Page 6: Editing Location* in addition to an empty text field to insert a description of

that location (see Figure 4.13). At the bottom of the page the *Save* button is not available until changes have happened to the data of the page. Additionally, there is a button *Back* to move the user backward to *Page 3: Editing Moment*.

The user is allowed to add any description they wish. The location description will help in retrieving moments when the description or part of it is used as a query term (the retrieval stage is not covered in this research thesis).

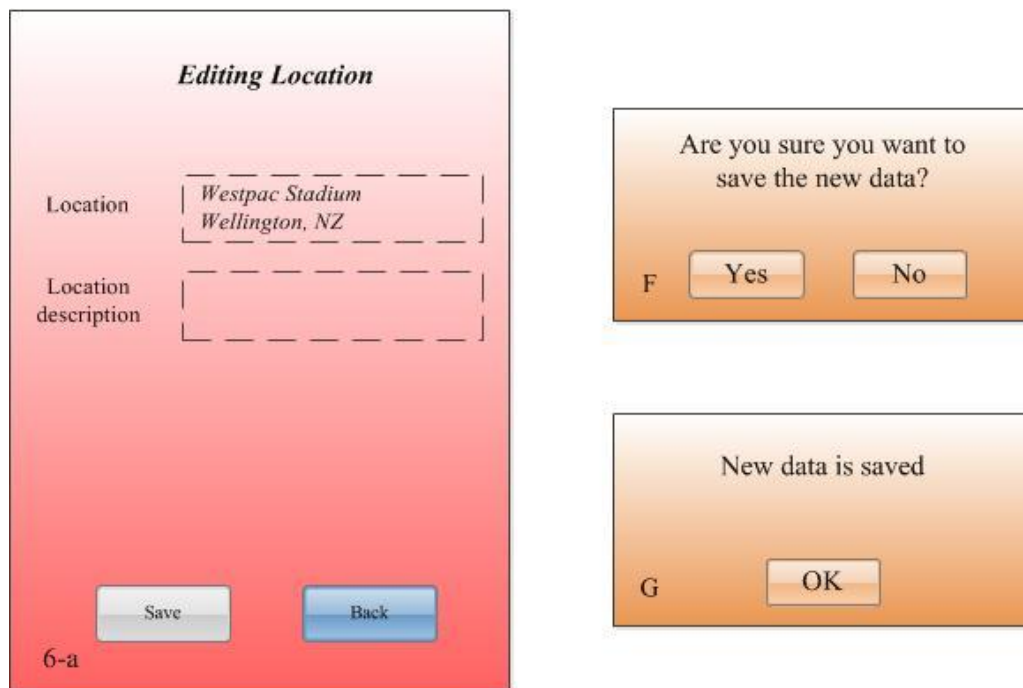


Figure 4.13: Editing location is available in *Page 6: Editing Location*. The associated message boxes are shown at the right side of the figure.

If the user finishes modifying the location of the bookmarked moment and wishes to save the changes that they did, the user can press the *Save* button. Before saving the changes, CatchIt ensures that the new information is really desired for saving. Thus, *Message Box F* is popped up. It asks the user if they want to save the new data. If the user wishes to continue with saving, the user will press *Yes* and then the *Message Box*

G will pop up to confirm the saving operation (see Figure 4.14, it shows *Page 6: Editing Location* after adding the location description). Otherwise the user will press *No* button on the same message box to back to *Page 6: Editing Location*.

The image shows a screenshot of a mobile application interface titled "Editing Location". The background is a red-to-white gradient. At the top, the title "Editing Location" is centered. Below the title, there are two text input fields. The first field is labeled "Location" and contains the text "Westpac Stadium Wellington, NZ". The second field is labeled "Location description" and contains the text "The place of the match between NZ Vs. Bahrain". At the bottom of the form, there are two buttons: "Save" and "Back". In the bottom left corner, there is a small label "6-b".

Figure 4.14: *Page 6: Editing Location* after adding a description of the bookmarked moment.

Text

CatchIt does not attach any textual note of the bookmarked moment automatically. Nevertheless, because the user may prefer to add some textual information about the bookmarked moment, CatchIt allows the user to add the textual note manually. CatchIt offers taking a textual note in *Page 7: Editing Text* which is accessible from *Page 3: Editing Moment*. There are one text field to type the note and two buttons *Save* to save the typed notes and *Back* to undo to *Page 3: Editing Moment* (Figure 4.15). The *Save* button is available once the changes have happened to the text field of *Page 7: Editing Text*.

In *Page 7: Editing Text* the user types their textual notes in the text field and then presses the *Save* button to save their notes. At that time, *Message Box F* pops up to ensure that the user really wishes to save the required information. Then the user makes a decision to save what they typed or cancel the operation of saving. If the user wishes to continue with saving, the user will press *Yes* which is located in *Message Box F* then *Message Box G* will pop up with the feedback about the last operation the user has done, saving new information (see Figure 4.16 which shows an example of a textual note in *Page7: Editing Text*). After that the user will be moved again to *Page3: Editing Moment*.

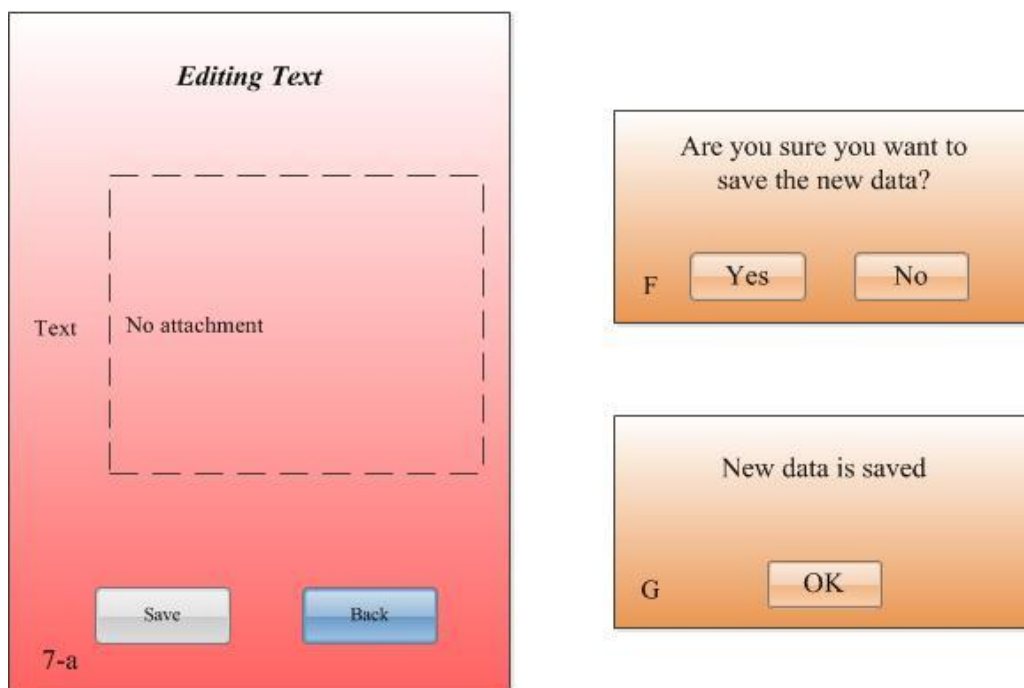


Figure 4.15: Editing text is available in *Page7: Editing Text*. The associated message boxes are shown at the right side of the figure.

In case the user has decided not to save the new textual notes, the user will press the button *No* in the *Message Box F*. After that the user will be moved back to *Page7: Editing Text* without any changes.

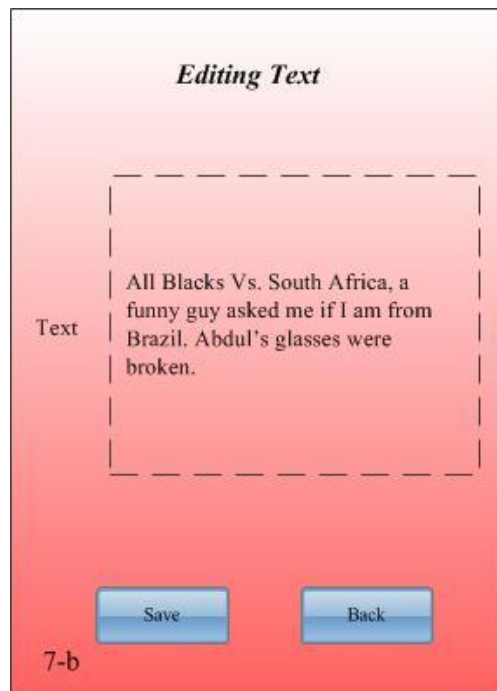


Figure 4.16: *Page7: Editing Text* after adding textual notes to be attached to the bookmarked moment.

Photo

The user of CatchIt can capture one photo or more and associate a caption to each photo from *Page8: Editing Photo*. The user can access *Page8: Editing Photo* if they selected the fifth item of *Page3: Editing Moment, Photo*, to be edited. The page contains an image to show the associated photos of the bookmarked moment with the ability to navigate the captured photos. Moreover, there is a text field to insert caption of the photo, two available buttons, *Add photo* and *Back* and one not available button *Save* (Figure 4.17).

Because there was no added photo when the user bookmarked the required moment, *Page8: Editing Photo* has no photo and no caption as well. The user however is

allowed to add photo and then the user is allowed to insert a caption of the captured photo.

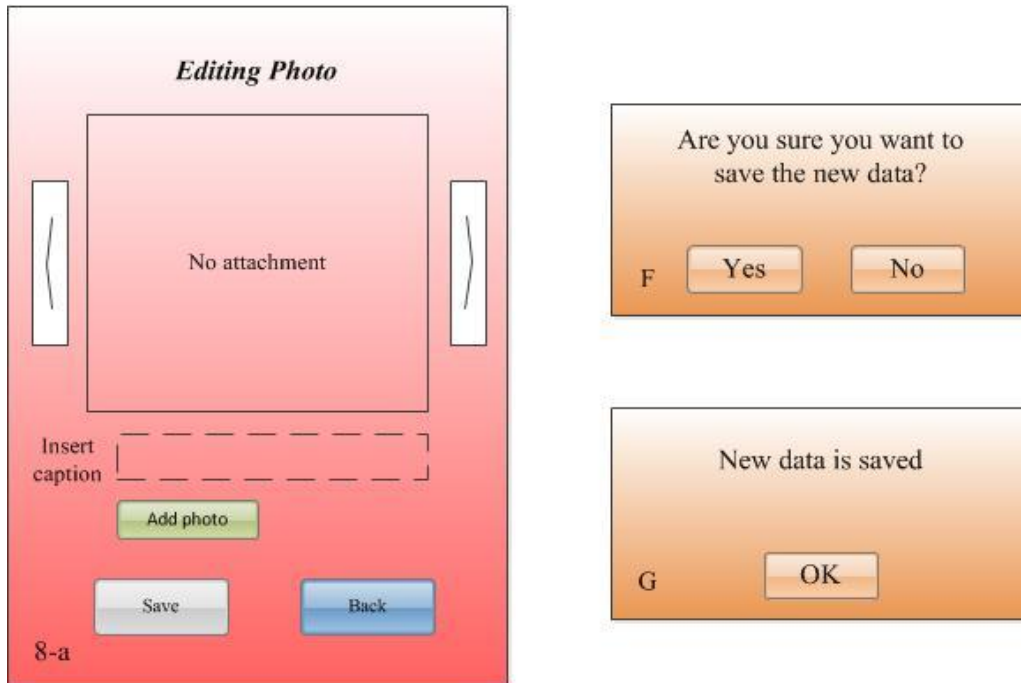


Figure 4.17: *Page8: Editing Photo* allows the user editing the photo of the bookmarked moment. The right side of the figure shows the message boxes that will pop up for the user according to their action on *Page8: Editing Photo*.

To add a photo the user should press the button *Add photo* to get access to the built-in camera in *Page9: Capturing Photo* (see Figure 4.18. The built-in camera is accessible to take a photo to add it to the bookmarked moment). After capturing the target, the user will be returned to *Page8: Editing Photo* to insert a caption of the captured photo if they wish and to save the new data (the photo and the caption or just the photo). In *Page9: Capturing Photo*, if the user retreats to take a photo, they can go back to *Page8: Editing Photo* by pressing *Back* button.

When the user captures the requested photo, they return to *Page8: Editing Photo* and the photo taken will appear in the middle of the page _ if the user took a photo. (See Figure 4.19 which shows *Page8: Editing Photo* after adding both photo and caption).



Figure 4.18: *Page9: Capturing Photo* has an access to the built-in camera to capture a photo of the bookmarked moment.

The user can save the new information by pressing the *Save* button which becomes activated after adding and changing the data of this view, and then *Message Box F* and *Message Box G* will pop up sequentially. *Message Box F* is for ensuring that the user really wishes to save the new data and *Message Box G* is for giving the user the feedback of capturing the new data. From *Message Box F* the user can go back on their decision in saving the new data by pressing the button *Back* (see the associated message boxes of *Page8: Editing Photo* in Figure 4.19).



Figure 4.19: *Page8: Editing Photo* after adding a photo with a caption.

Video

Recording video is one of the capturing types of moment cues that CatchIt offers to the user. The user can record a part of the bookmarked moment in a video format and attach that video to the moment. *Page10: Editing Video* of CatchIt is a page which manages the video recording. *Photo10: Editing Video* is reached when the user selects *Video record* from *Page3: Editing Moment*.

Page10: Editing Video consists of a video view, media button control, video slide bar and text field to insert the caption of the video. *Page10: Editing Video* has also three buttons, *Add video*, *Back*, and *Save* (Figure 4.20).

As a result of the absence of a video record in *Page10: Editing Video* after bookmarking the target moment, video slide bare and the play button and *Save* are not

enabled. However, the user can capture part of the target moment in video format by pressing *Add video*.

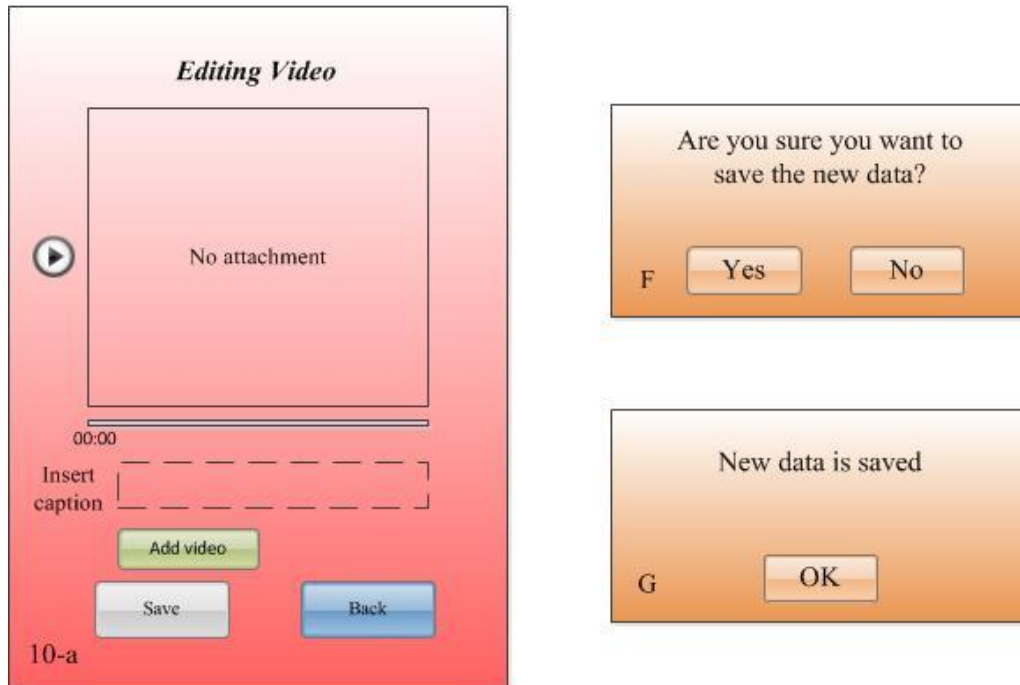


Figure 4.20: Editing video is allowed in *Page10: Editing Video*. The right side of the figure shows both message boxes *Message Box F* and *Message Box G* that will pop up for the user according to their action toward *Page10: Editing Video*.

Add video transfers the user from *Page10: Editing Video* to *Page11: Capturing Video*. Starting capturing a part of the bookmarked moment and stop capturing by using the built-in camera are available in *Page11: Capturing Video* (see Figure 4.21. The built-in camera is accessed and the recording video is available). After capturing the target piece of the bookmarked moment, the user can use the recorded video in *Page10: Editing Video* by pressing the button *Use* in *Page11: Capturing Video* and add a caption for that video in the text field of *Page10: Editing Video* (see Figure 4.22, the captured video is used in *Page10: Editing Video*). Otherwise, the user can skip using the captured video by pressing the button *Back* to move back again to *Page10: Editing Video*.



Figure 4.21: *Page11: Capturing Video* has an access to the built-in camera to capture a piece of the bookmarked moment in a video format.

The user also can play and pause the recorded video from *Page10: Editing Video* and see the duration of the recorded video under the video slide bar. In addition, saving the recorded video and its caption are enabled at this time because there is data to be saved. Moreover, the user can capture more parts of the bookmarked moment by pressing *Add video*. To save the all of the new data of video recording, including the video and the caption, the user presses the *Save* button which triggers *Message Box F*. *Message Box F* asks the user if they are sure about saving the new data. If the user presses the button *Yes*, the new data will be saved or else the new data will not be saved. Once the new data has been saved, *Message Box G* displays to let the user know that the new data is saved and then the user will go back to *Page3: Editing Moment* (see Figure 4.20, it has the contents of the message boxes of *Page10: Editing Video*).



Figure 4.22: *Page10: Editing Video* after adding a video with a caption.

Audio

An audio record is a way of capturing parts of the bookmarked moment in an audio format that CatchIt supports in its view, *Page12: Editing Audio*. The user can reach *Page12: Editing Audio* by selecting “Audio record” from the list in *Page3: Editing Moment* to edit it.

Page12: Editing Audio consists of a media control button, an audio slide bar and a text field for insertion of a caption of the recorded audio. *Page12: Editing Audio* has also three buttons *Add audio*, *Back* and *Save* (Figure 4.23).

The inactivated play button, audio slide bar and the *Save* button mean that there is no audio record to be attached with the chosen bookmarked moment. On the other hand, *Add audio* button is in active mode, is the way to record part of the bookmarked

moment in an audio format. Once the user presses that button the *Page13: Capturing Audio* will appear (see Figure 4.24, the user can record an audio and stop recording).

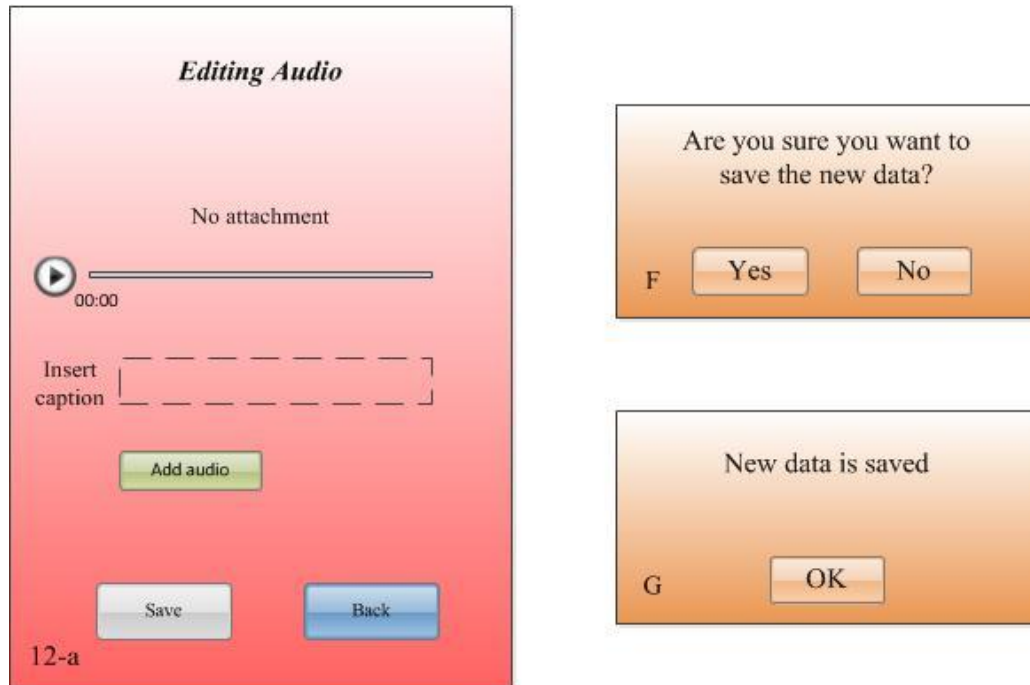


Figure 4.23: Editing an audio is allowed in Page12: Editing Audio. The right side of the figure shows the associated message boxes of Page12: Editing Audio.

Page13: Capturing Audio gives the user the ability to capture the target moment by recording it in an audio format in addition to showing the length of the recording. Then the recorded audio will be used by pressing *Use* button. After pressing *Use* button, CatchIt will move the user to *Page12: Editing Audio*. Otherwise the user will go back to *Page12: Editing Audio* (see Figure 4.25. This figure shows the activated media button as well as the audio slid in *Page12: Editing Audio* when a new audio record has been added).

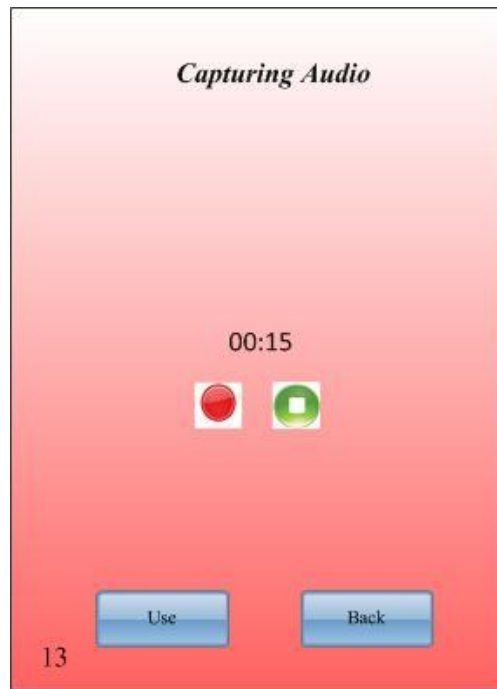


Figure 4.24: Recording audio is allowed in *Page13: Capturing Audio* and the audio is ready to be attached to the bookmarked moment.

When the user uses the recorded audio and goes back to *Page12: Editing Audio*, *Page12: Editing Audio* will look like Figure 4.25. The media button, audio slide bar and the *Save* button are activated at this time. In addition the user is allowed to insert a caption to the recorded audio and save all new data by pressing the button *Save*. When the user indicates that they wish to save the new data, *Message Box F* pops up to ask the user if they are sure about saving that data. Then *Message Box G* will pop up if the user is sure about saving the new data. From *Message Box G* the user will go back to *Page3: Editing Moment* once they see the message and press the button *OK* of that message box.

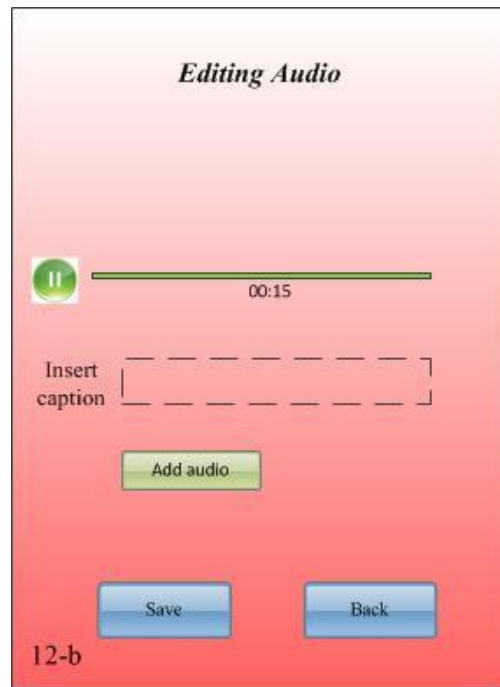


Figure 4.25: *Page12: Editing Audio* after adding a recorded audio.

People

People is one of the three contexts CatchIt deals with (the other contexts are time and location) of the target moment. People list of the target moment is captured manually by adding people names from the user contact list, in contrast of capturing time and location contexts which are captured automatically. The page of people list in CatchIt is presented in *Page 14: Editing People* (see Figure 4.26, there is no name in the people list but the user is enabled to add peoples' name from different page) while the page of selecting people from the contacts list is presented in *Page 15: Contacts* (see Figure 4.27, the contacts list is transferred from the user's phone to the system).

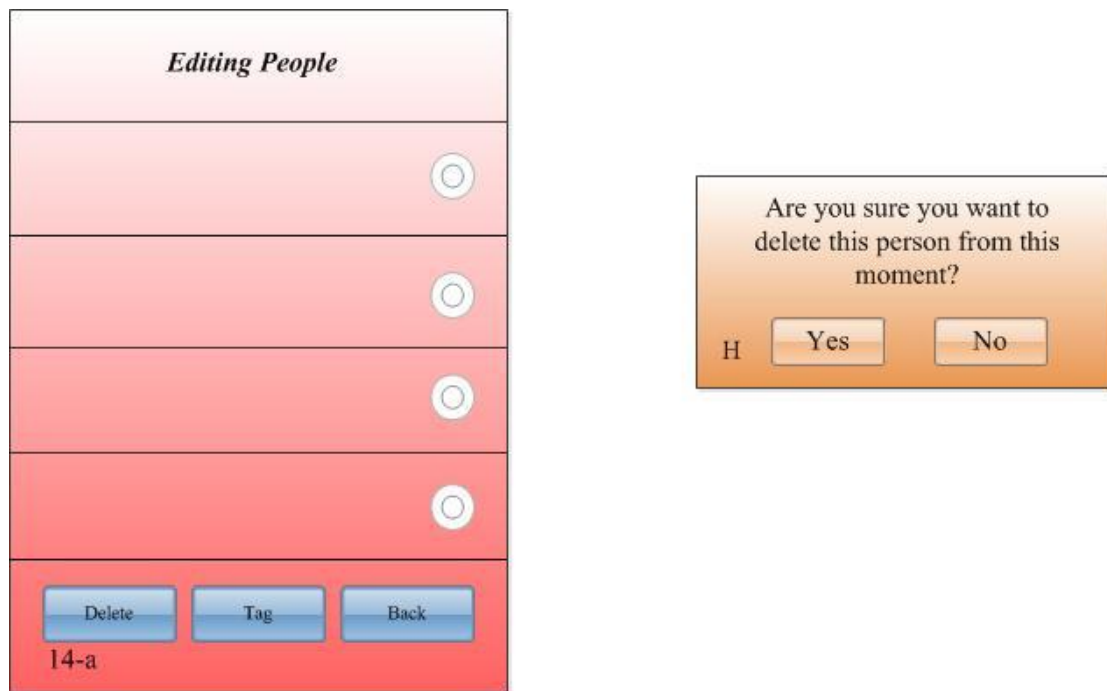


Figure 4.26: On the left side of the figure the initial view of Page 14: Edit People is shown. The check box control and the button Delete are not available as long as there is no name in the list. On the right side of the figure the associated message box Message.

The user can reach *Page 14: Editing People* from *Page 3: Editing Moment* by selecting the option *Tag people* and then by pressing *Edit*. In *Page 14: Editing People*, the user can add peoples' names to the people list (which initially does not have any name) as well as be able to delete any person's name from the list (if there is any name).



Figure 4.27: Contacts list of the user's phone is fetched to Page 15: Contacts. The user can select more than one name to be added to the bookmarked moment.

Adding one person's name or more is enabled by pressing the button *Tag* of Page 14: *Editing People*. That action will move the user to Page 15: *Contacts* that fetches the contacts list of the user from the existing list in the user's phone. From the contact list the user can select one name or more to be transferred to the people list.

The check box control of Page 14: *Editing People* is not available, nor is the button *Delete* in case there are no names at all in the list. However, if there is at least one name in the list as shown in the Figure 4.28, the check box control and the button *Delete* will be available. Consequently, the user can delete any checked name from the list. The response of CatchIt when the user deletes a person's name from the people list, is to provide a pop-up *Message Box H*. *Message Box H* asks the user if they really wish to delete the selected person's name from the list. Then the user can

click on *No* to ignore the deletion operation or keep going by pressing *Yes* in that message box.



Figure 4.28: *Page 14: Editing People* after adding people's names to the people list.

The View of CatchIt after Revising

After adding a collection of new data (moment's name, time, location, text note, photo, video and audio) into the bookmarked moment or modifying the existing ones, *Page3: Editing Moment* will look like (Figure 4.29). From *Page3: Editing Moment* the user can go back again to *Page2: Bookmarked Moments* and then from *Page2: Bookmarked Moments* to the main page *Page1: The Main Menu* (see Figure 4.30 which shows the list of the bookmarked moments in *Page2: Bookmarked Moments* after associating the moment's name to the moment).

<i>Editing Moment</i>		
Moment's name	<i>New experience new...</i>	<input type="radio"/>
Time	<i>30.July.2011 08:00 PM</i>	<input checked="" type="radio"/>
Location	<i>Westpac Stadium, Wel...</i>	<input type="radio"/>
Text	<i>No attachment</i>	<input type="radio"/>
Photo	<i>See photos</i>	<input type="radio"/>
Video Record	<i>Watch video</i>	<input type="radio"/>
Audio Record	<i>Listen to audio</i>	<input type="radio"/>
Tag people	<i>Tagged people</i>	<input type="radio"/>
<div style="text-align: center;"> <input type="button" value="Delete"/> <input type="button" value="Edit"/> <input type="button" value="To recent moments"/> </div>		
3-b		

Figure 4.29: The appearance of *Page3: Editing Moment* after adding or modifying a collection of capturing types.

<i>Bookmarked Moments</i>		
30.July.2011 08:00 PM	<i>First experience with Ruby</i>	<input checked="" type="radio"/>
01.July.2011 11:46 PM	<i>No Name</i>	<input type="radio"/>
13.May.2011 10:18 AM	<i>No Name</i>	<input type="radio"/>
23. February.2011 07:33 PM	<i>No Name</i>	<input type="radio"/>
<div style="text-align: center;"> <input type="button" value="Delete"/> <input type="button" value="Edit"/> <input type="button" value="To main menu"/> </div>		
2-b		

Figure 4.30: The moment's name is added to the first bookmarked moment in *Page2: Bookmarked Moments*.

Synchronize CatchIt with Digital Parrot

Translating the bookmarked moments to a desktop application Digital Parrot is a significant step in terms of retrieving those moments. Digital Parrot visualizes the retrieval information in linked nodes. Additionally this application has several methods of searching for the target information such as searching by time, location and people.

CatchIt synchronizes with Digital Parrot and transfers the bookmarked moments to Digital Parrot from *Page 1: The Main Menu*. The user presses the button *Synchronize* in that page and then CatchIt will transfer the user's bookmarked moments to Digital Parrot. Once the bookmarked moment is translated to Digital Parrot, message box *Message Box 1* will pop up (see Figure 4.31).



Figure 4.31: *Message Box 1* informs the user that the transferring data from CatchIt to Digital Parrot has been done.

4.4 Expert Walk-through

Walk-through the initial user interface prototype of CatchIt will help us identify the usability and functionality problems of the initial user interface prototype by covering three areas of that prototype. These areas are pages, message boxes and events.

Firstly, the heuristics of pages will include: page itself (inappropriate/ missing pages), contents (inappropriate/ missing contents) and titles (inappropriate titles). Secondly,

heuristics of events will include: event itself (inappropriate/ missing events). Thirdly, the heuristics of message boxes will include: message box itself (inappropriate/ missing message boxes), contents (inappropriate contents) and titles (missing) (see Table 4-1).

Table 4-1: Heuristics of the walk-through that the experts will test the system by. The study involves pages, events and message boxes of the initial user interface by testing the area itself, content, controls, and title. The white cell means that the item of heuristics is covered because it was existing in the initial user interface prototype. Each of covered item will be identified inside the white cell. Otherwise, the item will not be covered in this study (the gray cells).

			User Interface		
			Pages	Events	Message Boxes
Heuristics	Itself	In-appropriate	P(1)	E(1)	M(1)
		Missing	P(2)	E(2)	M(2)
	Content	In-appropriate	P(3)		
		Missing	P(4)		
	Title	In-appropriate			M(3)
		Missing	P(5)		

4.4.1 Methodology

The walk-through will involve three experts (E1, E2 and E3) with background of using a very wide range of mobile phone applications. They will be asked to walk through the initial user interface prototype to find usability and functionality problems

according to the heuristics that we have mentioned. Their feedback will be anonymous and will be covered in our final user interface prototype to achieve the goal of this walk-through.

4.4.2 Result

As a result of walk through CatchIt will be divided into three parts: pages, events and message boxes. This section will explain each identified problem of the initial user interface prototype of the system according to those areas in turn.

4.4.2.1 Pages

Pages of CatchIt have different kinds of content such as text boxes and buttons in addition to identified titles. The pages had some usability and functionality problems that the experts had found when they walked through the initial user interface prototype. The problems were in the pages themselves, the content of some pages and the title of some pages.

Page Itself: Inappropriate Pages (P1)

Selecting people from the contact list is an inappropriate way to tag people in the bookmarked moment. Instead, there should be another list of people who were tagged in previous bookmarked moment. As a result *Page 15: Contacts* has to be replaced by another page.

Page Itself: Missing Pages (P2)

1. A page to show the moment's location on a map. It is useful to see the location of the bookmarked moment on a map and no page has this option. The new page will be placed between *Page 3: Editing Moment* and *Page 6: Editing Location* which means that the new page will be before the user is going to edit the location of the bookmarked moment. The new page should enable the user to move backward to the previous page and also to make the user go to the page of editing the target location. The new page will be called *Page 3-6: Location on Map* (see Figure 4.32).

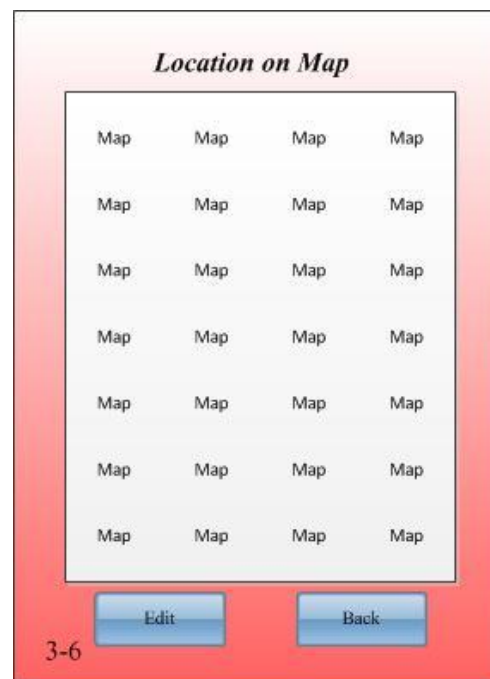


Figure 4.32: Page 3-6: Location on Map is a new page of CatchIt. It is placed between Page 3: Editing Moment and Page 6: Editing Location. The aim of this page is to show the location of the bookmarked moment on a map.

As a consequence of showing the target location in three different formats, CatchIt matches more of the user needs toward the location-context. All experts (E1, E2 and E3) agreed that CatchIt should the target location in

different ways. That opinion supports Domnitcheva (2001) when she said that each location model is suitable for particular aims of the application. In addition, showing the location on map was one of the system suggestions of Allalah (2011) that we did not cover in our requirements (see Chapter 3).

2. A page to show a list of photos. It is useful to look at all the related photos of the target bookmarked moment in one glance. Accordingly, there is a missing page between *Page 3: Editing Moment* and *Page 8: Editing Photo*. The new page has to hold all related photos of the target moment and it should have the ability to go backward to the previous page, add more photos to the list and delete or edit the selected photo from the list. We will call the new page, *Page 3-8: Photos* (see Figure 4.33).

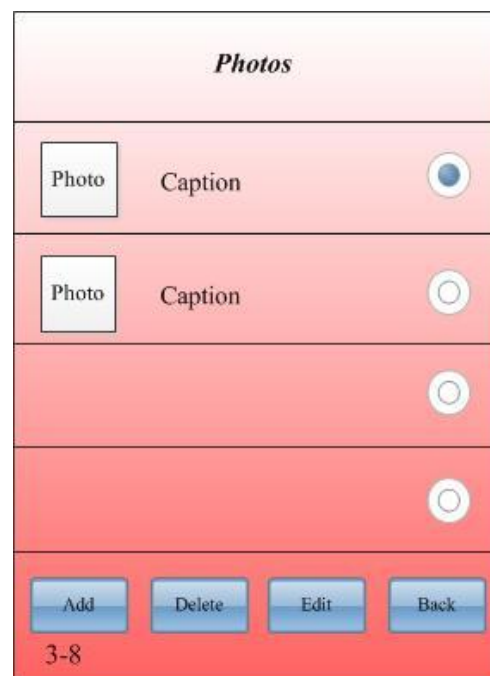


Figure 4.33: New page, *Page 3-8: Photos* is placed between *Page 3: Editing Moment* and *Page 8: Editing Photo* with a slight change of the order of viewing pages.

The order of viewing the pages will be changed slightly after adding *Page 3-8: Photos*. When the user wishes to add more photos to the bookmarked moment, the user will be moved to *Page 9: Capturing Photo*. After capturing the target photo, the user will be moved to *Page 8: Editing Photo*. On the other hand the user can reach *Page 8: Editing Photo* if they wish to edit the selected photo from *Page 3-8: Photos*.

3. A page to show a list of videos. It is useful to know all the related videos of the target bookmarked moment are in one place. So, the missing page is between *Page 3: Editing Moment* and *Page 10: Editing Video*. The new page has to view the captions of all videos because the caption is a good reference to the associated video in addition to an order number. Also, the new page has to have the ability to go backward to the previous page as well as the ability of deleting the selected videos. Moreover, this page should be able to add more videos to the video list. *Page 3-10: Videos* will be the name of the new page (see Figure 4.34).

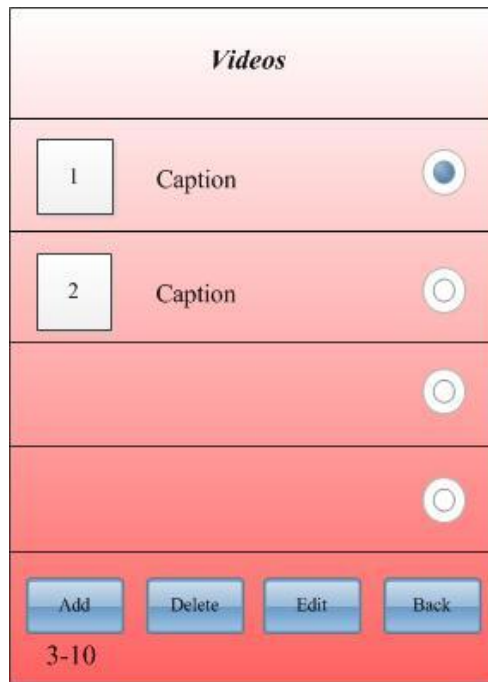


Figure 4.34: *Page 3-10: Videos* is a new page which lists the captions of the all related videos of the bookmarked moment in a table.

As a result of adding *Page 3-10: Videos* the movement among the pages will be changed. From *Page 3: Editing Moment* the user reaches *Page 3-10: Videos* rather than reaching *Page 10: Editing Videos*. However, *Page 10: Editing Videos* can be reached if the user wishes to edit one video from the list of videos in *Page 3-10: Videos* but if the user wishes to add more videos to the video list, the user will be moved to *Page 11: Capturing Video*.

4. A page to play a video in a full screen mode. Because the area of playing the videos in *Page 10: Editing Video* is small, playing the video on a full screen is required (see Figure 4.35). The new page will come after *Page 10: Editing Video* when the user decides to watch the video by pressing the play button. The user can go backward to the previous page and play and stop the video. This new page will be called *Page 10.5: Play Video*.



Figure 4.35: Watching the recorded video is available in *Page 10.5: Play Video*.

5. A page to show a list of audios (see Figure 4.36). It is useful to know all the related recorded audios of the target bookmarked moment are in one place before going to edit or to review one of audio items. So, the missing page is between *Page 3: Editing Moment* and *Page 12: Editing Audio*. The new page has to view the captions of all recorded audios because the caption is a good reference to the associated audio. In addition, the new page has to have the ability to go backward to the previous page, delete the selected audios and add more audios to the audio list. We will refer to this new page by *Page 3-12: Audios*.

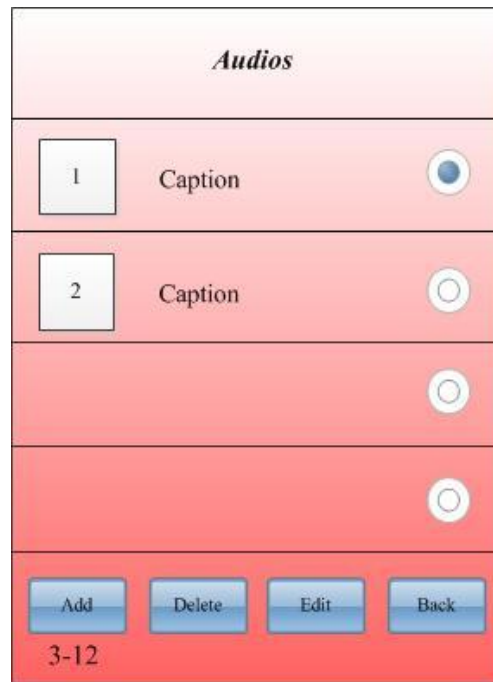


Figure 4.36: The list of recorded audios with their captions is shown in *Page3-12: Audios*.

Adding *Page 3-12: Audios* will affect on the sequence of the movement among pages. Firstly, the user will reach this page from *Page 3: Editing Moment*. Then from the new page, *Page 3-12: Audio* the user can record a new audio from *Page 13: Capturing Audio*, edit the selected recording audio from *Page 12: Editing Audio*.

6. A page to show a list of all tagged people. As it was mentioned in (P1), *Page 15: Contacts* is inappropriate page and there is a need to add a new page which does the same function, tagging people, but in an appropriate way. The new page will have got a list which will hold every person who is tagged in any bookmarked moment. Then the user can use this list later to tag people in the new bookmarked moments. The new page will be identified by the title *Page 15: People* (see Figure 4.37). In addition to selecting people from the list

in the new page, the user must have the ability to add a new person to this list if the target person has not been listed yet. Additionally, the user can access the previous page and delete any person from the existing list.



Figure 4.37: All tagged people in all previous bookmarked moments are listed in *Page15: People*.

7. A page to add a person into the people list. As was mentioned in the previous point, adding a new person if that person should be supported in CatchIt. So, this function will be addressed in a new page, *Page 16: Person* (see Figure 4.38). This page will allow the user to add a new name of the target person and save it. In addition, the user can go back to the previous page.

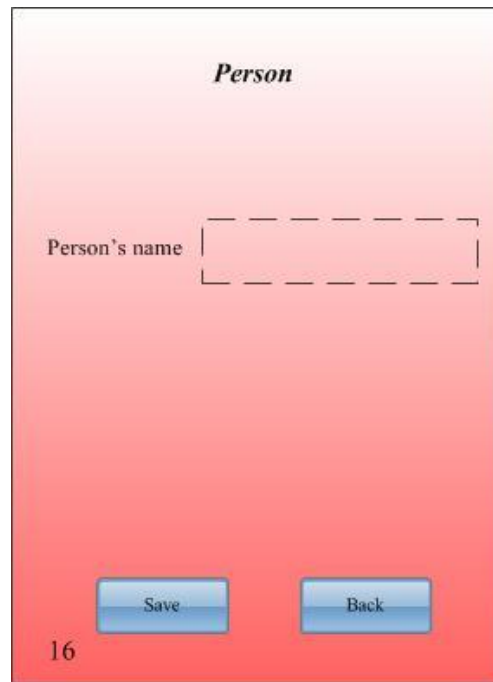


Figure 4.38: A new person can be added from *Page 16: Person* so then this name will be shown in the list of people's name in *Page 15: People*.

Contents: Inappropriate Contents (P3)

1. *Page 1: The Main Menu*: this page has brief information about the application. The label, which contains this information, is not useful particularly when the user is familiar with the system. Instead, this label could be used to introduce the application before installing the application (the new view of *Page 1: The Main Menu* is shown in Figure 4.39). In case of the user study, the brief information about CatchIt will be attached to the participants of the study. On the other hand, in case of installing the system via the market or store of the mobile application, the user can read blurb about CatchIt before installing it.

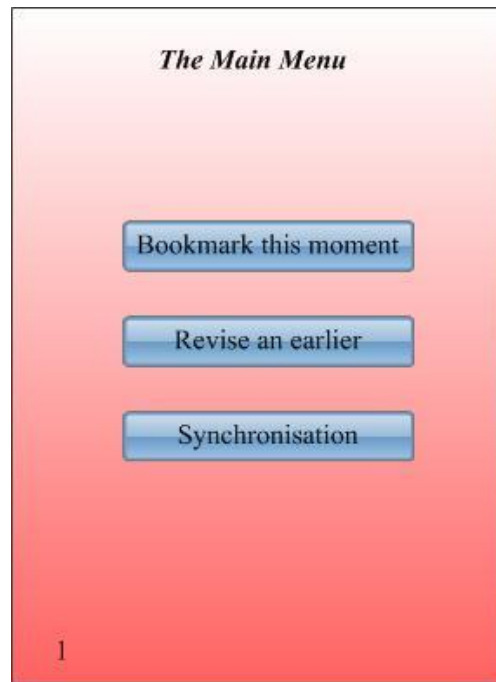


Figure 4.39: The new view of *Page 1: The Main Menu*. The brief information about the system is removed from the page.

2. As a result of inserting the page, *Page 15.5: Play Video* (P2.4) the contents of *Page 10: Editing Video* will be split into several separate pages. The slide bar, the current time of the video and the video view will be in *Page 10.5: Play Video* whereas *Add video* button in *Page 10: Editing Video* will be moved to *Page 3-10: Videos* (see Figure 4.34). So, the play button and save button and the text box of inserting the caption will be in the same page, *Page 10: Editing Video* (see the new view of *Page 10: Editing Video* in Figure 40).

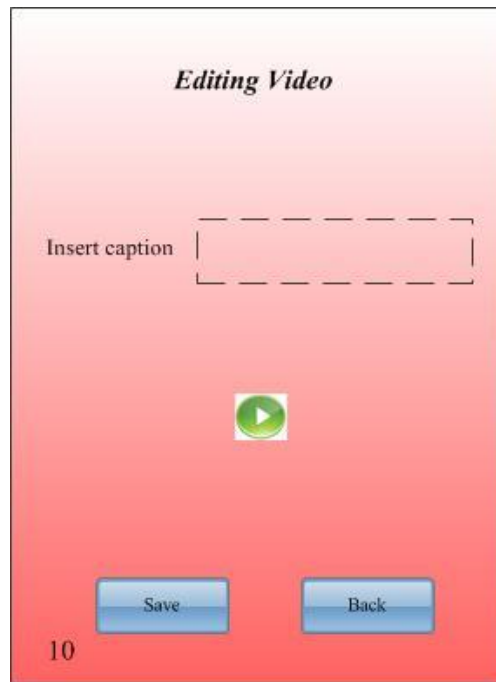


Figure 4.40: *Page 10: Editing Video* after changing its contents. The remaining contents are the text box for inserting caption of the recorded video and play, save and back buttons.

3. In various pages there are some inappropriate labels of the text boxes. These labels do not explain the requested inserted text. In *Page 4: Editing Moment's Name* the label *Moment's name* does not explain the reason of existing the associated text box as much as *Insert or change the label of the moment*. *Page 5: Editing Time* has the label *Time description*. It should be changed to *Insert or change the time of the moment*. *Page 6: Editing Location* has the label *Location description* which should be replaced by *Insert or change the location of the moment*. *Page 7: Editing Text* has the label *Text* which needs to be clarified by using the label *Your Notes*. *Page 8: Editing Photo*, *Page 10: Editing Video* and *Page 12, Editing Audio* have the same label *Insert caption*. It is recommended to use *Insert or change the label of X* since X is photo, video or audio.

Contents: Missing Contents (P4)

1. CatchIt used to show the location of the bookmarked moment in one format - Geographical location. However, after adding *Page 3-6: Location on Map* CatchIt has two ways of showing the target location. To increase the location models of the target location experts (E1 and E3) suggested changing the content of the third item of *page 3: Editing Moment*, Location, to be in the physical location model (latitude, longitude). As a result CatchIt will cover more needs of the user toward the location-context.
2. The user is allowed to type their notes in *Page 7: Editing Text*. However, the expert (E3) preferred to add a text box to assign a label of the notes. Thus the user can recognize their notes by adding a label to their notes (see Figure 4.41).

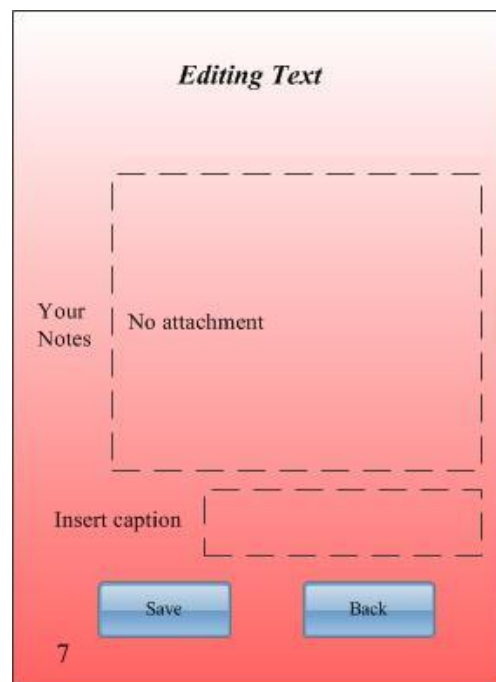


Figure 4.41: New text box to contain the caption of the attached text is added in *Page 7: Editing Text*.

3. In *Page 13: Capturing Audio* the user can capture a new audio file but the user cannot insert a caption of that recorded audio. Therefore, *Page 13: Capturing Audio* should allow the user to insert or modify a caption to the recording audio after recording it - the same as in *Page 12: Editing Audio* (see Figure 4.42).

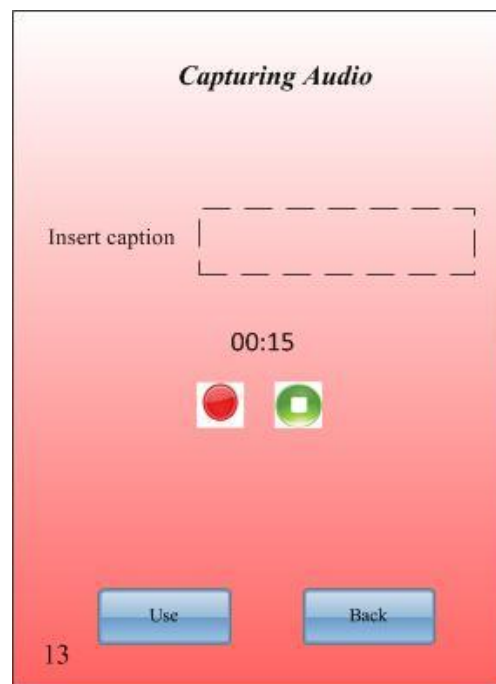


Figure 4.42: User can insert a caption for the recorded audio from *Page 13: Capturing Audio*.

Title: Inappropriate Title (P5)

The titles of some pages were either made difficult to distinguish the purpose of the page or the titles were too long. The unclear title was in *Page 7: Editing Text* and it was recommended that it be replaced with *Note*. Recommendations for the rest of the pages were: *Page 3: Editing Moment* to *Moment Details*, *Page 4: Editing Moment's Name* to *Moment Label*, *Page 5: Editing Time* to *Moment Time*, *Page 6: Editing Location* to *Moment Location*, *Page 8: Editing Photo* to *Photos*, *Page 10: Editing*

Video to Video, Page 12: Editing Audio to Audio and Page 14: Editing People to Tagged People.

4.4.2.2 Events

Events of CatchIt exist in pages, message boxes and the event between both pages and message boxes. However, there were inappropriate events and missing ones in the initial user interface.

Events Itself: Inappropriate Events (E1)

In *Page 3: Editing Moment* the user is allowed to delete the whole captured data of the same capturing ways such as deleting all photos or videos of the bookmarked moment. According to the experts E2 and E3, this deletion operation will not help the user to delete selective captured data. Thus, either the user will lose all associated captured data of a particular capture ways of the bookmarked moment or the user will keep unwanted captured data in the portable device, which means using up more storage space. Accordingly, the *Delete* button and the deletion operation in this page will be removed (P3.4). Instead, the deletion function will be distributed on *Page 3-8: Photos, Page 3-10: Videos* and *Page 3-12: Audios* (P4.4) to allow the user to delete one captured data or all captured data as the user wishes.

Events Itself: Missing Events (E2)

One of the functions of *Page 1: Main Menu* is to bookmark the moment. Once the user bookmarks the target moment *Message Box A* pops up to tell the user that the target moment has been bookmarked. Then the user moved back to the same page. Expert (E1) recommends moving the user to *Page 3: Editing Moment* instead of going

back to *Page 1: Main Menu*. Consequently, the user will have the ability to add or edit some contents of the recent bookmarked moment rather than pressing *Revise an earlier* in *Page 1: Main Menu*, going to *Page 2: Bookmarked Moment* and then finding the last bookmarked moment to modify that bookmarked moment.

4.4.2.3 Message Boxes

Message boxes inform the user before the user's interactions with the system. Moreover, these messages warn the user about the possible events when they intend to do something that could affect their needs in the future. For example, when the user intends to delete a particular bookmarked moment, CatchIt warns the user (by using message box) to tell them they will lose this moment.

Some message boxes were useless for the experts (E1, E2 and E3) when they walked through the initial user interface of CatchIt while others requested to be in the system. Additionally, there were no titles for the message boxes at all.

Message Box Itself: Inappropriate Message Box (M1)

1. *Message Box F* was used to make sure if the user wishes to save the new data. That message box was useless because the procedure of adding a new data takes more steps and it will not be done if the user does not really wish to do it. Therefore, the user does not need to be asked if they wish to save the new data. In addition, if the user goes back on their decision about saving this data, they can go back to the previous page without doing a further action.

2. As long as *Message Box F* will be removed the related message box (*Message Box G*) will be removed as well. *Message Box G* was used to tell the user that the new data has been saved.
3. *Message Box E* was used to tell the user that the changed date/time of the moment is not available. It popped up when the user inserted a forward date/time because the system does not accept the future date/time to be the moment's time. The recommendation of the walk-through was to remove this message box because there is possibility to prevent the user programmatically to set the date/time of the target moment to be in the future.

Message Box Itself: Missing Message Box (M2)

1. As a consequence of removing the deletion operation from *Page 3: Editing Moment* and distributing it to several pages, *Message Box C* has to be moved to those pages that the deletion operation is distributed to (*Page 3-8: Photos*, *Page 3-10: Videos* and *Page 3-12: Audios*).
2. There is no message box that pops up from the new page, *Page 15: People* if the user wishes to delete people from the people's list. Thus, *Message Box J* will be added to this page. It ensures if the user really wishes to delete the selected person from all moments and also from the people's list (see Figure 4.43).



Figure 4.43: *Message Box J* pops up when the user wishes to delete a person from the list in *Page 15: People*.

Titles: Missing Titles (M3)

All message boxes of the system did not have titles. So, the suggested titles were:

- *Bookmarked Moment* for *Message Box A*.
- *Delete Bookmarked Moment* for *Message Box B*.
- *Delete Captured Data* for *Message Box C*.
- *Invalid Date/Time* for *Message Box D*.
- *Delete a Person from Moment* for *Message Box H*.
- *Synchronized* for *Message Box I*.
- *Delete a Person from All Moments* for *Message Box J*.

Message Box E, *Message Box F* and *Message Box G* were removed from the system.

4.5 The Final Prototype

The recommendations of the expert walk-through affected the sequence of pages and message boxes in the initial user interface of CatchIt. As a result, the overview prototype has to be adjusted in accordance with the changes that happened. Figure 4.44 shows the changes that have done on the overview prototype.

In the pages perspective there are six new pages that have been added:

- *Page 3-6: Location on Map* allows the user to move from *Page 3: Moment Details* to *Page 6: Moment Location*, and vice versa.
- *Page 3-8: Photos* allows the user to move from *Page 3: Moment Details* either to *Page 8: Photos* or to *Page 9: Capturing Photo* and then go back to *Page 3: Moment Details*. Also, from new page, *Page 3-8: Photos*, *Message Box C: Delete Captured Data* pops up if the user wishes to delete one of the captured photo and then from this message box the user comes back to *Page 3-8: Photos*.
- *Page 3-10: Videos* allows the user to move from *Page 3: Moment Details* either to *Page 10: Video* or to *Page 11: Capturing Video* and then go back to *Page 3: Moment Details*. Also, *Message Box C: Delete Captured Data* pops up from *Page 3-10: Videos*. This message box returns the user to *Page 3-10: Videos*.
- *Page 10.5: Play Video* is reached from *Page 10: Video*. From *Page 10.5: Play Video* the user can move back to *Page 10: Video*.
- *Page 3-12: Audios* allows the user to move from *Page 3: Moment Details* either to *Page 12: Audio* or to *Page 13: Capturing Audio* and then come back to *Page 3: Moment Details*. Also, *Message Box C: Delete Captured Data* pops up from *Page 3-12: Audios* and from this message box the user is moved back to *Page 3-12: Audios*.

- *Page 16: Person* allows the user to move from *Page 15: People*, and vice versa.

In addition, there was one replaced page, *Page 15: People* instead of *Page 15: Contacts*. There were no any pages that were dropped and not replaced with another one.

On the other hand, in the message box perspective a new message box has been added: *Message Box J: Delete a Person from All Moments*. It pops up from *Page 15: People*". Additionally, *Message Box C: Delete Captured Data* has been removed from *Page 3: Moment Details* (the new name of *Page 3: Editing Moment*) and distributed to three other pages *Page 3-8: Photos*, *Page 3-10: Videos* and *Page 3-12: Audios*. As there have been new message boxes added, two message boxes have been dropped, *Message Box F* and *Message Box G*.

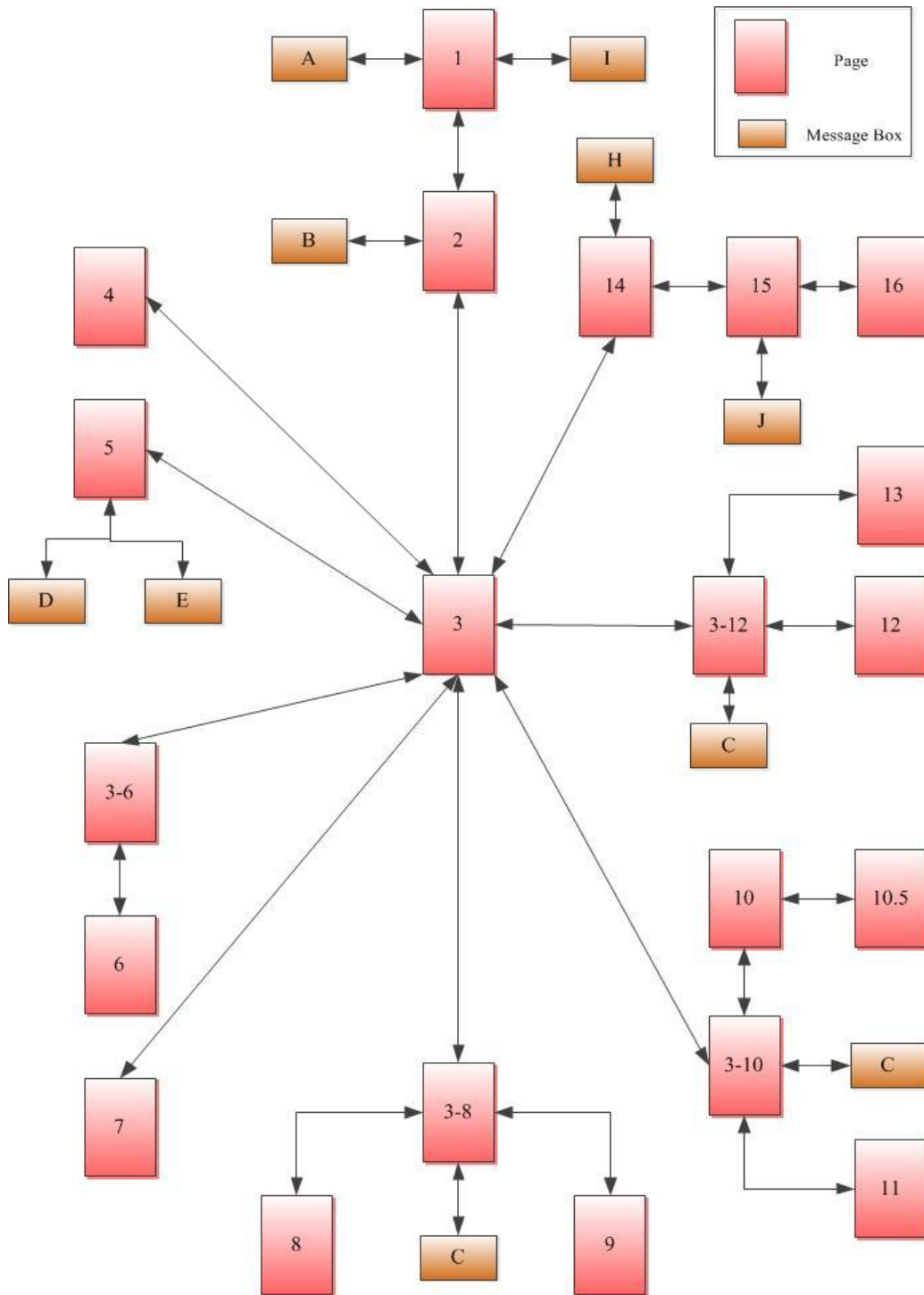


Figure 4.44: Adapted overview prototype according to the changes that have been completed after taking the recommendation of the expert walk-through into our account.

4.6 Summary

This chapter concentrated on the conceptual design of our system, CatchIt. First of all, the conceptual architecture explained how CatchIt links two different approaches, *bookmarking a moment* and *Digital Parrot*. To do so CatchIt has three main steps: bookmark a moment, revise an earlier moment and synchronize the mobile application (CatchIt) with the desktop application (Digital Parrot) enable the transfer of data from the former to the latter application.

After detecting the main steps of the system, the overview prototype of the user interface was shown and the initial user interface prototype was introduced. Some usability and functionality problems were detected in that version of the user interface when the expert walk-through method was applied as a test.

According to the recommendation of the expert walk-through method, many changes have happened to the user interface. New pages have been added and one page has been replaced by a much more appropriate one. Additionally, some contents of some pages were changed or added and almost all pages have been given new titles.

Furthermore, there were some changes on some events of the pages and there was a missing event that, when added, would make the system more useful. Moreover, message boxes had few changes as well. Some message boxes were dropped, some were distributed to different pages and some new ones added. In addition, all message boxes have been identified by appropriate titles.

Finally, the initial prototype was adjusted to match the recommendations of the experts in the walk-through and the final user interface was introduced. In the next chapter, we are going to implement the system according to the result of this chapter.

5. Implementation

In this chapter we will shed light on the implementation of our system, CatchIt. The conceptual architecture of the system takes account of the portable device, the user of the system, the local storage, the user interface, and also a way to synchronize with the desktop application, Digital Parrot (see Figure 5.1).

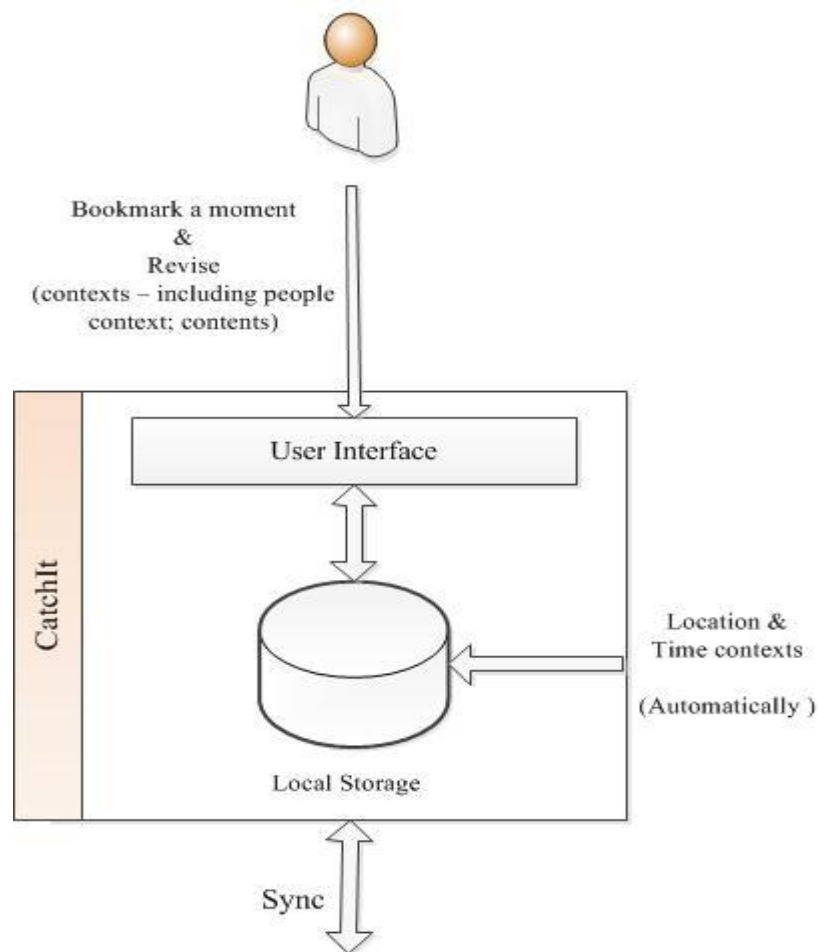


Figure 5.1: A diagrammatic view of a part of the conceptual architecture of CatchIt.

5.1 Portable Device

CatchIt will be implemented on a portable device to achieve the main goal of the system, bookmarking a moment. According to the requirements of the system in Chapter 3, the user should be comfortable using the portable device (NFR1), the portable device which should have appropriate size (NFR2), large storage capacity (NFR3), location device (NF4), digital camera (NF5), audio recorder (NF6), and should keep the personal information safe (NF8).

In general, smart phones are the most possible portable devices to be used in the implementation of CatchIt. The user of the system does not need to carry another portable device such as PDAs while using CatchIt. In addition, smart phones have a collection of functions such as making phone and visual calls, which make the usability of these phones more comfortable (NFR1).

The *iPhone* is a smart phone that was introduced by *Apple* (Apple, 2012c). It is ideal for implementing CatchIt due to its features and functions that cover the requirements of the portable device. iPhone has appropriate size (NFR2), several suitable storage capacities (NFR3), a built-in GPS (NFR4), a built-in digital camera (NFR5), and has the ability to record the surrounding environment in an audio format (NFR6). In addition, the user of iPhone can manage the location service of each application in the phone by allowing the phone to use their current location (NFR8). Moreover, as mentioned, regarding the importance of showing the target location on a map, iPhone has reasonable screen size, which ensures that it contains a map with an adequate view.

As a consequence of using the iPhone mobile, the software development kit of *iOS* is required (*iOS SDK*) (Apple, 2012b). The used version of *iOS SDK* is 5.0, which includes the project builder (*Xcode*), some simulators of Apple devices such as iPhone, and iPad (Apple, 2012b). *Xcode* supports writing by various programming languages; one of them is Objective-C which is being used to program our system.

Furthermore, we decided to work on *iOS SDK* rather than other software development kits such as *Java ME SDK* because we are more familiar with the ready to use built-in frameworks that the *iOS SDK* has, such as *MapKit* framework that allows us to deal with maps (Apple, 2011b) and *CoreLocation* framework that allows us to deal with locations and regions (Apple, 2011a).

5.2 Target User

The target user of our system is a person who needs some help to remember a selective of their moments. In addition to the ability of capturing selective contents of the moments by using supported capturing ways (taking notes, taking photos, recording video, recording audio), *CatchIt* assists the user to capture three contexts of the selective moments (time, location, people). After reviewing this collection of the cues, the user would come up with more details of the selected moments.

CatchIt is also a perfect application for the users who wish to keep their memory safe. There are neither gender nor age constraints for using *CatchIt* except that the user should have a particular smart phone, iPhone.

5.3 Local Storage of CatchIt

The local storage of CatchIt will be a database. It is useful to build a light database in the portable device for two reasons. The first reason is that saving information into database will keep the stored information in a safe place even if the portable device is switched off by mistake or by a malfunction of an application. On the other hand, the second reason is that the user will not be limited by the Internet connection, and thus, would be able to use CatchIt in a place with not Internet connections.

Designing the database of CatchIt will take account of entity relationship diagram in two versions: enhanced entity relationship diagram and relational database schema.

5.3.1 Entity relationship diagram (ERD)

The initial relational entity diagram for CatchIt will be introduced in this section in two versions. The first version which is the logical ERD will contain all sorts of relationships, whereas the second one, which is also known as the physical ERD, will deal with *many-to-many* relationships of the first version.

5.3.1.1 The First Version of ERD

The first version of the ERD (the logical ERD) of our system is shown in Figure 5.2. It can be seen that CatchIt has three relational entities; MOMENT, PEOPLE and CAPTURE, where the attributes of each entity has two relationships, *has* and *is captured*.

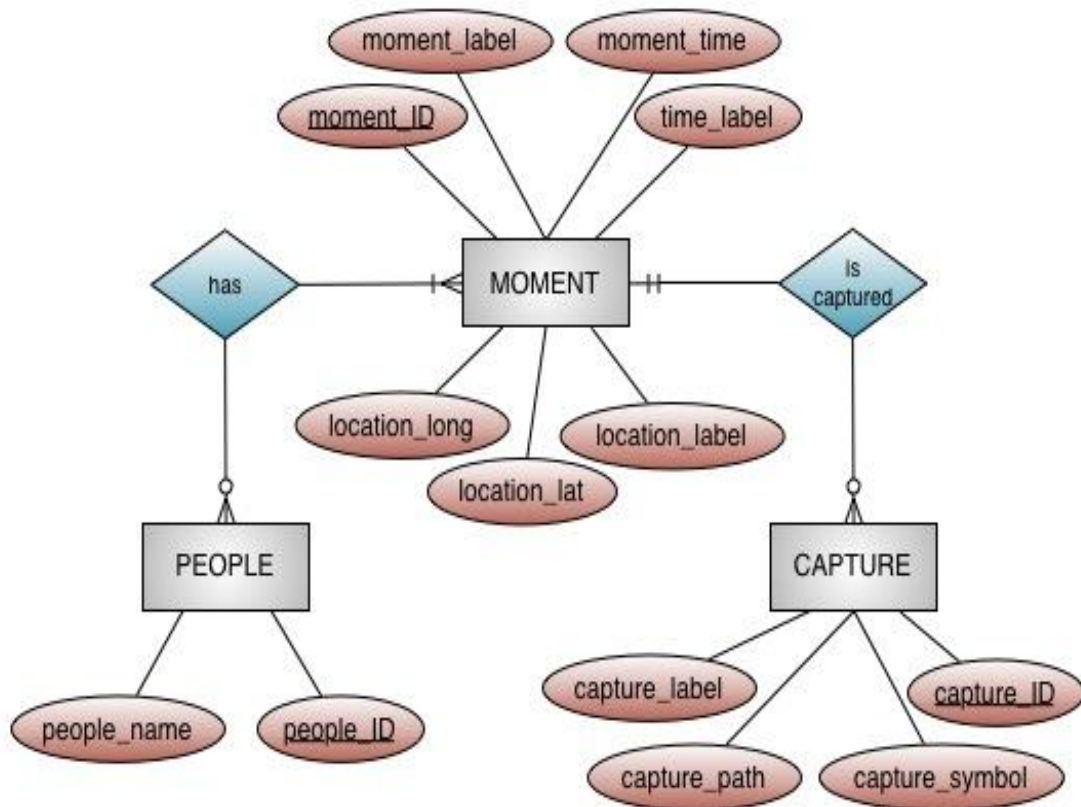


Figure 5.2: The first version of ERD of CatchIt.

Entities and Attributes

The MOMENT entity has a key attribute, *moment_ID*, and other six attributes, *moment_label*, *moment_time*, *time_description*, *location_lat*, *location_long* and *location_description* (see Table 5-1 for further information about on entities of the first version of the ERD and Table 5-2 for further information on both the key attributes and attributes of the same diagram). The attribute key of the second entity, PEOPLE, is *people_ID*. This entity has an additional attribute, *people_name*. The third entity of the ERD is CAPTURE and it has *capture_ID* as the key attribute and *capture_symbol*, *capture_label* and *capture_path* as attributes.

Table 5-1: The entities of the first version of ERD of the system.

Entity	Description
MOMENT	The entity that holds information about the bookmarked moment.
PEOPLE	The entity that holds information about the people being tagged in the bookmarked moments.
CAPTURE	The entity that holds information about capturing the data by using one of the supported capturing ways of CatchIt, such as: note, audio, video and photo.

Relationships

MOMENT entity has two relationships with both entities PEOPLE and CAPTURE. The relationship between MOMENT and PEOPLE is *many-to-many*. One record of MOMENT entity has the ability to connect from zero to more than one record of PEOPLE entity. However, one record of PEOPLE entity must be connected to at least one record of MOMENT entity or more than one record (see Table 5-3 for further information about all relationships of the ERD). The other relationship of MOMENT entity is with CAPTURE entity. This relationship is *one-to-many*. Each record of MOMENT entity can connect to zero or more than one records of CAPTURE but each record of CAPTURE must be connected with exactly one record of MOMENT entity.

Table 5-2: The attributes of the entities of the first version of the ERD of the system.

Attribute name	Entity	Data Type	Primary Key	Required	Description
moment_ID	MOMENT	Integer	Yes	Yes	This unique number identifies the bookmarked moment.
moment_label	MOMENT	Varchar(25)	No	No	A label, which may be used by the user to identify the bookmarked moments
moment_time	MOMENT	Varchar(20)	No	Yes	A time of the bookmarked moment. It takes the format “dd-MM-yyyy hh:mm a” (“dd” = day of the month, “MM” = month, “yyyy” = year, hh = hours (1-12), mm = minute and “a” period (am or pm)).
time_label	MOMENT	Varchar(25)	No	No	A label of time for the bookmarked moment.
location_lat	MOMENT	Varchar(15)	No	Yes	The latitude of location of the bookmarked moment.

location_long	MOMENT	Varchar(15)	No	Yes	The longitude of location of the bookmarked moment.
location_label	MOMENT	Varchar(25)	No	No	A label of location of the bookmarked moment.
people_ID	PEOPLE	Integer	Yes	Yes	This unique number identifies the people taking a place of the bookmarked moment.
people_name	PEOPLE	Varchar(25)	No	Yes	Name of people taking a place of the bookmarked moment.
capture_ID	CAPTURE	Integer	Yes	Yes	This unique number identifies the captured data by using one of the supported capturing ways of CatchIt.
capture_symbol	CAPTURE	Varchar(1)	No	Yes	The symbol of the way that the bookmarked moment is captured by. The symbols are 'N', 'A', 'V' and 'P' and they refer to Note, Audio, Video and Photo respectively.
capture_label	CAPTURE	Varchar(25)	No	No	A label of the captured data.

capture_path	CAPTURE	Varchar(90)	No	Yes	File path of the captured data in the portable device.
--------------	---------	-------------	----	-----	--

Table 5-3: The relationships between entities of the first version of the ERD of the system.

Relationship name	Between entities	Type	Description
Has	MOMENT/ PEOPLE	M:N	A bookmarked moment may have no people, one person or more than one person, while each person must be in one or more than one bookmarked moment.
Is captured	MOMENT/ CAPTURE	1:N	Contents of the bookmarked moment may not be captured at all or may be captured by using one of the supported capturing ways of CatchIt. However, the captured data must belong to exactly one bookmarked moment.

5.3.1.2 The Second Version of ERD

In the first version of the ER diagram of the database of CatchIt, there was one *many-to-many* relationship between the MOMENT entity and PEOPLE entity. However, in the second version of that diagram, the physical ERD, the *many-to-many* relationship is broken up (see Figure 5.3). The result of that operation is one new entity, PEOPLE_IN_MOMENT, and two *one-to-many* relationships. Presenting the physical ERD helps us to describe the way of storing the data in the device and thus helps in the implementation of the database (Elmasri & Navathe, 2003).

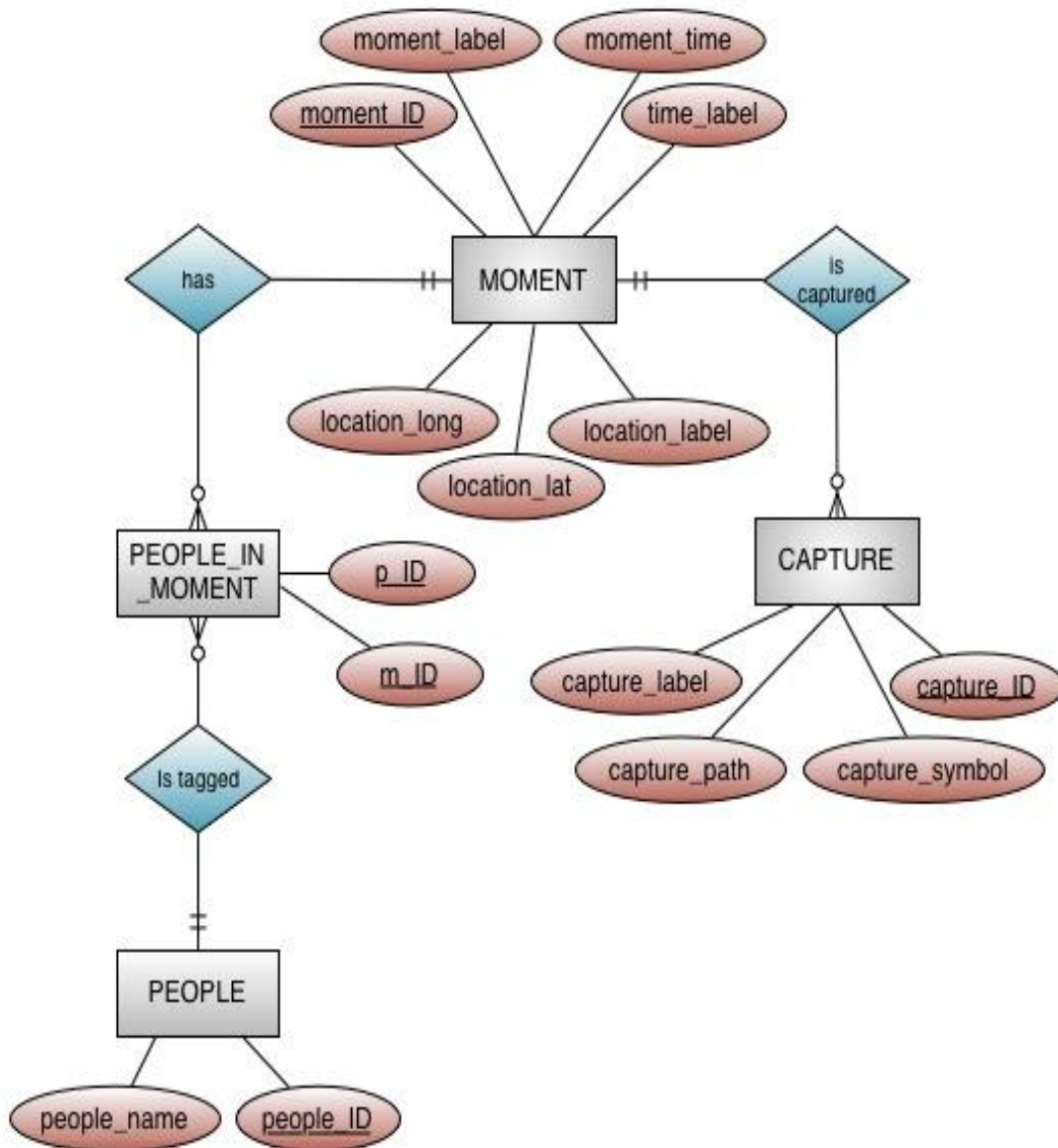


Figure 5.3: The second version of ERD. One more entity, PEOPLE_IN_MOMENT, with two key attributes, *p_ID* and *m_ID* and two *one-to-many* relationships, *has* and *is_tagged* are placed after breaking the *many-to-many* relationship between the entities MOMENT and PEOPLE of the first version of the ERD.

Entities and attributes

In addition to three entities of the first version of ERD (MOMENT, PEOPLE AND CAPTURE), there is a new entity in the second version, PEOPLE_IN_MOMENT. This entity holds information about all the people who joined the user in each bookmarked moment.

The PEOPLE_IN_MOMENT entity takes the key attributes of the MOMENT entity and the PEOPLE entity. Thus, the PEOPLE_IN_MOMENT has two key attributes: *m_ID*, which is known by *moment_ID* in the MOEMNT entity and *p_ID*, which is known by *people_ID* in PEOPLE entity (see Table 5.4 for further information on the attributes of PEOPLE_IN_MOMENT, which is produced in the second version of ERD).

Relationships

As mentioned at the beginning of this section (Section 5.3.1.2), the *many-to-many* relationship of the first version of ERD (between the entities MOMENT and PEOPLE) has produced two *one-to-many* relationships.

The first relationship is between the MOMENT entity and the PEOPLE_IN_MOMENT entity. This shows that a record of the MOMENT entity may relate from zero to more than one records of PEOPLE_IN_MOMENT, entity but one record of PEOPLE_IN_MOMENT entity must be related to exactly one record of MOMENT entity. Similarly, the second new relationship of the broken *many-to-many* relationship is between PEOPLE_IN_MOMENT and PEOPLE, which is a *one-to-many* relationship. In this way a record of PEOPLE may relate from zero or more than one records of PEOPLE_IN_MOMENT. In addition, each record of PEOPLE_IN_MOMENT entity must relate to exactly one record of PEOPLE entity (see Table 5-5 for further information on the relationships of the second version of ERD).

Table 5-4: The attributes of the new entity of the second version of ERD, PEOPLE_IN_MOMENT, of the system.

Attribute name	Entity	Data Type	Primary Key	Required	Description
m_ID	PEOPLE_IN_MOMENT	Integer	Yes	Yes	This unique number identifies the bookmarked moment. This attribute combines with another, <i>p_ID</i> , to creating composite primary keys for the PEOPLE_IN_MOMENT entity.
p_ID	PEOPLE_IN_MOMENT	Integer	Yes	Yes	This unique number identifies the people taking a place of the bookmarked moments. This attribute combines with another, <i>m_ID</i> , to creating composite primary keys for the PEOPLE_IN_MOMENT entity.

Table 5-5: All the relationships between the entities of the second version of the ERD of the system.

Relationship name	Between entities	Type	Description
Has	MOMENT/ PEOPLE_IN_MOMENT	1:N	These relationships appear when one or more than one people were tagged in any bookmarked moment where each record of the PEOPLE_IN_MOMENT entity has to be related to both one and only one record of the PEOPLE and the MOMENT entity.
Is tagged	PEOPLE_IN_MOMENT/ PEOPLE	1:N	
Is captured	MOMENT/ CAPTURE	1:N	Contents of the bookmarked moment may not be captured at all or may be captured by using one or more of the supported capturing ways of CatchIt. However, the captured data must belong to one and only one bookmarked moment.

5.3.2 Enhanced Entity Relationship Diagram (EERD)

In the enhanced entity relationship diagram the specializations of the entity relationship including both super-classes and sub-classes of the database of CatchIt will be addressed (see Figure 5.4). In addition, this diagram shows some constraints of the specializations.

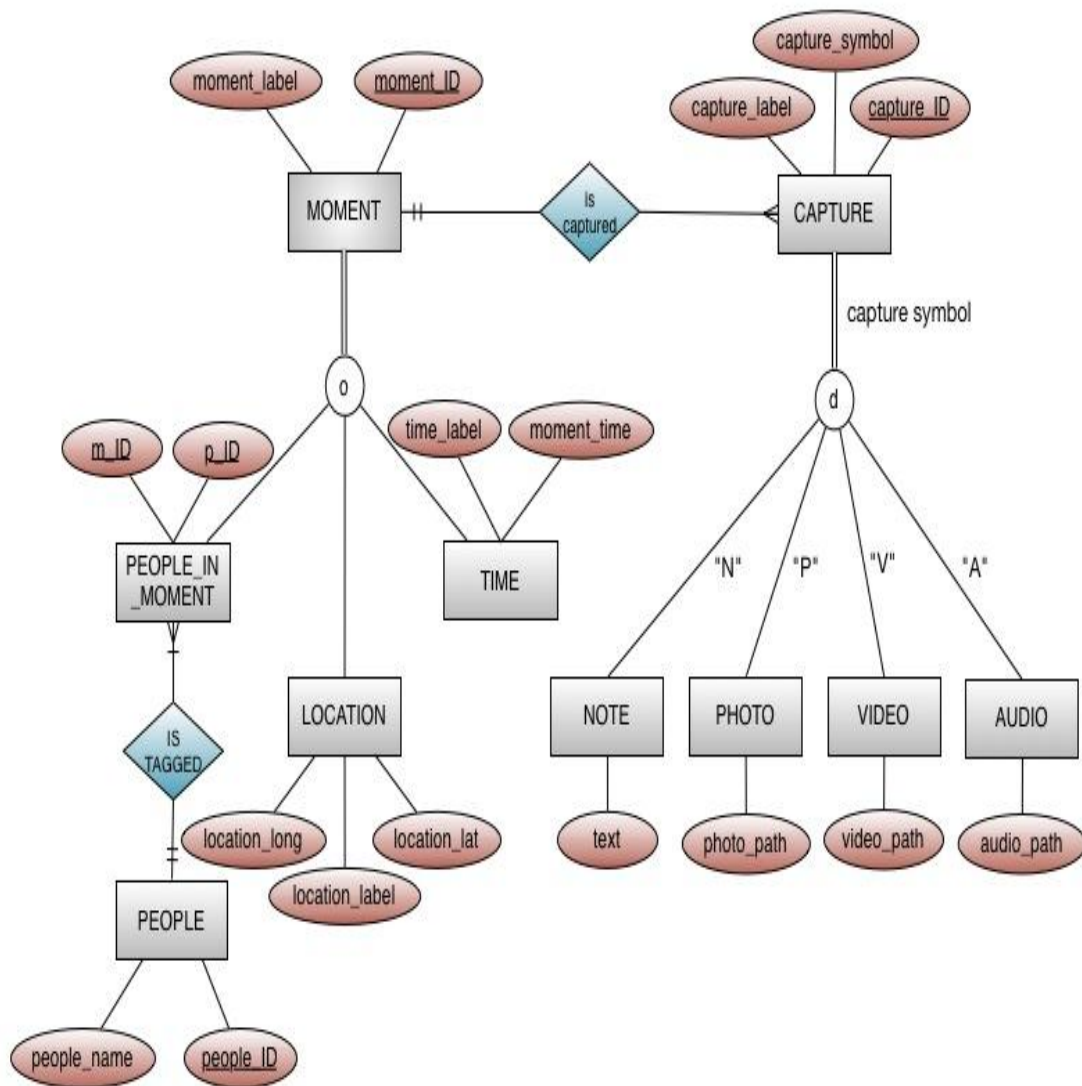


Figure 5.4: The enhanced entity relationship diagram of CatchIt. The set of sub-classes {PEOPLE_IN_MOMENT, LOCATION, TIME} is a specialization of the super-class MOMENT while the set of sub-classes {NOTE, PHOTO, VIDEO, AUDIO} is a specialization of the super-class CAPTURE. The relation between PEOPLE and PEOPLE entities is in a specific relationship type.

5.3.2.1 Specializations

There are two specializations in the EERD of CatchIt: (1) the specialization of the super-class MOMENT distinguishing among the moment entities based on the moment context and (2) the specialization of the super-class CAPTURE which distinguishes among the captured entities based on capture type.

Specialization of the Super-class MOMENT

The specialization of the super-class MOMENT has three sub-classes, PEOPLE_IN_MOMENT, LOCATION and TIME. The super-class MOMENT distinguishes among the bookmarked moment based on the context. The relationship between the MOMENT super-class and its sub-classes is known as *IS-A*. TIME, LOCATION and PEOPLE_IN_MOMENT are the context of the MOMENT entity.

The attributes of the specialization of the super-class MOMENT, *moment_ID* and *moment_label*, will be inherited to all sub-classes of this specialization {PEOPLE_IN_MOMENT, LOCATION and TIME}. The rest of the attributes of MOMENT super-class, which are discussed in second version of the ERD (see Figure 5.3), are distributed to the {LOCATION and TIME} sub-classes in EERD and these attributes become local attributes of the sub-classes. The LOCATION sub-class has local attributes *location_lat*, *location_long* and *location_label* while the TIME sub-class has the local attributes *moment_time* and *moment_label*.

The PEOPLE_IN_MOMENT sub-class is another sub-class of the MOMENT super-class. It has two local attributes, *m_ID* and *p_ID*, where both of them are key attributes. The PEOPLE_IN_MOMENT sub-class has a relationship with PEOPLE entity in the second version of the ERD. However, this relationship becomes a

specific relationship type in the EERD of CatchIt. In addition, the PEOPLE entity keeps its attributes as it was in the second version of the ERD, the key attribute is *people_ID* and the attribute *people_name*.

Specialization of the Super-class CAPTURE

The set of sub-classes, {NOTE, PHOTO, VIDEO, AUDIO} is a specialization of the super-class CAPTURE. The CAPTURE super-class distinguishes among the captured entities based on the capture symbol. Each symbol represents a particular supported capturing way that CatchIt has.

Similar to the specialization of the super-class MOMENT, the relationship between the CAPTURE super-class and its sub-classes is known as *IS-A*. NOTE is a supported capturing way and also for the rest of the sub-class PHOTO, VIDEO and AUDIO.

The CAPTURE entity in the second version of ERD has four attributes, *capture_ID*, *capture_label*, *capture_symbol* and *capture_path*. However, in the EERD, the super-class CAPTURE inherits the attributes *capture_ID*, *capture_label*, *capture_symbol* to the sub-classes of the specialization of CAPTURE. On the other hand, in the EERD the CAPTURE super-class dispenses with the *capture_path* attribute. *capture_path* becomes a local attribute for the sub-classes PHOTO, VIDEO and AUDIO after changing the name of the attribute appropriately. It is replaced to *photo_path*, *video_path*, and *audio_path* respectively. The local attribute of the NOTE sub-class is *text*, which does not need to be stored in a file path. This will be stored as a text in the database.

5.3.2.2 Constraints of Specializations

Different constraints of specialization of the EERD are applied on the specializations of the MOMENT and the CAPTURE super-class.

Constraints of the Specialization of the MOMENT Super-class

The two specialization constraints applied on the specialization of the MOMENT super-class are disjointness and completeness. The disjointness constraint is an overlap and the letter 'o' in the circle represents this constraint. This overlap is due to the MOMENT super-class may have more than one of its sub-classes {PEOPLE_IN_MOMENT, LOCATION and TIME}.

The second constraint of specialization of the MOMENT super-class is completeness constrain. This specialization is a total specialization and not partial because every record of MOMENT must be at least a member of {PEOPLE_IN_MOMENT, LOCATION or TIME} sub-classes. Hence, there is a double line in the connection between the super-class MOMENT and the circle of EER diagram in Figure 5.4.

Constraints of the Specialization of the CAPTURE Super-class

The specialization of the CAPTURE super-class is an attribute-defined specialization, because all sub-classes in the specialization of the CAPTURE super-class have their own membership condition *capture_symbol* on the same attribute of the super-class. So, the *capture_symbol* attribute is a defining attribute of the specialization. Based on the value of this attribute, the super-class will inherit its attributes to one of its sub-classes.

In the EERD of CatchIt the defining attribute's name of the CAPTURE specialization, *capture_symbol*, is placed next to the arc, between the circle and the CAPTURE sub-class. Next to each arc, which is between the circle and each sub-class, the particular value of the *defining attribute* is placed. The value 'N' makes the sub-class NOTE to take the place of the specialization, 'P' makes the sub-class PHOTO take the place of the specialization, 'V' makes the sub-class VIDEO take the place of the specialization and 'A' makes the sub-class AUDIO take the place of the specialization.

Two other constraints apply to the specialization of the CAPTURE class: disjointness and completeness constraints. The disjointness constraint in this specialization is a disjoint and this constraint is represented in the EERD by the letter 'j' in the circle of this specialization. It is a disjoint because the CAPTURE super-class must have at least one of the sub-classes of specialization NOTE, PHOTO, VIDEO or AUDIO.

On the other hand, similar to the specialization of the MOMENT super-class, the specialization of the CAPTURE super-class is a total specialization because every entity of the CAPTURE super-class must be at least a member of NOTE, PHOTO, VIDEO or AUDIO sub-classes. Hence, there is a double line in the connection between the super-class CAPTURE and the circle of this specialization.

5.3.3 Relational Database Schema

All entities of the EERD are located in the relational database schema. Each entity, in addition to its own attributes, includes the foreign attribute, which it refers to the key attribute of the related entity (see Figure 5.5). The foreign key in the CAPTURE entity, *m_ID* points at the key attribute of the MOMENT entity, (*moment_ID*). *p_ID* is one of the composite keys of the PEOPLE_IN_MOMENT entity as well as a foreign key which refers to the key attribute of the PEOPLE entity, *people_ID*.

Furthermore, the sub-classes in the specialization deal with the key attribute of the super-class of that specialization as their own key attribute. The key attributes of the sub-classes in the relational databases schema relate to their super-classes particularly to the super-classes' key attributes. The sub-classes PEOPLE_IN_MOMENT, LOCATION and TIME have the key attribute *m_ID* and this attribute refers to the key attribute, *moment_ID*, of the MOMENT super-class. On the other hand, the sub-classes NOTE, PHOTO, VIDEO and AUDIO have the key attribute *c_ID*, which points at the key attribute *capture_ID* of the CAPTURE entity.

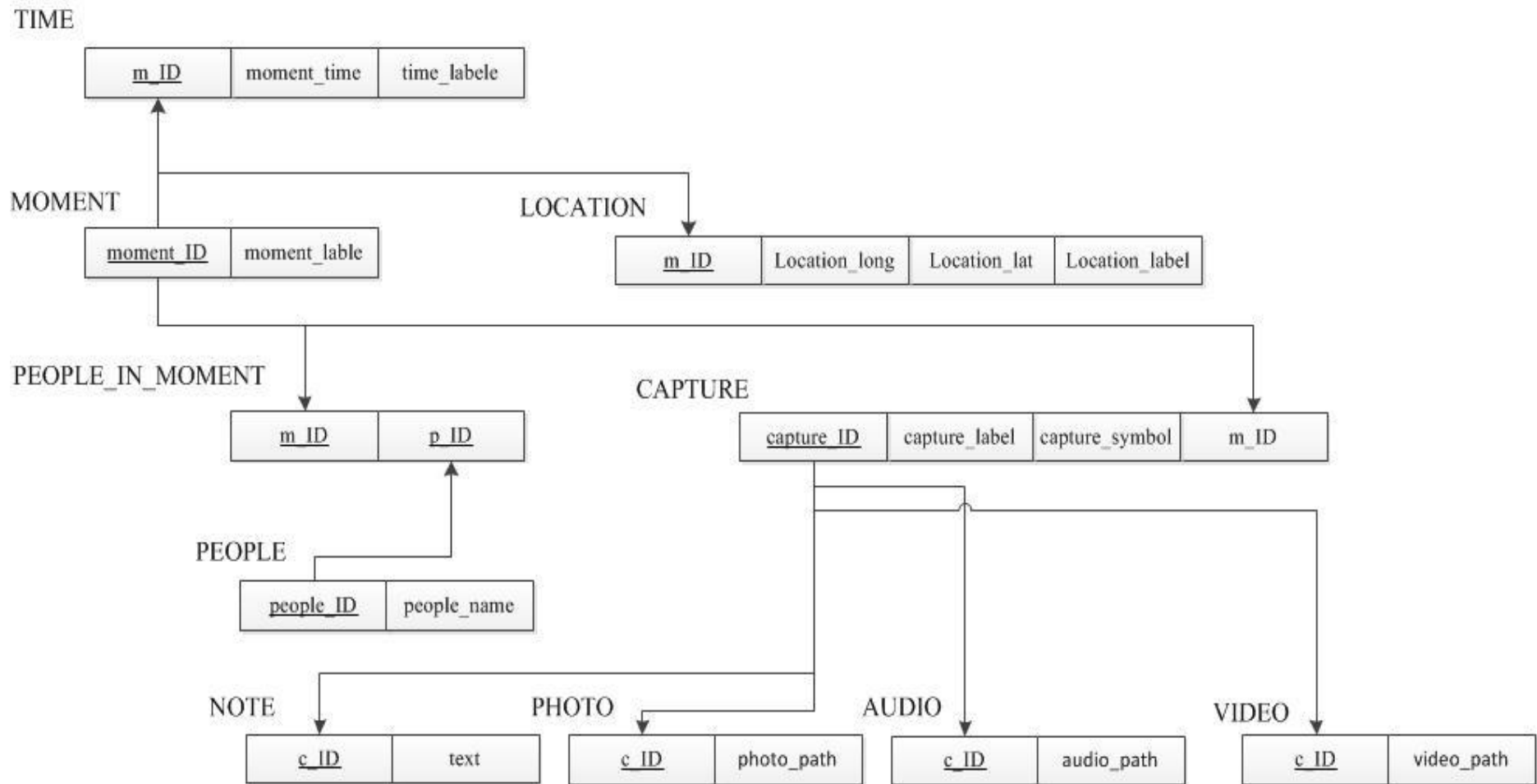


Figure 5.5: The relational database schema.

5.4 The User Interface

In this section the scenario of our persona Baraa (introduced in Chapter 2) will be reviewed following the way of the interaction between Baraa and CatchIt user interface.

5.4.1 Review of the Scenario

Baraa is a typical user of CatchIt. His life is busy and he is interested in telling people about his interesting moments. Two of the moments that he wants to keep are the moment when he and his friends gathered at the university before going to watch a rugby match in the stadium, and the current moment, the moment of watching the rugby match.

On Saturday 30th June 2011 at 04:30 PM, Baraa is waiting for his friends Mu and Simon at the university in order to go together to Westpac Stadium to watch the rugby match between the All Black and South Africa. Baraa's friends arrive at the university; together they go to the stadium. However, even though Baraa has wished to bookmark this moment by using CatchIt, he forgets to do that.

On Saturday 30th June 2011 at 06:00 PM, Baraa is watching the rugby match with his friends. This experience is the first for Baraa - watching a rugby match and the All Blacks haka. The haka reminds him of a traditional dance of his home country.

Meanwhile, Baraa has a new friendship. It is with Judi who at first thinks Baraa is Brazilian which makes him laughs. While Baraa and Judi are talking Baraa's friend,

Mu, offers hotdogs to Baraa and Judi. In addition, Simon breaks his glasses and Baraa's mother phones Baraa but he prefers to call her back when he returns to his apartment.

5.4.2 CatchIt User Interface

We will explain the user interface of CatchIt using, again, the scenario. The persona of the scenario, Baraa, wishes to capture two moments while he is in the stadium. The first one is the moment of gathering with his friends and the second one is the moment of watching the rugby match. After a period of time, Bara wishes to revise the bookmarked moments.

The Moment of Gathering

In the stadium Baraa remembers the moment when he and his friends had gathered at the university. He wants to bookmark that moment. Because CatchIt allows Baraa to bookmark the previous moments; therefore, Baraa runs the application on his iPhone by touching the icon of CatchIt (see Figure 5.6 (a)). CatchIt starts working while its main window appears. The window has three buttons *Bookmark a moment*, *Revise and earlier* and *Synchronize* (see Figure 5.6 (b)). Each button has one main function of CatchIt.



Figure 5.6: (a) is the icon of CatchIt while (b) is the main window of CatchIt, which has the main functions of the applicaiton.

When Baraa's wishes to bookmark the moment of gathering with his friends, he presses the *Bookmark a moment* button. CatchIt captures the coordinates of the current location of Baraa, the stadium and the current time (30.06.2011 05:45 PM) and then moves Baraa to the *Moment Details* window. The message box pops up and tells Baraa that the target moment is bookmarked (see Figure 5.7 (a)). Once he pressed *OK* button in the message box, he could use the functions of the *Moment Details* window.

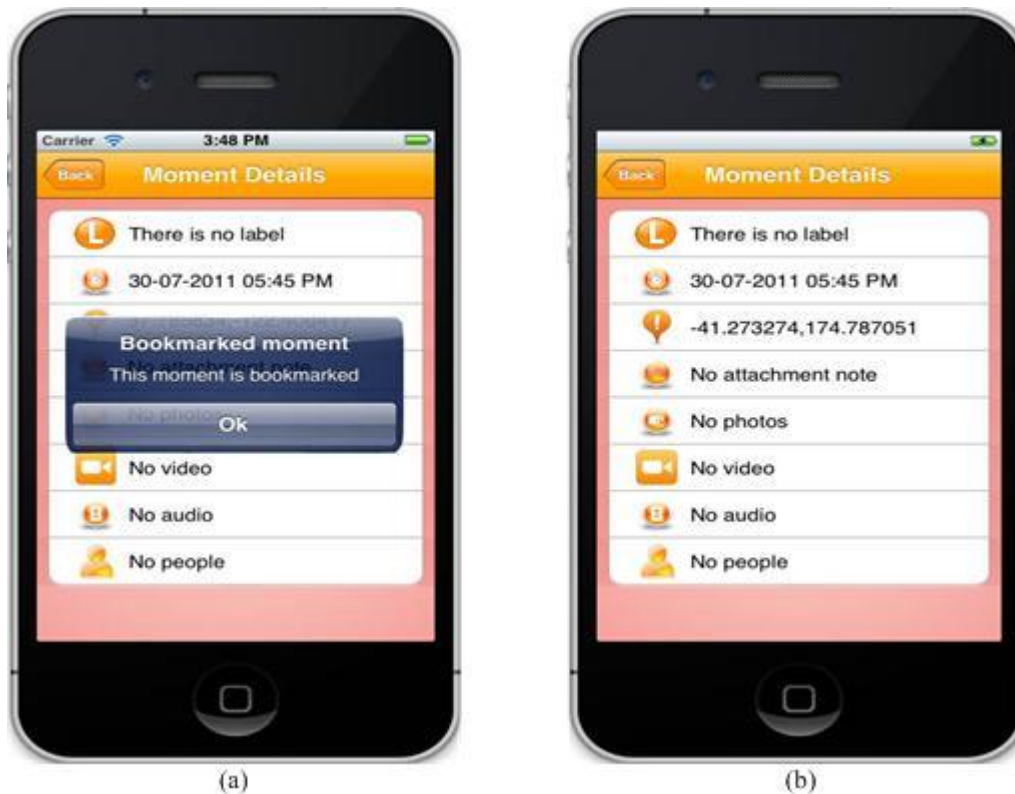


Figure 5.7: (a) Message box pops up when the target moment is bookmarked. (b) The *Moment Details* window has a table of functions of the bookmarked moment.

The *Moment Details* window has a table with eight rows. Each row has an image that describes the content and the label of that row. The eight rows are: the label of the bookmarked moment, the time and the location of that moment, note, photos, videos and audios of that moment. The last row is for tagging people who are with the user (Baraa at that moment). All these rows except the rows of time and location show that there are no data because CatchIt does only capture the time and location automatically (see Figure 5.7 (b)).

Baraa right now wants to label the bookmarked moment. Also, he wants to change the time and the location of the bookmarked moment to the time of gathering with his friends at the university. To do so, firstly, Baraa touches the first cell of *Moment Details* window which is for the label of the bookmarked moment. Then he moves to

Moment Label. In this window Baraa types *Gathering with friends to go to watch a rugby match* as a label of the bookmarked moment (see Figure 5.8). After that he saves the changes by pressing the save button; CatchIt moves Baraa to the *Moment Details* window.



Figure 5.8: The Moment Label window holds the label of the bookmarked moment.

To change the time of the bookmarked moment, Baraa touches the second cell of the table in that window and he moves to *Moment Time* window. He picks the required data and time (30.07.2011 04:30 PM) from the time picker and he sees the picked date and time at the bottom of the window. Also, Baraa inserts *Before going to the stadium*, a label of time of the bookmarked at the top of the same window (see Figure 5.9). By pressing the *Save* button Baraa moves back to the *Moment Details* window.



Figure 5.9: In the *Moment Time* window the user can reset the time of the bookmarked moment and insert a description of that time.

From the *Moment Details* window Baraa touches on the third cell of the table which has the geometric data of the captured location to show that location on a map in the *Location* window (see the location of the bookmarked moment on the map in Figure 5.10 (a)). Then he presses the *Edit* button of that window to edit the location to the required target location. Once he presses that button he moves to the *Moment Location* window. He changes the captured address to the target ones and then he inserts a description of the target location, *University*, and saves the changes by pressing the *Save* button (see Figure 5.10 (b)).



Figure 5.10: (a) The *Location* window shows the target location on the map while (b) shows the *Moment Location* which allows the user to change the location of the bookmarked moment by inserting the address of the target location and also this window also allows the use to insert a description of the target location.¹

After that CatchIt moves Baraa automatically to the previous window (*Location*) and from there Baraa goes back to the *Moment Details* and then to the main window of CatchIt.

The Moment of Watching the Rugby Match

After fifteen minutes of bookmarking the first moment (at 06:00 PM) Baraa wishes to bookmark the current moment while he is watching the rugby match. Therefore, he runs CatchIt again and bookmarks the moment. This time he wants to attach a note, photos, a video record, and an audio record.

¹ We are using Google map in CatchIt but for illustration purposes we will use our mockup map.

In order to add notes Baraa goes to the *Note* window via the *Moment Details* window by touching the fourth row of the table. In the *Note* window he types his note, *Mu gave me hotdogs* and *Simon broke his glasses* as Figure 5.11 shows. There are no more notes Baraa wants to take, so he presses the *Done* button which is located at the top of the *Note* window, to save his notes without adding a label for that note. Then he moves back to the *Moment Details* window.

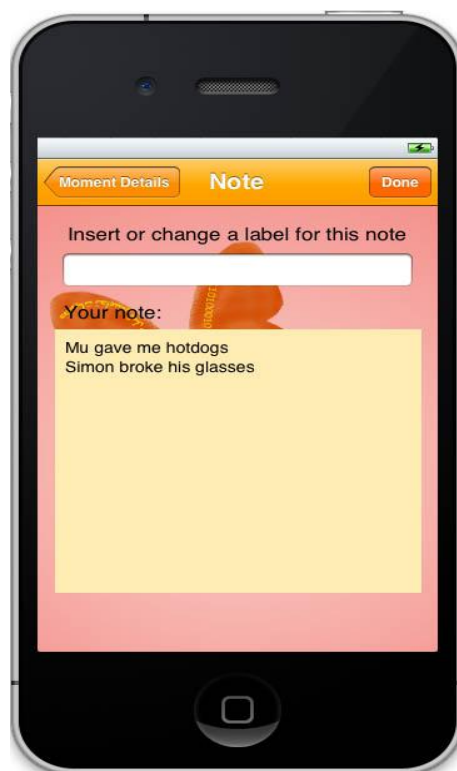


Figure 5.11: Taking notes of the bookmarked moment and inserting a label for the note are available in the *Note* window.

After taking notes Baraa starts taking photos. He touches the photo's cell of the table in the *Moment Details* window to move to the *Photos* window. In the *Photo* window there is an empty list of photo because Baraa has not taken any photo of the bookmarked moment (see Figure 5.12).



Figure 5.12: There is no photo in the *Photos* window as long as the user has not taken a single photo of the bookmarked moment. To add photo the user should press on the camera button at the bottom left corner of the window.

To add a photo, Baraa presses the camera symbol at the bottom left corner of the window and the camera starts working. He takes the first photo, uses it by pressing the *Use* button (see Figure 5.13 (a)) but he does not insert a caption for it in the *Photo* window. He saves the first photo by pressing the *Save* button, returns to the *Photos* window and takes another photo by the same procedure he used the first time. This time he inserts a caption for this photo (*Judi*) and saves it (see Figure 5.13 (b)).



Figure 5.13: (a) The user takes a photo and then he can use it by pressing the *Use* button. (b) After using a photo, the user moves to the *Photo* window where the user can insert a caption of the photo.

When Baraa comes back to the *Photos* window, the two photos are listed in small size with their associated captions. Because the first photo does not have a caption, the first photo is listed with *No label* as a caption of it (see Figure 5.14).



Figure 5.14: *Photos* window after taking photos of the bookmarked moment. Each photo is shown in a small size with its caption. The caption *No label* means there is no associated caption for the photo.

Baraa changes his mind about saving the first photo. He decides to delete it from the list of photos. From the *Photos* window he presses the editing button, *Edit*. The *Edit* button is changed to *Done* and the deletion symbol is placed at the left side of the photo (see Figure 5.15 (a)). Baraa chooses the first photo which visualizes the *Delete* button next to it (see Figure 5.15 (b)). Baraa presses the *Delete* button to delete the photo. When Baraa captures from the deletion, he presses the *Done* button and the “Photo” window returns to the regular view (same as Figure 5.14). Then he goes back to the *Moment Details* window.



Figure 5.15: (a) The first step of deleting a photo from The *Photo* window is to choose the target photo. (b) The second step is choosing a particular photo. In this step the deletion symbol rotates and the *Delete* button is made visible.

Similarly, Baraa adds one video record. He chooses the row of recording video of the *Moment Details* window and then he moves to the *Videos* window, which has an empty list of video as there is no captured video (see Figure 5.16).

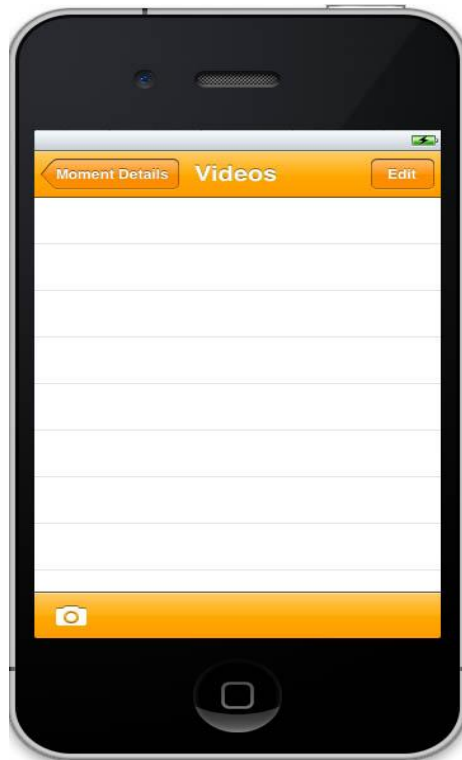


Figure 5.16: An empty video list in the *Videos* window because of there is no recording video of the bookmarked moment.

Baraa presses the camera symbol to make the built-in camera work; then he starts capturing the All Blacks haka (see Figure 5.17 (a)). Since CatchIt does not accept recording video longer than one minute, therefore after one minute of recording, a message box pops up. This informs Baraa that the video recording has reached the maximum length (see Figure 5.17 (b)).

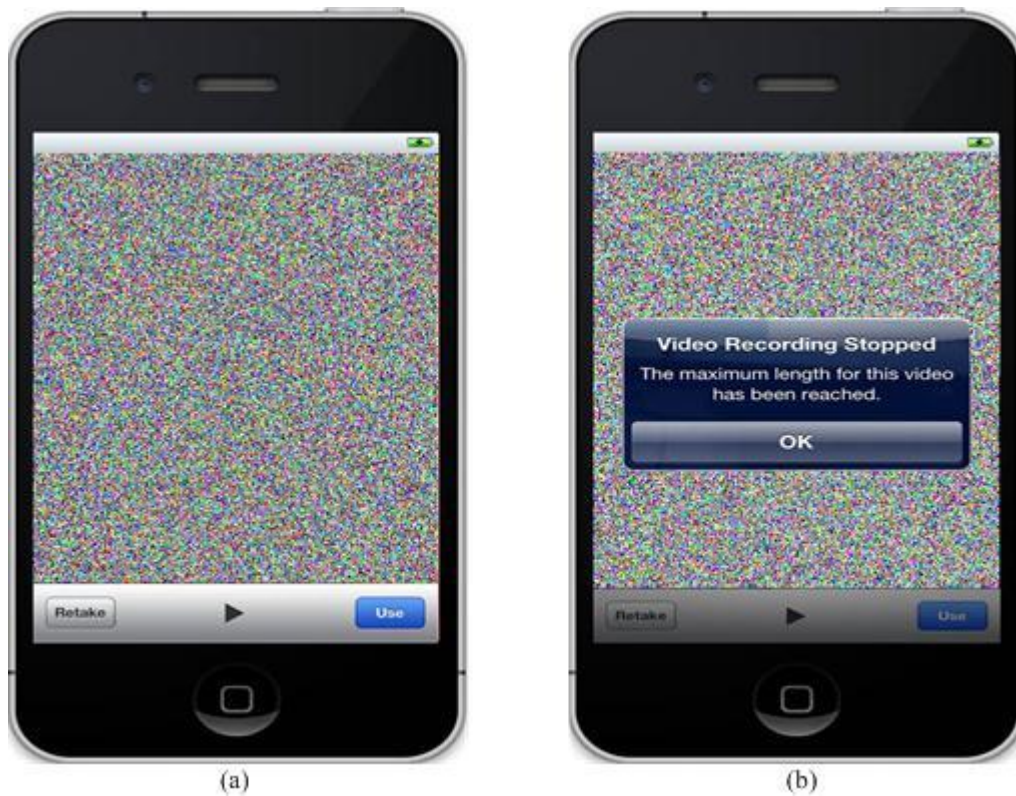


Figure 5.17: (a) The built-in camera works in a video mode. (b) The video recording stops if the recording time reaches one minute and a message box pops up to inform the user.

Once Baraa decides to use the captured video, he moves to the *Video* window to insert a caption for that video (All Blacks haka) and to save it (see Figure 5.18 (a)). After saving the captured video, CatchIt moves Baraa to the *Videos* window where the caption of the new captured video is listed next to the serial number (see Figure 5.18 (b)).



Figure 5.18: (a) The *Video* window when the user adds a caption to the recorded video. (b) The *Videos* window shows the list of captured video.

Baraa can delete a recorded video in the same way of deleting a photo. Firstly, Baraa should press the *Edit* button. Then he can select the video that he wishes to delete; he then presses the *Delete* button and finally presses *Done* button which ensures that Baraa has finished deleting the target video. If, however, Baraa did not wish to delete a recorded video, and thus, he would press the *Done* button without choosing any of recorded to be deleted.

Baraa wishes to record part of a conversation with Judi. To do so Baraa chooses the seventh cell of the table in the *Moment Details* window. The empty list of recording audio then appears in the *Audios* window (see Figure 5.19).



Figure 5.19: The audio list in the *Audios* window is empty because of there is no recorded audio of the bookmarked moment.

Baraa presses the '+' sign at the bottom left corner of the window. He moves to the *Audio* window to start recording the conversation (see Figure 5.20 (a)). He knows that the maximum time for each recorded audio is one minute, because there is instruction on the recording limit at the top of the window. Baraa presses the recorder button and the timer runs to inform him on the duration of the recording.

Since Baraa wants to record twenty minutes of the conversation he presses the stop button. Now he allows associating a caption to the recorded audio. He inserts "A conversation with Judi". He saves the recorded audio and then he moves back to the *Audios* window where the new recording is listed (see Figure 5.20 (b)).



Figure 5.20: (a) The *Audio* window allows the user to record the target moment and it includes an instruction of the maximum length of the recorded audio. (b) When the user stops recording, the user is allowed inserting a caption of the recorded audio.

Baraa tries to record another part of the conversation with Judi. He starts recording and when the recording time reaches the maximum available time, the recording stops automatically and he is allowed to insert a caption to that audio. However, Baraa does not want to save the last recorded audio. Thus, he skips the saving operation and goes back to the audio list. If Baraa wishes to delete any recorded audio from the list, he will do the same procedure as for deleting video or photo from CatchIt.

Baraa goes back to the *Moment Details* window. This time all the contents of the rows (notes, photo, video and audio) have changed as a result of attaching data. The note row contains the beginning of the notes. The photo, video and audio rows have

changed to *Photo*, *Video* and *Audio*. Therefore, later on, Baraa is able to recognize which data he inserted in the bookmarked moment (see Figure 5.21).



Figure 5.21: The *Moment Details* window after attaching notes, photos, videos and audios to the bookmarked moment. The values of the fourth, fifth, sixth and seventh cells in the table have changed.

Revising an Earlier Moment

At home, when Baraa returns from the stadium, he wishes to watch the recorded video of the second bookmarked moment (All Blacks haka) and tag his friends in both bookmarked moments. Therefore Baraa presses the *Revise an earlier* from the main window *CatchIt*. He moves to the *Moments* window where all the bookmarked moments are listed. Each bookmarked moment is represented by its time as a constant content of the each cell in the table, in addition to the label of that moment, if it is associated (see Figure 5.22).



Figure 5.22: All bookmarked moments are listed in the *Moments* window including the data and time of the bookmarked moment as well as the label of the bookmarked moment in the case of adding a label to that moment.

Baraa selects the second bookmarked moment and then chooses video from the table of the *Moment Details* window. Then he chooses the target video (All Blacks haka). In order to play the movie Baraa presses the play button of the *Video* window as shown in Figure 5.18 (a); he is now watching the video (see Figure 5.23).

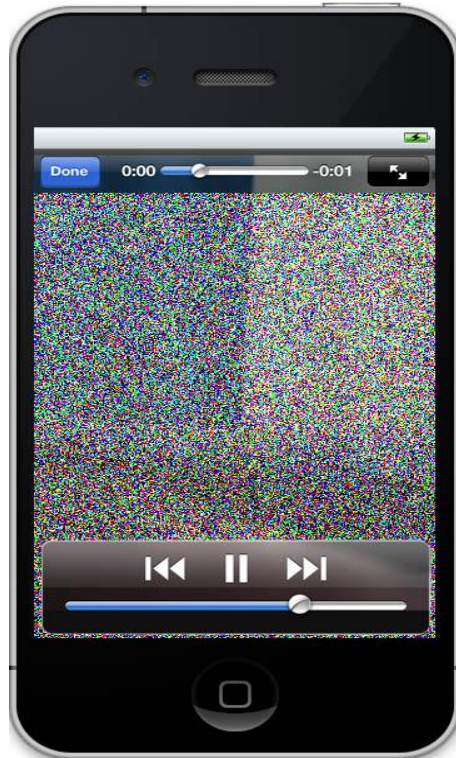


Figure 5.23: Playing the recorded video in CatchIt.

Baraa starts tagging people in the first bookmarked moment, the moment of gathering, by pressing that bookmarked moment. He moves to the *Moment Details* window of the target moment. He presses the last cell of the table; then he moves to the *Tag People* window, which has no people's name in its list as no one has been tagged in this moment (see Figure 5.24 (a)). However, in this window Baraa can tag people to the bookmarked moment. Therefore, he presses the '+' sign in the bottom left corner of the window and then he moves to the *Select People* window. As long as Baraa has not tagged people ever in any of his bookmarked moment, there is no name on the list of peoples' names (see Figure 5.24 (b)). So, Baraa should firstly feed this list by names of his friends who were with him in the first bookmarked moment and secondly he could select people from this list and tag them in the target moment.



Figure 5.24: Both lists in (a) the *Tag People* window and (b) the *Select People* window are empty because there are no people tagged in the current moment, and there are no people tagged in any previous moments.

Baraa presses the '+' on the *Select People* window to go to the *Person* window (see Figure 5.25). From that window Baraa is allowed to insert the name of his friends who shared with him the target bookmarked moment.

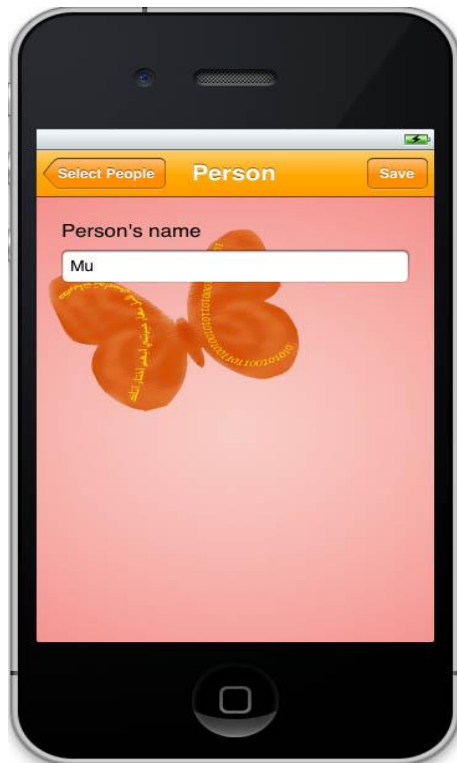


Figure 5.25: The *Person* window allows the user to insert a new person's name as a first step to tag this person in the bookmarked moment.

Baraa inserts his friends Mu and then inserts Simon and saves their names. As a result he sees both names in the list of the *Select People* window. He selects both names and presses the *Select* button (see Figure 5.26 (a)). The selected names are saved and then added immediately in the list of names of the *Tag People* window (see Figure 5.26 (b)). Then Baraa goes back to the main window via the *Moment Details* window.



Figure 5.26: (a) Selected names in the *Select People* window will go to (b) the *Tag People* window.

After that Baraa wants to tag people in the second bookmarked moment, the moment of watching the rugby match. He follows the procedure as in the previous bookmarked moment, when he tagged his friends. However, this time when Baraa reaches the *Select People* window, he finds the names of his friends who shared with him the previous moments. Nevertheless, he wishes to tag his new friend Judi in that moment as well as his friends Mu and Simon, who shared with Baraa the moment of gathering. So, Baraa adds the name of his new friend firstly in the *Select People* window; he then selects all the names of that window to be tagged in the second bookmarked moment. Both lists of peoples' names are listed alphabetically so that Baraa can find and select the target name easily from the *Select People* window. Additionally, he can recognize the tagged people easily in the *Tag People* window.

When Baraa revises the bookmarked moments, he wishes to transfer them to his desktop, particularly to Digital Parrot. Further information about transferring the data of the bookmarked moments from the iPhone to Digital Parrot will be specified in the next section (see Section 5.5).

5.5 Transferring the Data of the Bookmarked Moments

Digital Parrot is a desktop application. It combines the context-awareness with semantic information in order to augment the human memory (Schweer & Hinze, 2008). However, Digital Parrot deals with the textual information in order to associate the context with the content of the user's experiences, semantically, and also in order to retrieve the required information.

As a consequence, CatchIt will feed Digital Parrot with the contexts and the textual contents of the bookmarked moment. A concern in this research thesis, in terms of feeding Digital Parrot, is to find a way to do that. However, at this stage of implementing CatchIt, we will concentrate on structuring the contexts (time, location and people) and the some of the textual contents (labels of bookmarked moment, time, location, notes, photos, videos and audios) in an XML format (plist file). After that, the XML file will be sent to the e-mail address that the user wishes to send to. The XML file is shown in Figure 5.27.

```
<?xml version = "1.0"?>
<moments>
<momentID>1</momentID>
<momentLabel>Gathering with friends to go to watch a rugby match</momentLabel>
<momentTime>30-07-2011 04:30 PM</momentTime>
<momentTimeDescription>before going to the stadium</momentTimeDescription>
<momentLocationLat>-41.289328</momentLocationLat>
<momentLocationLong>174.767547</momentLocationLong>
<momentLocationDescription>University</momentLocationDescription>
<people>
<person>Mu</person>
<person>Simon</person>
</people>
</moments>
```

Figure 5.27: The XML file has the required data of the bookmarked moments to be sent to Digital Parrot.

The same file might be used in Digital Parrot after restructuring the file as Digital Parrot requires. The user interface of Digital Parrot might need to be changed to receive the XML file from CatchIt.

In the user interface of CatchIt the persona of the scenario, Baraa, can send an e-mail to himself or to any e-mail address he wishes by pressing the *Synchronize* button (see Figure 5.28). The XML file (plist file) will be attached to the e-mail automatically on behalf of Baraa.



Figure 5.28: XML file (plist file) which includes the data of the bookmarked moments is attached with an email to be sent.

5.6 Summary

The focus of this chapter is on implementing our application, CatchIt, based on the selected suitable portable device, iPhone, and the target users of the application. The local storage was preferred for storing the data of the bookmarked moments that the user bookmarked through CatchIt.

Consequently, a light database was designed and established in order to install the database in the portable device. To build a good database, the entity relationship diagram (ER) was drawn in two versions (logical and physical). Then the enhanced relationship diagram (EER) was drawn after grouping the entities of the second version of the (ER) to specializations where the super-classes and sub-classes were

determined. Afterwards, the relational database schema was drawn to define how the entities are related to each other.

Following on from establishing the database of CatchIt, the suggested user interface was implemented in the portable device using iOS 5. The scenario was used to clearly explain the interaction of the user with the user interface of CatchIt.

Finally, the proposed idea of synchronizing CatchIt with Digital Parrot (explained in the design chapter, Chapter4) was implemented. The XML file that has the textual information of the bookmarked moment is attached in an e-mail to be sent to Digital Parrot. Synchronizing CatchIt with Digital Parrot however is not the subject of research of this thesis.

The next chapter will evaluate the usability of CatchIt with the aim of making it easier to use in order to feed Digital Parrot appropriately.

6. Evaluating the Usability of *CatchIt*

This chapter describes the end-user study. This part of the research aims to evaluate the CatchIt system's usability. The study included comparisons between:

- (1) CatchIt and Hansel where Hansel is a simple mobile application which tags selective places to revisit and.
- (2) CatchIt and Contacts where Contacts is a built-in iPhone application that allows the user to add contacts details of people.

Several goals were formulated at the beginning of the study. After defining these goals of the study, the design of the study is described in accordance with those goals. Finally, the results of the study are addressed and then discussed.

6.1 Goals of the Study

There are three central goals of the study. The first goal is to investigate the user needs of the ease of the user interface of the mobile application and of the effective cues to recall the stored information. Accordingly, the second goal is to improve the usability of the user interface of CatchIt to feed the desktop application, Digital Parrot in the best way, which retrieves the stored personal information in a powerful way.

The user study was designed according to the goals defined above. A total of 12 participants were involved in the study. The participants were asked to walk through several steps in order to achieve the goals of the study.

6.1.1 Method

The study is based on using two mobile applications, CatchIt and Hansel. Additionally, a regular interaction application of the mobile phone was involved in the study (adding a new contact details to the mobile phone's contacts list by using a built-in iPhone application, Contacts). The participants used both applications, CatchIt and Hansel for the same length of time (one week). Subsequently, the participants were interviewed individually.

6.1.2 The Applications of the Study

The study involved three applications CatchIt (the application that we developed), Hansel and Contact. For this study, participants tested only the first two applications. The third application, Contacts, was not used for this study because the participants were already familiar with it. Contacts only used to compare between its user interface and the user interface of CatchIt. However, the applications Hansel and Contacts were involved in the study these applications share some aspects, which are similar to aspects of CatchIt (see Table 6-1).

Hansel uses a combination of time and location to present the required data of each tagged place in the same way what CatchIt does. In addition, Hansel and CatchIt allow the user to insert a label of the target place as well as for the moment. On the other hand, Contacts has different types of the required data. It uses semi-required data where one of those data is enough to save the target contact. Nonetheless, Contacts selects one of the semi-required data to label the contact on behalf of the user. Similar to CatchIt, the user of Contacts is able to add relatives' names to the target contact. In addition, Contacts allows the user to insert textual information such

as e-mail and company name of the target contact, as well as photo of the target contact.

Table 6-1: Common aspects between CatchIt and Hansel and between CatchIt and Contacts. The common aspects with CatchIt are symbolized by ✓ whereas aspects that are not common are symbolized by ×.

Aspects of CatchIt		Applications	
Category	Aspect	Hansel	Contacts
Required data	Time	✓	×
	Location	✓	×
Optional data	Label	✓	✓
	People	×	✓
	Notes (textual data)	×	✓
	Photo	×	✓

Moreover, these two applications were involved in the study due to the differences of the interacting ways between the user and the user interface of these applications. Those differences provided us with further knowledge regarding the preferred aspects of the user interfaces of the mobile applications.

6.1.3 Participants

Total number of participant was 12 (6 females, 6 males). Participants were recruited via personal contacts. Because CatchIt is developed for iPhone in addition to the ability of installing the *Hansel* application on iPhone, all participants had to have iPhone to participate in the study. (The application Contacts is already on iPhone, by default).

6.1.4 Procedure

The study was conducted over three weeks. The first two weeks were for using CatchIt and Hansel when there was no concentration on adding new contact details during that time because of regularity of doing that. The last week of the study was for interviewing the participants.

At the beginning of the study, participants were given brief information about our system and about the study. These pieces of information were addressed in the participant information sheet. In addition, the research consent form was handed in to the participants which has informed them of their right in the study. The participant information sheet and the research consent form are shown in Appendix A.

In general, the study has three significant steps: (1) initial questionnaire, (2) using the applications CatchIt and Hansel, and (3) guided interview.

Initial Questionnaire

The first step of the study was the initial questionnaire. The questionnaire was used to learn more about the preferred ways to capture information in different situations in

order to recall it. For this purpose, different scenarios were introduced in this questionnaire. Table 6-2 lists the questions of the initial questionnaire (see Initial Questionnaire in Appendix A).

Table 6-2: Questions of the initial questionnaire.

# Q	Question
1	Age group?
2	Gender?
3	Which way would you prefer to capture the events in your friends' birthday party?
4	Which way would you prefer to capture the important notes in a lecture?
5	Which way would you prefer to capture what you should buy from the supermarket?
6	Which way would you prefer to capture the details of the meeting with your supervisor/ advisor?
7	Which way would you prefer to capture the interesting information about a product?
8	Tick the best two ways that you use to record moments in your life?

Using CatchIt and Hansel

The second step of the study consisted in using both applications, CatchIt and Hansel. In this step, participants were divided to two groups, Group A and Group B. Each group had six participants (3 females, 3 males). Each participant used the applications individually. Participants of Group A started using CatchIt while participants of Group B started using Hansel. After one week, participants of Group A moved to use Hansel while Group B moved to use CatchIt (see Table 6-3).

Table 6-3: Two groups of participants used each application, CatchIt and Hansel, for a week. Group A started using CatchIt then used Hansel whereas Group B started using Hansel then used CatchIt.

	Group A		Group B	
	3 females	3 males	3 females	3 males
First week	CatchIt		Hansel	
Second week	Hansel		CatchIt	

The aim of testing the application Hansel was to make the participants different user interface of CatchIt. The user interface of Hansel is very easy to use because of using only one button to tag a particular place. On the other hand, the reason behind testing CatchIt and Hansel for the same length of time (one week) was to help the participants to become familiar with using these applications and thus to make a more informed judgment concerning the ease of using CatchIt as well as the usability problems of CatchIt.

Additionally, using both applications for the same period of time helped the participants to be familiar with using those applications to make a fair judgment about the ease of use CatchIt as well as to find the usability problems of CatchIt.

Guided Interview

A guided interview was the third step of the study. It was carried out when the participants completed using both applications, CatchIt and Hansel. Each participant was interviewed individually. In the interview, participants were asked five categories of questions.

The first category of questions concerned the participants' experiences of using our system, CatchIt. The second category of questions concerned the participants' experience of using the system Hansel. The third category of questions aimed to compare the two applications, CatchIt and Hansel, according to the participants' experience in using them. The fourth category of questions designed to compare between CatchIt and Contacts. The fifth category of questions aimed to extract the participants' feelings and interests using in applications. All questions of the five categories are listed in Table 6-4.

Table 6-4: Five categories of questions in the guided interview.

Category	#Q	Question
The Ease of Using CatchIt	1	How easy is it to bookmark a moment?
	2(a)	Have you added a label to any of your bookmarked moments?
	2(b)	How easy is it to add a label to the bookmarked moment?
	3(a)	Have you added notes to any of your bookmarked moments?

	3(b)	How easy is it to add notes to the bookmarked moment?
	4(a)	Have you added photo/s to any of your bookmarked moments?
	4(b)	How easy is it to add photo/s to the bookmarked moment?
	5(a)	Have you added video/s to any of your bookmarked moments?
	5(b)	How easy is it to add video/s to the bookmarked moment?
	6(a)	Have you added audio/s to any of your bookmarked moments?
	6(b)	How easy is it to add audio/s to the bookmarked moment?
	7(a)	Have you tagged people in any of your bookmarked moments?
	7(b)	How easy is it to add people to the bookmarked moment?
	8(a)	Have you changed the time of any bookmarked moment?
	8(b)	How easy is it to change the time of that bookmarked moment?
	9(a)	Have you changed the location of any bookmarked moment?
	9(b)	How easy is it to change the location of that bookmarked moment?
	10	How easy is it to go back to the previous window?
The Ease of Using Hansel	1	How easy is it to tag a place?
	2(a)	Have you added a label to any of your tagged place?
	2(b)	How easy is it to add a label to the tagged place?
	3	How easy is it to go back to the previous window?
CatchIt Vs Hansel	1	Which application is easier to deal with regarding bookmarking a moment in CatchIt, or tagging a place in Hansel?
	2	Which application is easier to distinguish the target: the bookmarked moment in CatchIt or the tagged place in Hansel?
	3	Which application is easier to insert the required data of the application (time and location for both application CatchIt and

		Hansel) to the bookmarked moment in CatchIt or to the tagged place in Hansel?
	4	How many moments did you bookmark?
	5	How many places did you tag?
CatchIt Vs Contacts	1	Which application is easier to deal with regarding bookmarking a moment in CatchIt or adding a new contact to the contact list in Contacts?
	2	Which application is easier to distinguish the target: the bookmarked moment in CatchIt or the contact in Contacts?
	3	Which application is easier to insert the required data of the application: time and location in CatchIt or the contact's first name, last name, phone number, etc (one of them is enough to save the new contact)?
	4	Which application is easier to add the optional data, photo, to the bookmarked moment in CatchIt or to the contact in Contacts?
	5	Which application is easier to add the optional data, textual data, to the bookmarked moment in CatchIt or to the contact in Contacts?
	6	Which application is easier to add the optional data, people, to the bookmarked moment in CatchIt or to the contact in Contacts?
CatchIt Feelings	1	What languages did you use while you bookmarked the moment?
	2	Did you find supporting of different languages in Catch useful?
	3	What are the capturing ways do you think that "CatchIt" should have?
	4	Which capturing ways in CatchIt did you like and would use in future?

	5	Which capturing ways in CatchIt would you not use and think can be left out?
	6	Can you think of another application that is similar to CatchIt?
	7	Are there any features in that application that you missed here?
	8	Do you have any recommendations or suggestions to improve CatchIt?

The aims of these categories of questions were to improve CatchIt in terms of ease of use and thus to feed Digital Parrot in an appropriate way (see the five categories of questions in Guided Interview in Appendix A).

6.1.5 Pilot Study

A pilot study with one participant was conducted prior to the main study. The related questions of the study, the questions of the initial questionnaire and the questions of the guided interview were reformulated as a result of using the pilot study. That led to making the questions more understandable and to enhance the results that were extracted from the main study. Also, the probable time taken for the interview was determined (40 minutes for all categories of questions).

6.2 Results

The results of the study are categorized into four categories: (1) demographic information of the participants, (2) preferred ways of capturing, (3) usability of CatchIt and (4) exploring the feelings and suggestions of using CatchIt. The first two categories came from the initial questionnaire while the second two categories came from the guided interview.

6.2.1 Demographic information of the Participants

Twelve international students participated in this study (6 females and 6 males). Eleven of these participants were from The University of Waikato (7 postgraduate students and 4 undergraduate students) while one PhD student was from another university.

The participants were from two different age groups (20 to 30 and 31 to 40 years) (see Figure 6.1). However, we did not differentiate between genders and age groups.

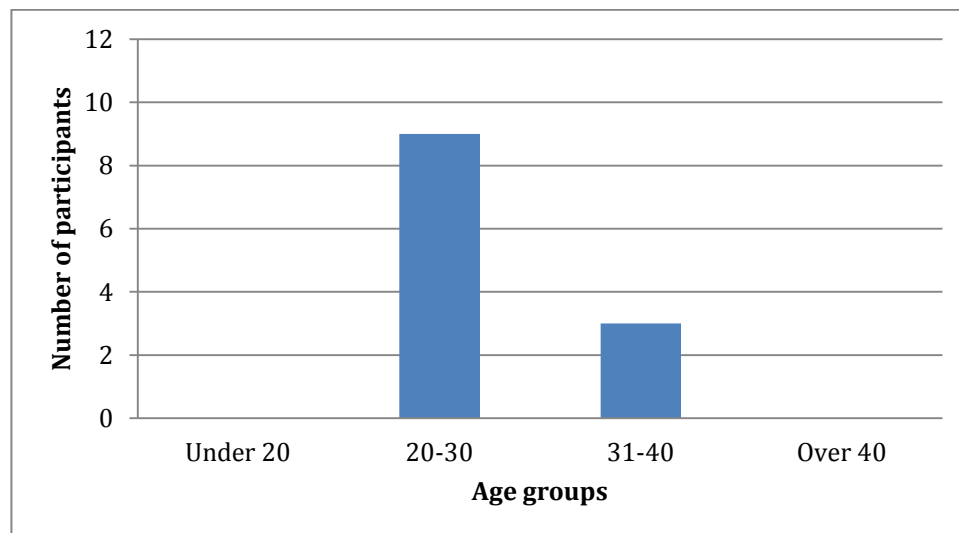


Figure 6.1: Age distribution of participants.

6.2.2 Preferred Ways of Capturing

Five situations are considered in the study to find out the preferred ways to capture the information on those situations in order to recall later. Each situation might need more than one capturing way. After discovering the preferred ways to capture the information of those five situations, result of the preferred way to capture the moment of life will be addressed.

6.2.2.1 The Situation of ‘Friend’s Birthday Party’

In the situation of a friend’s birthday party, there are two preferred capturing ways of the events of that day (see Figure 6.2). While all participants (12) proffered taking photos to capture the birthday parties, 10 participants preferred to do that by recording video. None of the participants preferred capturing the event by taking notes (neither on paper nor by using text message) or by recording audio. Also, there were no other capturing ways that not supported in the initial questionnaire.

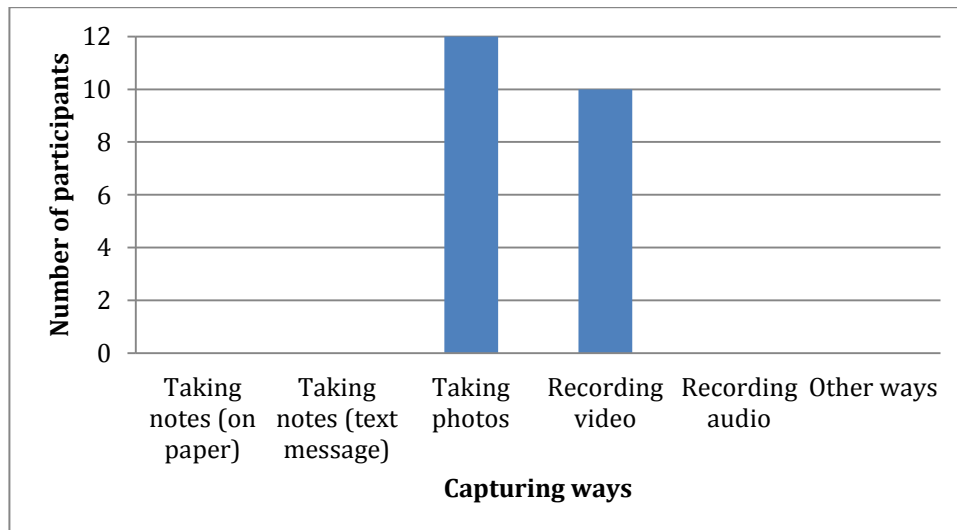


Figure 6.2: Frequency of the preferred capturing ways, to capture the events at friend's birthday party.

6.2.2.2 The Situation of ‘Setting in a Lecture’

Participants preferred to use several ways to capture important information of a lecture (see Figure 6.3). The most important capturing way was taking notes on paper; ten participants preferred this method. Recording audio and video, taking photos and taking notes in text message, were preferred to capture the lecturer notes but by 5 participants, 3 participants, 2 participants and 1 participant respectively. One participant preferred an additional capturing way, which is not supported in the initial

questionnaire. This participant preferred to capture the lecturer notes by using his/her own laptop.

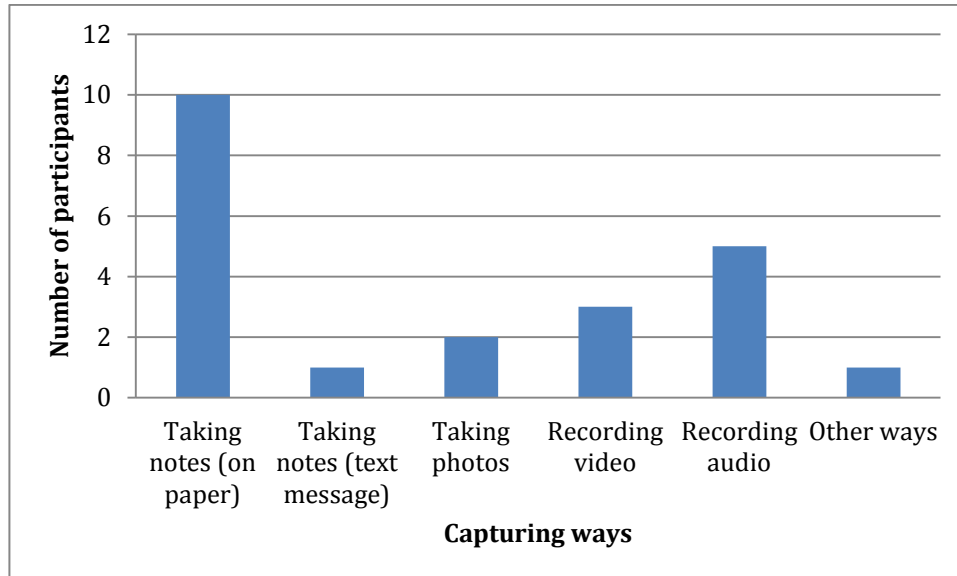


Figure 6.3: Frequency of the preferred capturing ways to capture the important notes in a lecture.

6.2.2.3 The Situation of ‘Shopping from Supermarket’

Three capturing ways were preferred in capturing the shopping list of grocery that has to be bought from the supermarket (see Figure 6.4). Ten participants preferred taking notes on paper to do that while 8 participants preferred to message the shopping list to themselves and 3 participants took a photo of the textual shopping list. There was no usage of recording video or audio or using other ways to capture the shopping list.

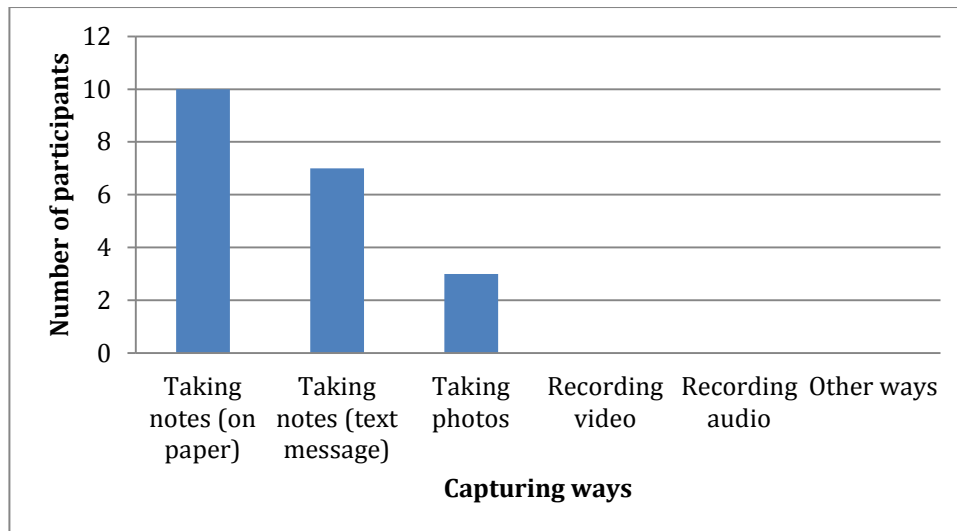


Figure 6.4: Preferred capturing ways to capture the shopping list.

6.2.2.4 The Situation of ‘Meeting with a Supervisor/Advisor’

The preferred ways of capturing the meeting with a supervisor/ advisor are shown in Figure 6.5. Taking notes on paper was chosen 11 times to be the best way to capture the meeting with supervisor/advisor. Recording audio came next with 8 times; taking notes in text message and recording audio with 2 times for each. On the other hand, taking photos or other ways of capturing were not used to capture the meeting with a supervisor/ advisor at all.

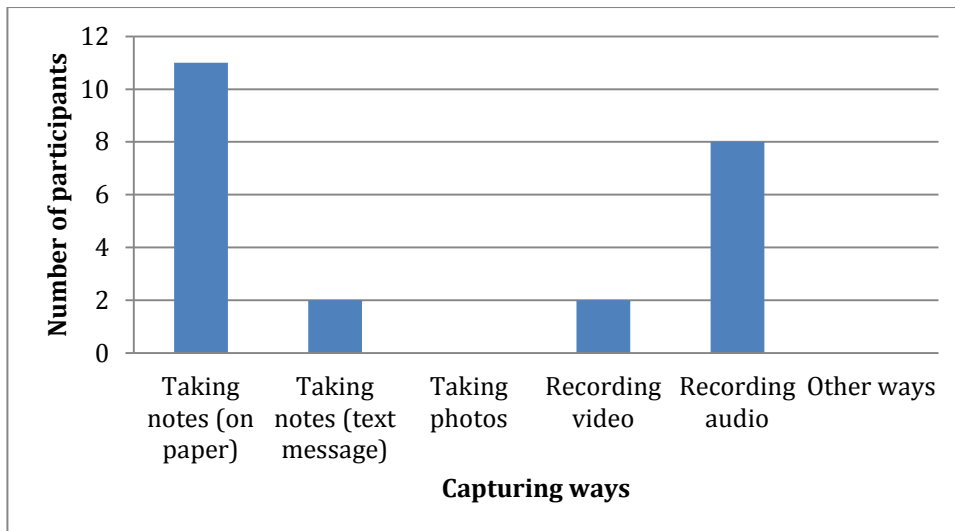


Figure 6.5: Frequency of the preferred capturing ways to capture the meeting with a supervisor/ advisor.

6.2.2.5 The Situation of ‘Interested in a Product’

All participants of the study depended largely on taking photos of the interesting product as it is shown in Figure 6.6. The second choice was taking notes on paper by 4 participants. Taking notes on text message was chosen twice and recording audio once.

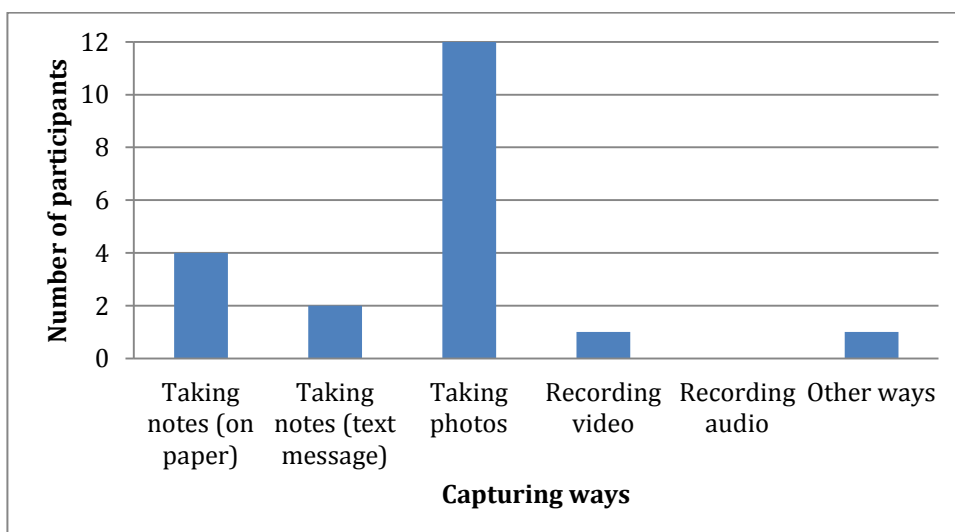


Figure 6.6: Frequency of the preferred ways to capture the information about a product.

6.2.2.6 The Preferred Ways to Record Moments in Life

In general, to record moments in the participants' lives, there were three ways of recording that were preferred. The most popular ways of capturing were taking photos (mentioned 12 times) and recording video (mentioned 11 times). They were much more used than taking note using pen and paper, which was recommended by two participants (see Figure 6.7). On the contrary, neither taking notes by using text message nor recording audio was used to record details of the moments in the participants' lives. Also, there were no other ways of capturing that were used for the same purpose.

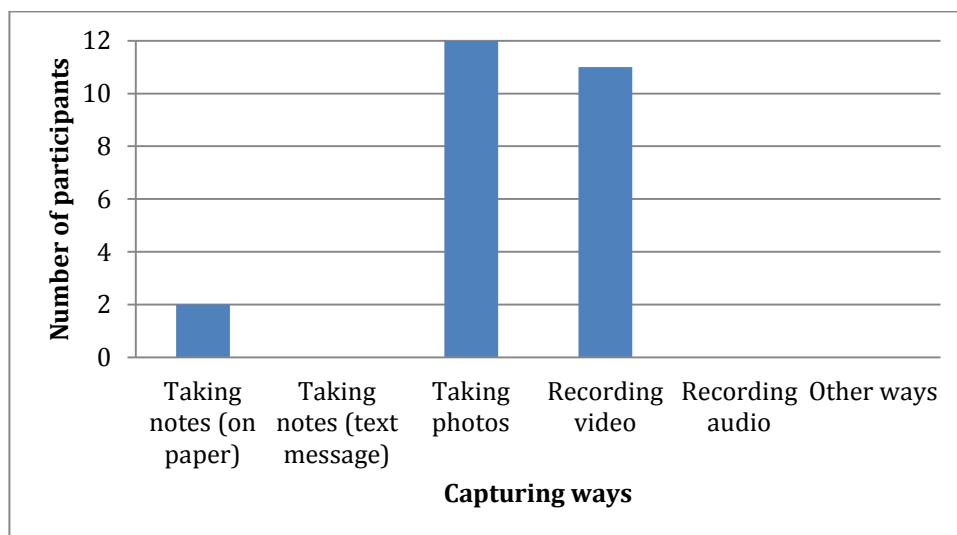


Figure 6.7: Frequency of the preferred capturing ways to record moments in participants' lives.

6.2.3 Usability of CatchIt

The results of the interviews with the participants are divided into five parts: the ease of using CatchIt, the ease of using Hansel, comparing CatchIt with Hansel, comparing CatchIt with Contacts and then extracting the feelings and the recommendations of using CatchIt. The results will help us to investigate further usability problems of the

user interface of CatchIt, which application between CatchIt and Hansel is easier to be used as a feeder of Digital Parrot and how can we make CatchIt easier to be used as a feeder of Digital Parrot by matching more of the user needs.

6.2.3.1 Ease of using CatchIt

To test the ease of using CatchIt, the results are classified into four categories. The first category is the result of bookmarking a moment. The second category is changing the required data of CatchIt (time and location). The third category is inserting the optional data of CatchIt (the label of the moment, notes, photos, videos, audios and people). The fourth category is the result of moving between the windows of the application.

Bookmarking a Moment

The ease of bookmarking a moment was either *very easy* or *easy*. In fact, five participants out of twelve considered the bookmarking a moment was *very easy* (see Figure 6.8). However, the majority of participants (7 participants) said that the bookmarking a moment in CatchIt was *easy*. Indeed, they did not choose *easy* to be as the level of ease of bookmarking the moment due to the large number of processes that happened when the moment was bookmarked (message box and moving to another window).

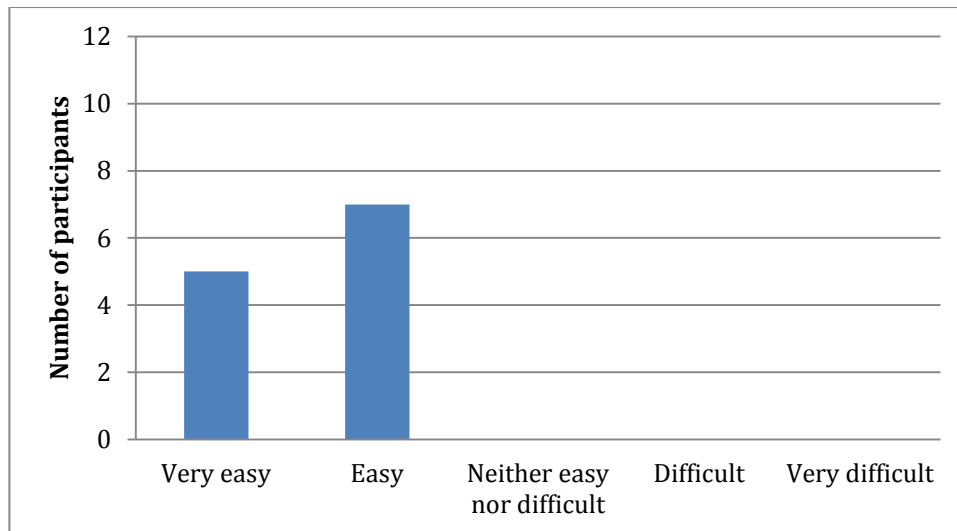


Figure 6.8: Ease of bookmarking a moment using CatchIt.

Changing the Required Data of CatchIt

The required data in CatchIt to bookmark the moment are time and location. The data of both time and location are captured automatically by the system. However, the user can change the captured values whenever they wish. In the study, four participants changed the values of time and location whereas eight participants did not do so because the changes were neither needed nor considered (see Table 6-5). Changing of time was not need for 7 out of 8 participants and was not recognized by 1 out of 8 participants. On the other hand, changing the value of the target location was not needed for 5 out of 8 participants and not recognized for 3 out of 8 participants.

Table 6-5: Number of participants who did not change the required data of CatchIt and the frequency of the reasons that prevented the participants to do so.

		# Participants	Reasons	
			Not needed	Not recognized
Required data	Time	8	7	1
	Location	8	5	3

Participants who changed either the time or the location (4 participants) found the changing *very easy*, *easy* or not decided (see Figure 6.9). Changing time was *very easy* and *easy* by the same number of participants (two). The location was *very easy* to be changed for two participants, *easy* to be changed for one participant and *neither easy nor difficult* for one participant as well.

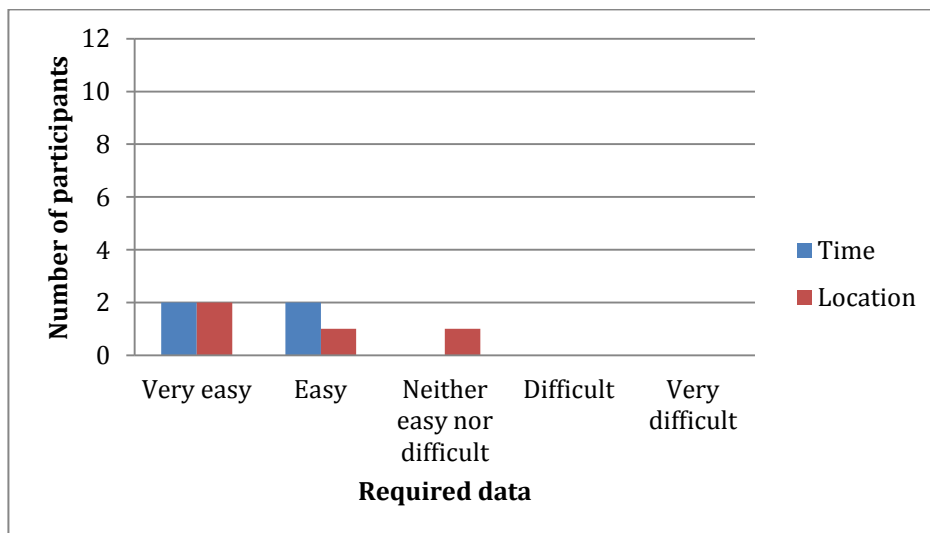


Figure 6.9: Ease of changing the required data of CatchIt, time and location; 4 participants changed the values of the required data.

Adding the Optional Data of CatchIt

The optional data of the bookmarked moment are: the moment's label, notes, photo, video, audio and people. In the study, all the participants added notes and photos to their moments while people were tagged by 11 participants (see Figure 6.10). Adding notes and video came next in terms of usage by 9 and 8 participants respectively then recording audio (only by 5 participants).

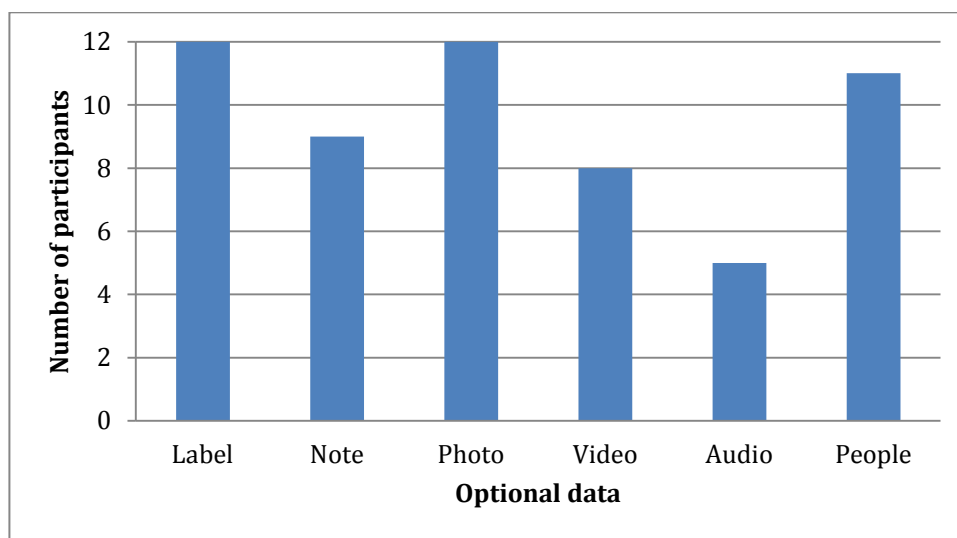


Figure 6.10: Number of participants who added the optional data of CatchIt.

There were various reasons that prevented the participants to add the optional data. As Figure 6.10 shows, all participants added labels and took photos in their bookmarked moments which means that they did not have any reason that prevented them to do so. In contrast, the rest of the optional data were not used because of at least one reason (see Table 6-6).

Table 6-6: Number of participants who did not add the optional data of CatchIt and the frequent of the reasons that prevented them to do so.

		# Participants	Reasons						
			Long time	No need	More capacity	Limited time	Not effective	Not clear	Shortage details
Optional data	Label	0	-	-	-	-	-	-	-
	Notes	3	2	1	-	-	-	-	-
	Photo	0	-	-	-	-	-	-	-
	Video	4	2	1	1	1	1	-	-
	Audio	7	-	3	-	-	4	-	1
	People	1	-	-	-	-	-	1	-

The wider range of reasons for not adding the optional data of CatchIt was adding the video. Four out of twelve participants did not add videos to their moments, because of five reasons that prevented them to do so. Taking long time was chosen twice as the main reason of not adding videos. Not needing to add video, the limited time of each video (1 minute), ineffectiveness of using video to remember the moment and the consumption of a large amount of the phone's capacity were other causes that prevented the participants to record the moment in video format, with same frequency (once for each reason).

The most ignored optional data was a recording audio. Seven out of twelve participants did not record any audio during their bookmarked moments. Some participants believed that the recording audio is either not an effective way to remember their moments (mentioned four times) or the audio record has less details of the target moment – it does not cover the visual data (mentioned three times). Not needing to record an audio in the bookmarked moment was another reason of ignoring this optional data (mentioned once).

While three out of twelve participants did not take notes of their bookmarked moments either due to the long time they needed if they would like to add notes (mentioned twice), or due to the fact that optional data was not needed (mentioned once).

Only one participant did not tag any person in any of the bookmarked moments because of one reason. The participant said that tagging people in the bookmarked moment was not a clear way.

The participants who did add the optional data of CatchIt (label, notes, photos, video, audio, and people) had different points of view on the ease of capturing each of the optional data. All participants labeled their target moments. The majority of participants (7 participants) said that it was *very easy* to label the moment while the rest of the participants (5 participants) said that it was *easy* to do so (see Figure 6.11).

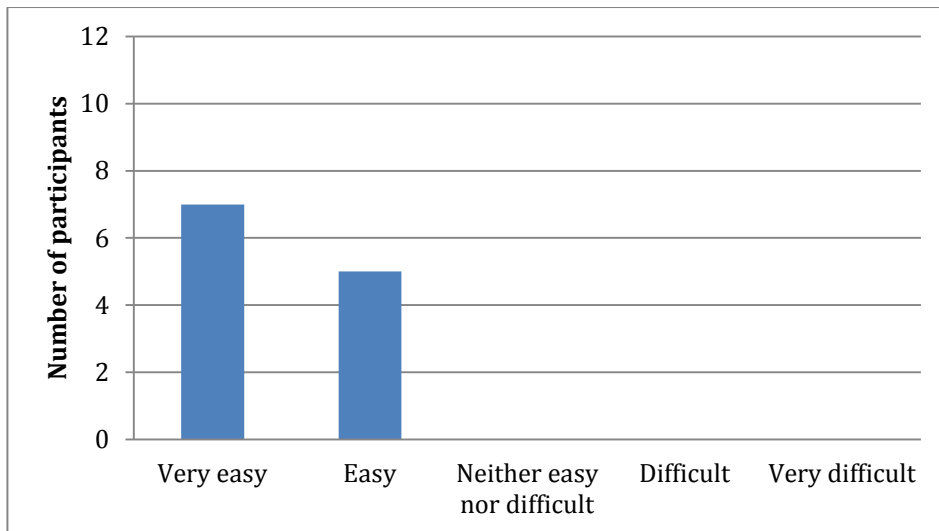


Figure 6.11: Ease of adding the optional data, label.

Similarly, 7 out of 9 participants who took notes in the target moments found taking notes in CatchIt *very easy* while 2 out of 9 participants found that *easy* (see Figure 6.12).

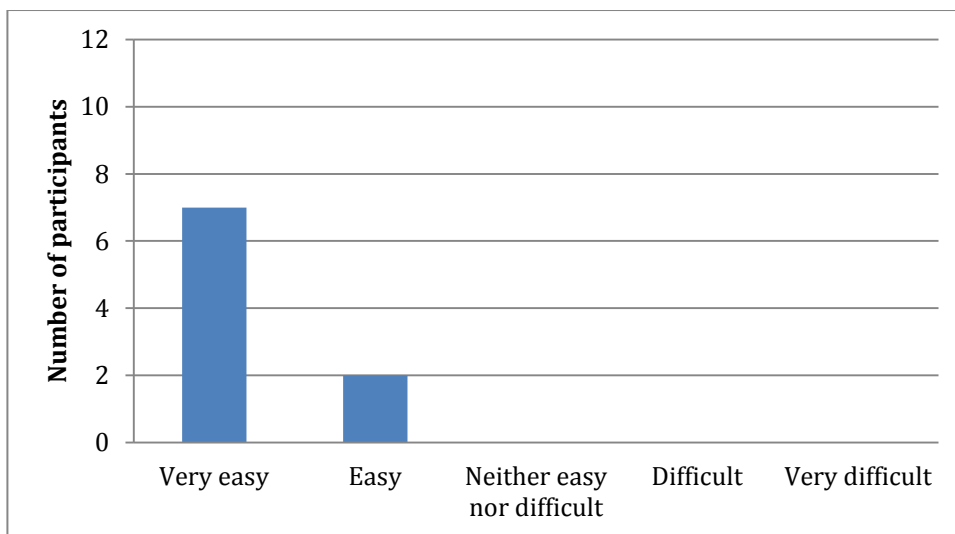


Figure 6.12: Ease of adding the optional data, notes.

All participants added photos to their bookmarked moment. Six participants who tried taking photos of the bookmarked moments found that *very easy*. Additionally, while 5

participants said taking photos was *easy*, one participant could not decide if taking photos was easy or difficult (see Figure 6.13).

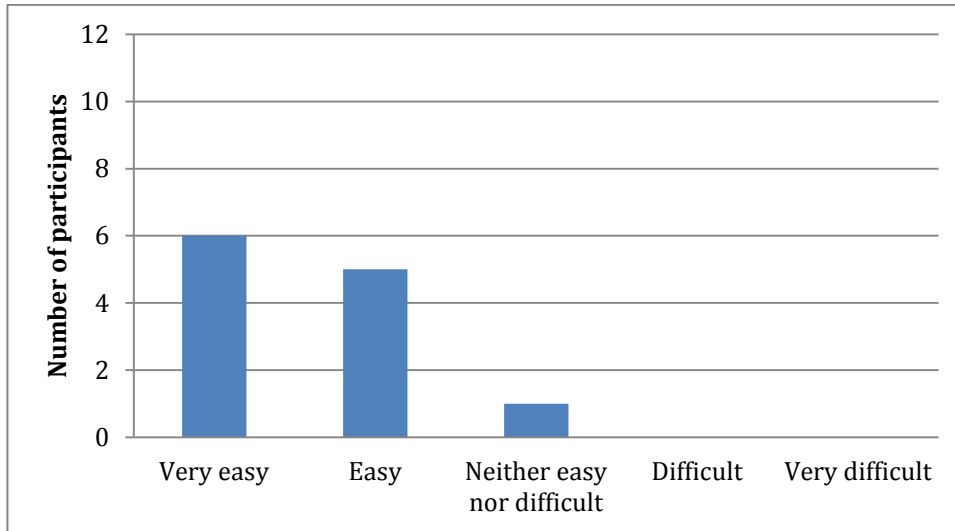


Figure 6.13: Ease of adding the optional data, photos.

The total number of participants who recorded video of their target moments was eight. The number of participants who found that recording video was *very easy* was the same as the number of participants who found doing that *easy*. That number decreased to two participants who could not decide if recording video in CatchIt was easy or difficult (see Figure 6.14).

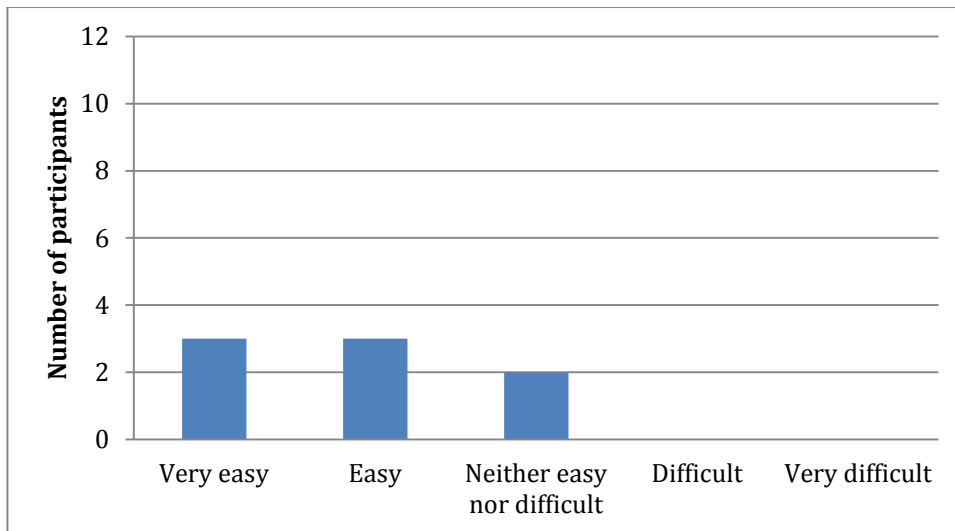


Figure 6.14: Ease of adding the optional data, video.

Recording audio has less popularity than other optional data since only 5 out of 12 participants recorded audio of their moments. Despite this, recording audio was *very easy* for 3 participants and *easy* for 2 participants (see Figure 6.15).

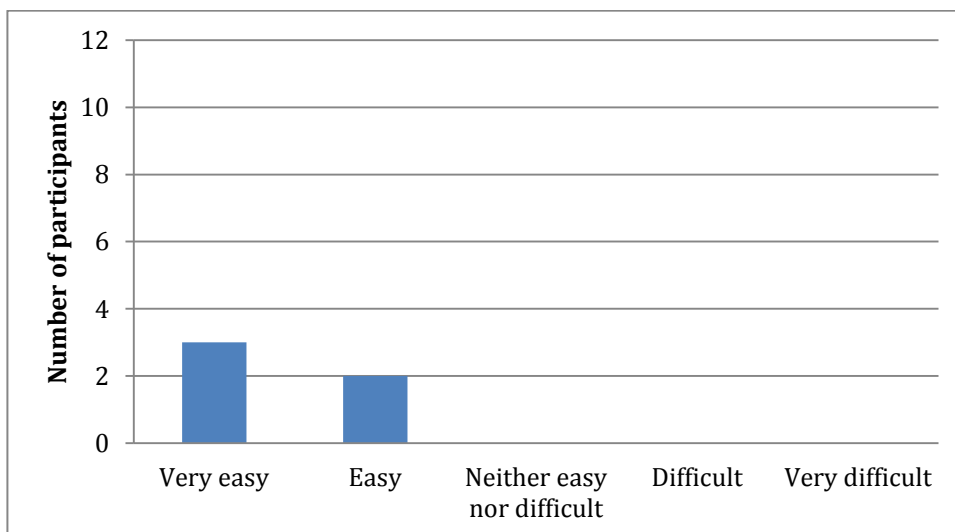


Figure 6.15: Ease of adding the optional data, audio.

Almost all the participants tagged people in their bookmarked moment (11 out of 12 participants). While the majority of those participants (6 out of 11 participants) ranked

the level of ease of tagging people in CatchIt as *easy*, four participants believed that tagging people was *very easy*. However, one participant said that tagging people was *very difficult* (see Figure 6.16).

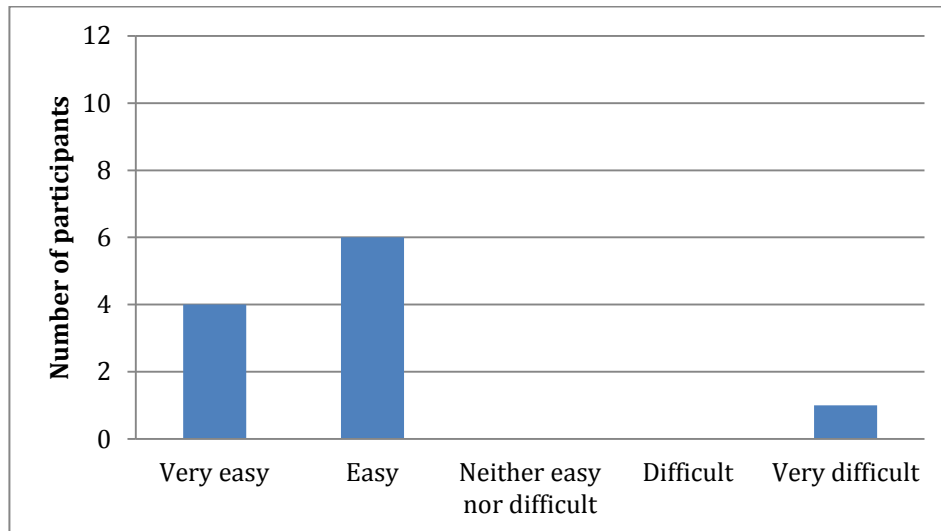


Figure 6.16: Ease of adding the optional data, people.

Moving between the Windows of CatchIt

Moving between windows of Cachet is allowed to the user. However, not all participants found going to the previous window very easy. While seven of the participants rated moving back to the previous window as *very easy*, four participants said it was *easy* (see Figure 6.17).

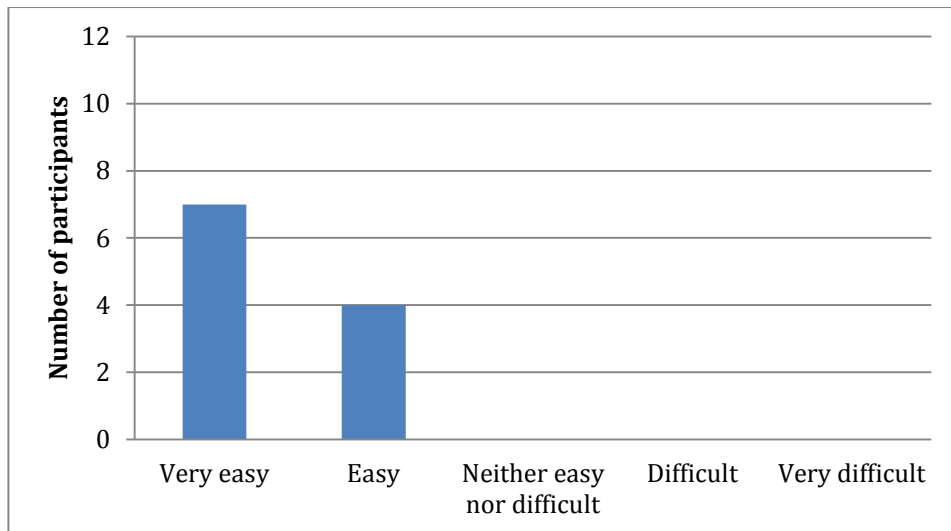


Figure 6.17: Ease of moving to the previous window in CatchIt.

6.2.3.2 Ease of using Hansel

Unlike CatchIt, the ease of using Hansel to tag locations lays in the ease of tagging the place, the ease of adding the optional data (a label) and the ease of moving between the windows of the application.

Tagging a Place

Although tagging a place in Hansel occurs by pressing a button, opinions differs about the ease of tagging the target place. Seven participants said it was very easy to tag a place, three participants chose the level of ease as easy and two participants could not decide if tagging the place was easy or difficult (see Figure 6.18).

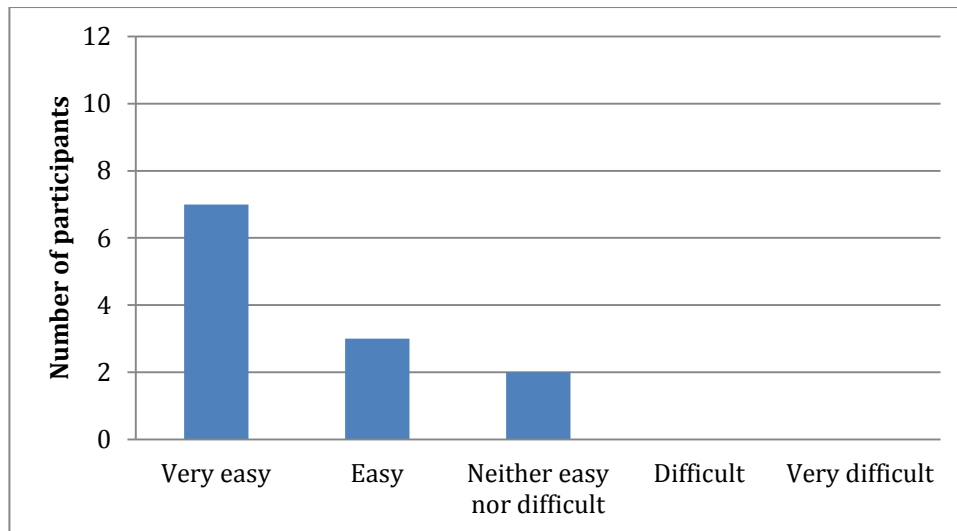


Figure 6.18: Ease of tagging a place using Hansel.

There were two reasons that made the participants not rating tagging the place as very easy (3 participants said easy; 2 participants said neither easy nor difficult). The first reason was the lack of reaction when the place is tagging. There was no sign when the place was tagged (mentioned twice). The second reason was that tagging a place in Hansel needs the Internet connection (mentioned once).

Adding the Optional Data of Hansel

Unlike CatchIt, there is one optional data in Hansel. It is adding a label to the tagging place. Eleven out of twelve participants added a label. The participant who did not add the label said that they did not recognize the purpose of the text field because there is no instruction or label explaining what the text field is for. However, the largest number of participants, 6 participants, said that it was very easy to add a label while the minority, just one participant, said it was easy. In addition, 2 participants said that it was neither easy nor difficult. Similarly, 2 participants said it was difficult to do so, (see Figure 6.19).

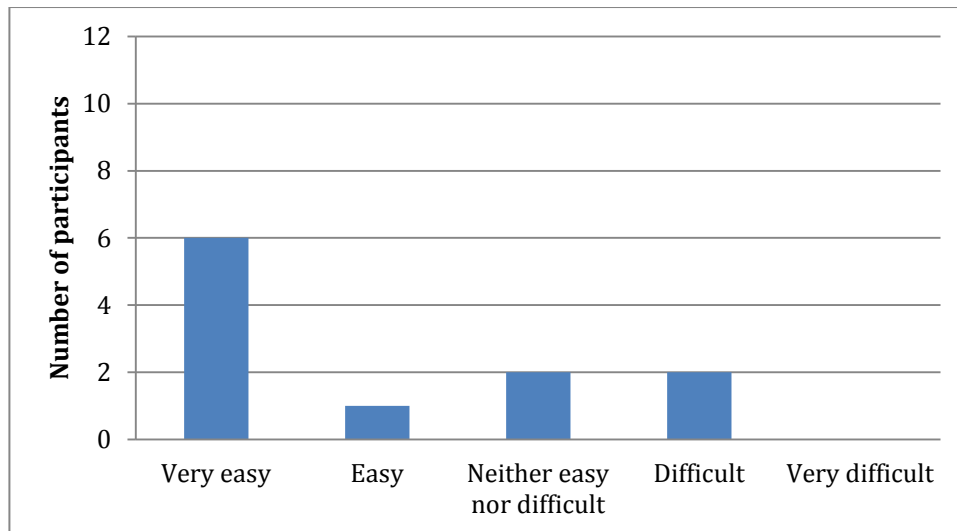


Figure 6.19: Ease of adding the optional data of Hansel.

Moving between the Windows of Hansel

Not all participants moved to the previous window of Hansel confidently because there is no back button in all the windows. Consequently, the participants would have to know exactly which window of the application they were in. Five participants felt moving back to the previous window was very easy and four participants said it was easy. Two participants could not decide the ease level of moving back to the previous window and one participant did so with difficulty (see Figure 6.20).

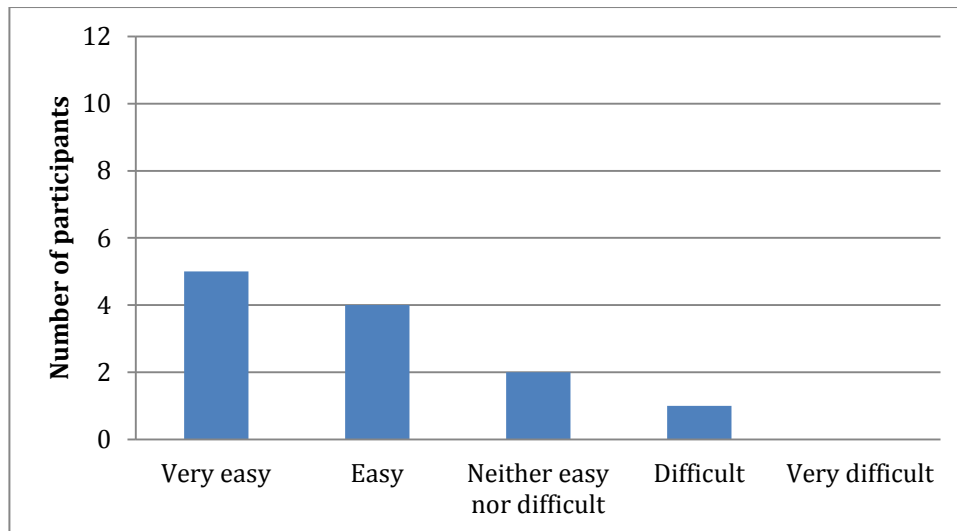


Figure 6.20: Ease of moving back to the previous window in Hansel.

6.2.3.3 CatchIt Vs Hansel

To compare between the ease of using CatchIt and Hansel, four points will be covered: the ease of dealing with regarding the main function of the applications, the ease of distinguishing the target, the ease of adding with the required data of both applications and the frequency of using each application.

Dealing with Regarding the Main Function of the Applications

The main function of the applications CatchIt and Hansel is bookmarking a moment and tagging a place respectively. The comparison between the ease of bookmarking a moment and tagging a place, with the reasons behind the decisions, are presented in Table 6-7. Overall, Hansel was easier than CatchIt in terms of dealing with regarding the main function of the application.

Table 6-7: Result of the compression of the ease of dealing with regarding the main function between CatchIt and Hansel. The result involves the Frequency of reasons that made participants prefer the application.

		# Participants	Reasons						
			Visibility	Appropriate feedback	Less information	No constraints	Less effort	Few steps	
Applications	CatchIt	5	1	5	-	1	-	-	
	Hansel	7	1	-	-	-	1	5	
	Same	0	-	-	-	-	-	-	

Hansel was easier to tag a place for 7 out of 12 participants; 5 times because Hansel has few steps to tag a place (if the Internet connection is not considered), once because Hansel displays the contents of the main window easily (a text field of the label and a button for tagging are in the same window) and once because less effort is required of the participants to tag a place.

On the other hand, owing to three different reasons, dealing with regarding the main function of CatchIt, bookmarking a moment, was easier for five out of twelve participants. All of those participants could recognize that their target moment has been booked because of the appropriate feedback that CatchIt gives to the user when the user bookmark the moment (popping up a message box and moving to another window). In addition, one of the five participants liked better bookmarking a moment in CatchIt because of the less limitation CatchIt has such as the ability to bookmark the moment with no Internet connection and the ability of modifying the required data of the moment (time and location). Also, one of the five participants found

bookmarking a moment easier than tagging a place because of the displaying way of the contents that CatchIt has (the label on the button of bookmarking is easy to understand the purpose of it).

Distinguishing the Target

In comparison of the ease of distinguishing the target moment in CatchIt and the ease of distinguishing the target place in Hansel, CatchIt was much easier for distinguishing the target according to 10 out of 12 participants (see Table 6-8). All participants believed that the amount of data that CatchIt shows for each bookmarked moment helped them to distinguish the target bookmarked moment. Additional reason, which affected the decision of 2 out of 10 participants, was the lack of constraints that display the list of moments as CatchIt does not need the Internet connection to show the list of the bookmarked moments. Only one participant added one more reason. It was that kind of data that CatchIt displays for each moment was more effective to distinguish the target moment.

Table 6-8: The result of the compression of the ease of distinguishing the target between CatchIt and Hansel. The result involves the frequently of the reasons that made the participants preferred to choose the application.

		# Participants	Reasons			
			Visibility	More data	Effectiveness	No constraints
Application	CatchIt	10	-	10	1	2
	Hansel	2	2	-	-	-
	Same	0	-	-	-	-

On the other hand, Hansel was selected by 2 out of 12 participants to be the easier application to distinguish the target. They selected Hansel only because of displaying the contents of the main window of Hansel that shows just the label of place (not messy).

Adding the Required Data

Although the required data, time and location of both applications, CatchIt and Hansel, are captured automatically, the ease of inserting the required data differed between these applications. While 3 participants found the ease same in the two applications because of the auto-capturing of the required data, 7 participants preferred CatchIt and 2 participants preferred Hansel (see Table 6-9).

Four of the participants who preferred CatchIt considered the option of the ability of inserting the time and location manually. The accuracy of the captured location was

enough reason for 3 out of 7 participants and the lack of constraints that make the application captures the required data automatically was additional reason for the same number of participants (3 participants). However, Hansel was easier to capture the required data for two participants due to fewer numbers of buttons that Hansel has (just one button in the main window).

Table 6-9: Result of the comparison of the ease of adding the required data between CatchIt and Hansel. The result involves the frequency of the reasons that made the participants prefer the application.

		# Participants	Reasons				
			More options	Accuracy	Visibility	No constraints	Auto- captured
Application	CatchIt	7	4	3	-	3	-
	Hansel	2	-	-	2	-	-
	Same	3	-	-	-	-	3

Frequency of Using the Application

CatchIt and Hansel were used in different frequencies although these applications were used for the same length of time (one month). The number of the bookmarked moments in CatchIt was much more than the number of the tagged places in Hansel. There were 91 bookmarked moment whereas the number of the tagged places was 64.

6.2.3.4 CatchIt Vs Contacts

The comparison between CatchIt and Contacts depended on these aspects of the applications that are similar. Dealing with regarding the main function of the application, distinguishing the target among the stored information, inserting the required data, inserting the optional data and the frequency of using each application were considered in the comparison.

Dealing with Regarding the Main Function of the Applications

While the main function of CatchIt is to bookmark a moment, the main function of Contacts is to add a contact details into mobile phone. In the study, dealing with regarding the main function of CatchIt was much easier than dealing with regarding the main function of Contacts.

In fact, the majority of participants (11 participants) found bookmarking a moment easier and one participant found both applications have the same ease of dealing with regarding the main function of the applications (see Table 6-10). The reasons that played a key role in the selection of CatchIt as an easier application to deal with regarding the main function of the application, bookmarking a moment, were that CatchIt has fewer steps to bookmark the moments (mentioned 4 times), captures the moment automatically (mentioned 4 times) and the user does not spend much time to bookmark the target moment (mentioned 3 times). The participant, who decided that CatchIt and Contacts have the same level of ease, found both applications needed no effort to bookmark or add contact of the target moment or person respectively.

Table 6-10: Result of the compression of the ease of dealing with regarding the main functions of CatchIt and Contacts. The result involves the frequency of the reasons that made the participants prefer the application.

		# Participants	Reasons			
			Few steps	Auto-captured	Less time	No effort
Application	CatchIt	11	4	4	3	-
	Contacts	0	-	-	-	-
	Same	1	-	-	-	1

Distinguishing the Target

Distinguishing the target person among the people in the contact list of Contacts was easier for 9 participants than distinguishing the target moment among the bookmarked moments (see Table 6-11). Various reasons were behind choosing Contacts to be easier for distinguishing the target. The most frequent reason was the familiarity of searching the person contact (mentioned 7 times) and then the ability of searching the target person's details among the contact list (mentioned 6 times). Two out of nine participants found the sorting way of the contact list (alphabetical sort) easier to distinguish the target contact in Contacts. Additional reason (by one participant) was the effectiveness of remembering the person's name.

Table 6-11: Result of the compression of the ease of distinguishing the target between CatchIt and Contacts. The result involves the frequency of the reasons that made the participants prefer the application.

		# Participants	Reasons			
			Effectiveness	Familiarity	Ability to search	Sorting
Application	CatchIt	3	2	-	-	-
	Contacts	9	1	7	6	2
	Same	0	-	-	-	-

Adding the Required Data

Adding the required data of both applications CatchIt and Contacts was considered in the compression of ease of using the applications (see Table 6-12). The amount of the required data and the auto-capturing of that data in CatchIt were the significant reasons of choosing 4 out of 10 participants (4 participants for each reason) CatchIt as easier application to add the required data. Moreover, inserting the required data in CatchIt uses less time than Contacts according to 2 out of 10 participants. Similarly, 2 out of 10 participants said that CatchIt has few steps to insert the required data.

Table 6-12: Result of the compression of the ease of adding the required data between CatchIt and Contacts. The result involves the frequency of the reasons that made the participants prefer the application.

		# Participants	Reasons					
			Less time	Auto- captured	Less information	Familiarity	Less effort	Few steps
Applications	CatchIt	10	2	4	4	-	1	1
	Contacts	1	-	-	-	1	-	-
	Same	0	-	-	-	-	-	-

In contrast, one of the 12 participants decided Contacts to be easier for the same purpose, inserting the required information because of the familiarity of the required data that Contacts has.

Adding the Optional Data

Comparing the ease of adding the optional data of CatchIt and Contacts was limited by the textual data, photos and people, the common optional data, (see Figure 6.21). Adding any textual information in CatchIt was chosen by 8 out of 12 participants to be an easier way to insert the textual data in the *Notes* section of the application while adding the textual information in Contacts such as the email address and the physical address of the target person was easier for 3 out of 12 participants. One participant felt that adding the optional textual information in CatchIt is the same as in Contacts.

Adding the photos in CatchIt was not as easy as Contacts. While 4 out of 12 participants chose adding photos in CatchIt easier than Contacts, the majority of the participants (8 participants) found adding photos in Contacts easier.

All participants found tagging people in CatchIt was easier than Contacts.

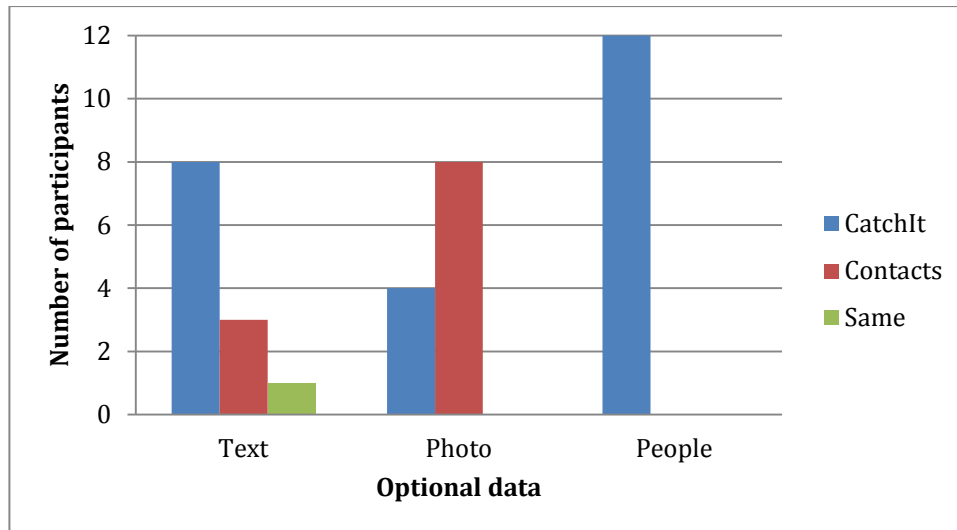


Figure 6.21: Frequency of choosing the preferred application (CatchIt or Contacts) to add the optional data.

There were several motivations that made the participants argue about the ease of adding the selected optional data between the applications CatchIt and Contacts.

These reasons are shown in Table 6-13.

Table 6-13: Proportion of reason of the ease of inserting the optional data of the applications CatchIt and Contacts.

Application	Optional data	# Participants	Reasons					
			Few steps	More options	Less time	Easy to find	Organized	Visibility
CatchIt	Text	8	3	-	1	-	-	5
	Photo	4	4	-	-	-	-	-
	People	12	4	-	6	5	-	-
Contacts	Text	3	1	-	-	-	3	-
	Photo	8	1	7	-	-	-	-
	People	0	-	-	-	-	-	-

Participants who did choose CatchIt as an easier application to insert the textual data had some reasons (8 participants). The most frequent reason (mentioned 5 times) was the way of displaying the textual data in one place, the “Notes” section of CatchIt. Moreover, adding the textual data in the “Notes” window of CatchIt had few steps according to 3 out of 8 participants. The third reason was applied to one participant when the participant spent less time to add the textual data in CatchIt. On the other hand the application Contacts was easier to add the textual data for some participants because adding this sort of data in this application was more organized; each data has a specific place (mentioned three times). In addition, the fewer steps for adding textual data in Contacts were found to be easier by one participant.

Adding photos of the moment was easy because CatchIt has fewer steps to add a photo as all four participants said. In contrast, Contacts was easier than CatchIt in taking photos. Seven out of eight participants said Contacts has more options to take a photo, directly from the built-in camera or by choosing a sorted photo from the device. Moreover, one out of eight participants said that Contacts has fewer steps to add a photo than CatchIt.

CatchIt has an easier way to tag people in the bookmarked moment due to three reasons: (1) six participants spent less time to tag a person in CatchIt; (2) it was easy for 5 participants to find the place that the participants could tag a person from in CatchIt and (3) four participants said CatchIt has fewer steps to tag a person.

6.2.4 Feelings and Suggestions

Exploring the participants' feeling of using CatchIt and taking the feedback from the participants on using CatchIt will lead to improving CatchIt in terms of the ease of use and in terms of covering more of the user needs. Overall, the participants' feeling of using CatchIt concentrated on the capturing ways of the surrounded information of the moment while the participants' suggestions covered the overall usage of CatchIt as well.

Capturing by Using Different Languages

Half the participants capture the textual information of their moment by using another language besides English. Most of participants however found using more than one language to insert the textual information either very useful (8 participants) or somewhat useful (3 participants). Just one participant could not decide whether

supporting multi-languages to insert the textual data useful or not useful (see Figure 6.22).

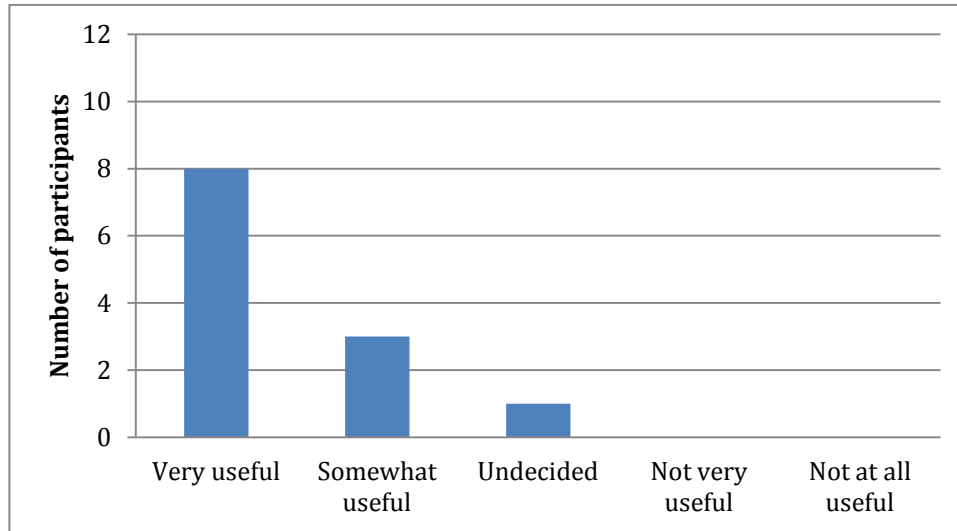


Figure 6.22: Usefulness of supporting multi-languages to insert the textual information in CatchIt.

Figure 6.23 shows the benefits of using more than one language to insert the textual information. Five participants liked supporting different languages because of the ease of describing the event by using the first language, and 3 participants said everyone can use the application if the application supports more than one language. The ease of remembering the event and reducing the spelling mistakes by using the first language of the user and practicing a second language were reasons given by one participant.

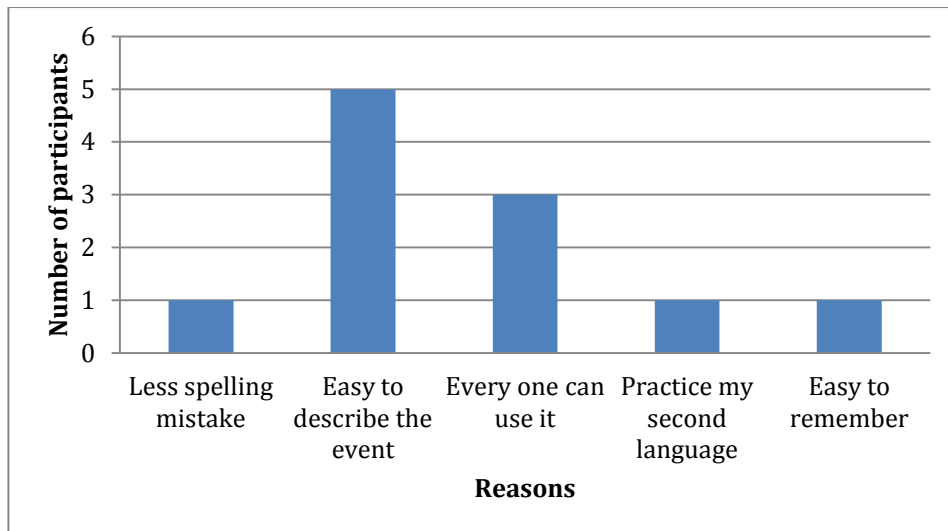


Figure 6.23: Frequency of the reasons behind the usefulness of supporting different languages to insert the textual information in CatchIt.

Missing Capturing ways

There were several capturing ways that three participants missed in CatchIt. These participants found that capturing data from the phone, such as notes, photos and people (from the contacts list) was missing in CatchIt as Figure 6.24 and 2 of them missed capturing data from at least one of the social networks.

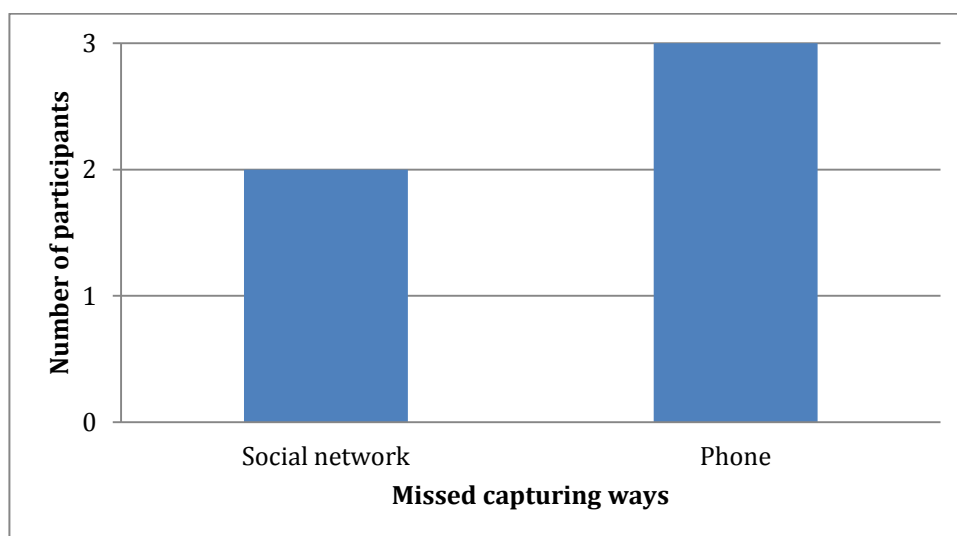


Figure 6.24: Frequency of the missed capturing ways in CatchIt (3 out of 12 participants asked for new capturing ways).

Capturing Ways to Keep Vs Capturing Ways to Leave

There were various opinions about the capturing ways that CatchIt has to keep and that CatchIt has to leave. Taking photos was the most preferred way to capture the target moment (for 11 participants) and was not recommended to be left from CatchIt at all, whereas recording audio was the most undesirable capturing way for 7 participants as it is shown in Figure 6.25 and in Figure 2.26 respectively.

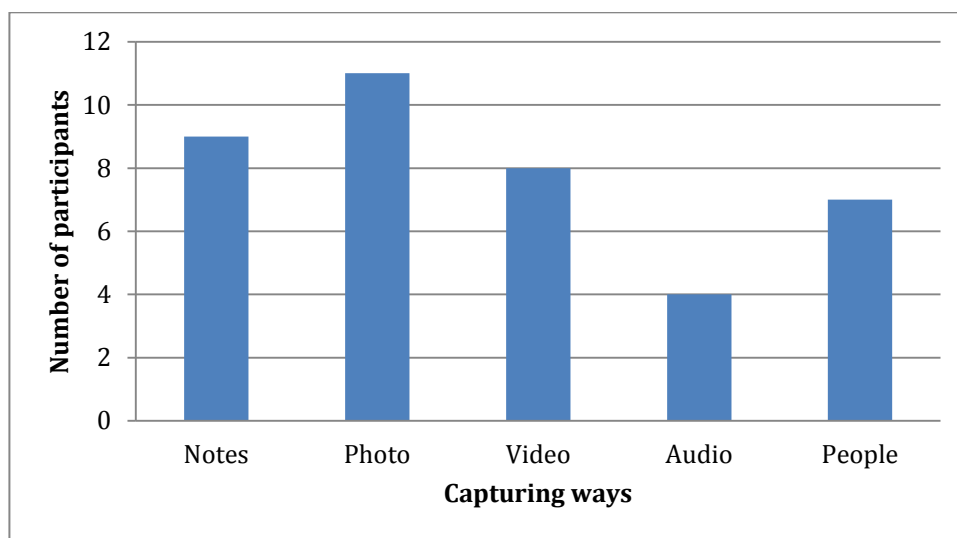


Figure 6.25: Frequency of choosing the capturing ways of CatchIt to be kept.

Taking notes was preferred to be kept in CatchIt for 9 participants and to be left out for 1 participant, while recording video was preferred by 8 participants to be in CatchIt; this means more participants (3) preferred to leave recording video out. Tagging people had similar decisions from participants in terms of keeping it (7 participants) or leaving it out (2 participants).

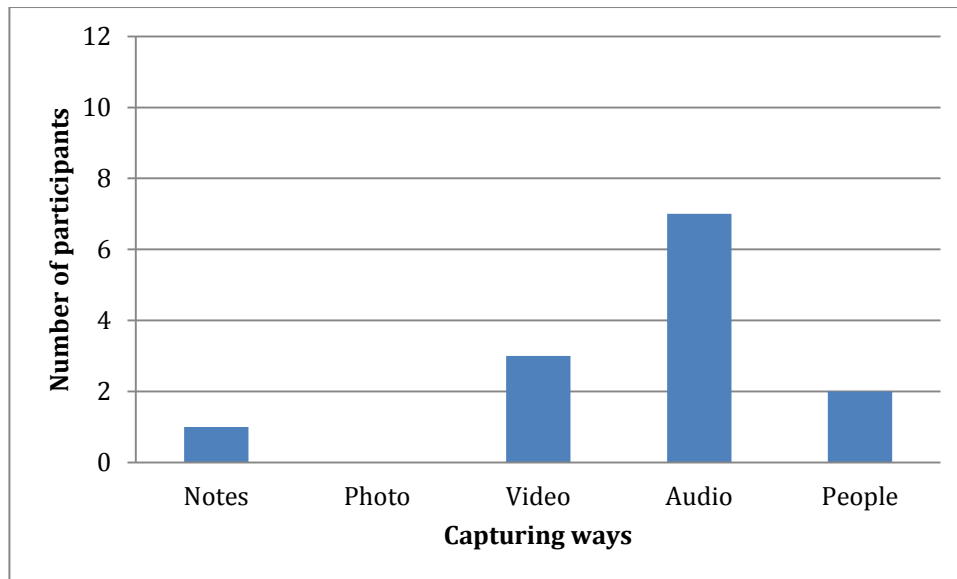


Figure 6.26: Frequency of choosing the capturing ways of CatchIt to be left out.

Similar Applications and Missing Features

Two built-in iPhone applications, *Note* and *Calendar*, and one desktop and mobile application *Facebook*, were selected as similar application to CatchIt. These applications have some features that CatchIt does not have. CatchIt does not have the ability of sharing as *Note*, *Calendar* and *Facebook* have. In addition, *Calendar* and *Facebook* have the ability to show the event on a calendar, which CatchIt cannot do.

Further Suggestions

Participants' suggestions are show in four groups. The largest group of suggestions based on additional features the CatchIt should have 22 suggestion (see Figure 6.27). The second group of suggestions concentrated on the displaying way of the content in CatchIt with 7 suggestions. The third largest group was for the undesirable features of CatchIt, with 4 suggestions. The last group (and smallest) of suggestions was for the missing information that CatchIt should have, with only two suggestions.

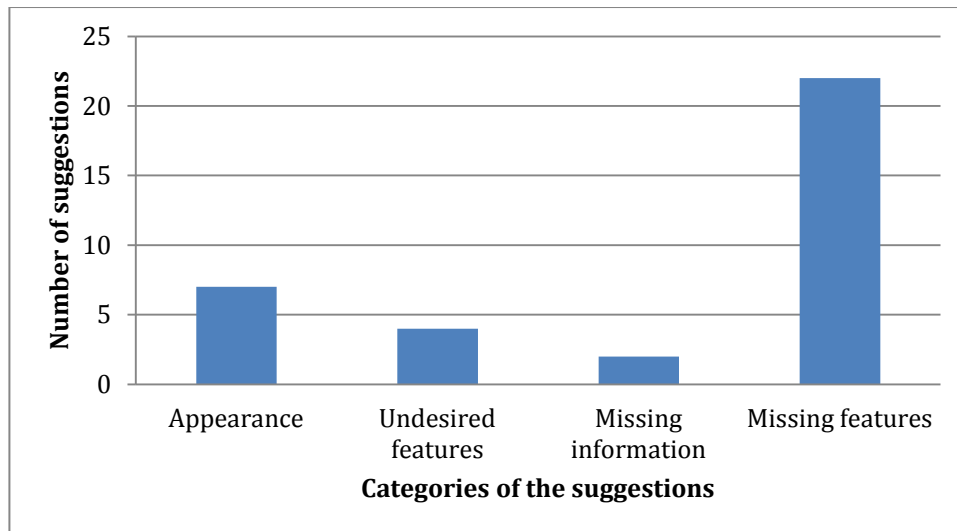


Figure 6.27: Frequency of the participants' suggestions of CatchIt for each category.

The missing features were various (see Table 6-14). Six participant liked to share their bookmarked moment with friends who have CatchIt, sending and receiving the bookmarked moment by e-mails, or to share their bookmarked moment with friends in one of the social networks. The ability to search among the bookmarked moment was the next missed feature for 3 participants. Taggeing people in photos, using a guideline of the application in a separate window, using a tab bar buttons to move between the main functions of CatchIt easily and playing music in the background as long as the user is looking at the details of the target bookmarked moment were missing features (two participants for each missing feature). Using a reminder whether for a random moment (1 participant) or for a specific moment (1 participant), changing the target location on the map by moving the red pin (1 participant), listing the favourite moments (1 participant) and moving between the photos by touching on the photos (1 participant) were missed features as well.

Table 6-14: Frequency of each missed feature in CatchIt.

Missed feature	# Participants
Sharing moment	6
Ability to search	3
Tagging people in photos	2
Using guide window	2
Using tab bar buttons	2
Playing music in the background	2
Random moment reminder	1
A specific moment reminder	1
Changing the target location by replacing the position of the red pin on the map	1
A favourite moments list	1
Moving between photos by touching	1

In the study each of the recommended the ways of display were addressed once. They were include:

- swapping data/time in the bookmarked moment in CatchIt with the moment's label.

- Maximising the size of the instruction about the maximum length of each recording audio.
- Making the noting page bigger.
- Instead of showing the lack of data by using the text, such as *No video* and *No Audio* in the *Moment Details* window, using a visual way might be better.
- Showing the notes of the moment in the map instead of showing the label of the moment.
- Showing a selected photo of the moment to be next to the label and the time of the moment in the *Moment* window.
- Changing the term *label* of the moment to the *name* of the moment.

On the other hand, the message box that instructs the user that the target moment has already been bookmarked, was the only undesired feature in CatchIt. In addition, the additional required information was just for people by adding a photo of each tagged person.

6.3 Discussion

The aim of this section is to discuss the results of the study from appropriate perspectives to find out what are the proper aspects of the mobile applications of the study (CatchIt, Hansel and Contacts) to feed the desktop application (Digital Parrot). To achieve this goal, firstly, the user needs of such system were among our concerns. Secondly, according to understanding the user needs, the aspects of CatchIt will be adopted to feed Digital Parrot in an appropriate way.

6.3.1 Understanding the User Needs

The user needs of capturing application for the target moment were laid on the ease of using the application and on the combination of the ease of remembering with ease of using the system.

6.3.1.1 The Ease of Use

The most important usability principles of user interface of mobile application, which is highlighted in the study, were the visibility (the way of displaying the contents of the user interface), efficiency (the number of steps to execute a task) and limitations that limit the user for using the application as the user requests.

Visibility

Visibility of the contents of the windows was a key factor to judge concerning the easier application to deal with regarding the main function (bookmarking a moment, tag a place or add a new contact). Placing the button of the main function on one window of the application without additional buttons or at least making that button more visible than others decreases the complexity and the ambiguity of interacting with the application's interface (Broadbent & Marti. 1997). As a result, in our study, the majority of participants preferred dealing with regarding the main function of the application Hansel.

Hansel makes the button of tagging the target place more visible than others by placing that button on the main window and the other buttons on the tab bar of the window. Therefore, despite the presence of more than one button of the main window, the object (P5) does mention only the main button. He said, "Simply, there is one

screen in Hansel which has a label and one button”. Beside the main button, Hansel displays one textual field to insert the label of the tagged place. The tagging button and the label field are the most important button and textual field in the application. On the other hand, the main window of CatchIt has three different buttons with same visibility and the main window of Contacts has one button but with lots of additional textual information. Thus, CatchIt and Contacts were more complicated when the participants intended to bookmark the target moment or add a new contact perceptively.

This result supports one of the recommendations of designing CatchIt that is recommended by Allalah (2011). She recommends only one button to bookmark the entire required data of the target moment to make the system easier to use.

On the other hand, participants preferred to label the target moment in CatchIt or to label the target place in Hansel because the label will help them to identify and thus will help them to distinguish the target moment or place respectively. When participants did not labeling the target, they could not distinguished the target. The object (P7) said “Hansel uses less information that makes me find my place clearly. However, if I did not write a label for the place, it would be too difficult for me to know which place I am looking for”. In addition, labeling the target helped the object (P3) to recall some bookmarked moments even though she did not intended to revise these moments. She said “When I read the labels of moments in the moment list, I laughed because I remembered those moments although they were not my target moment”. As a result, CatchIt should place the text field of labeling the moment in the main window.

Efficiency

The number of required steps to execute a certain task of the application was a key to find out how easy it is to interact with the user interface of the application. As long as the application has fewer steps to execute a particular task, the application becomes easier, and vice versa. If the user needs to follow more steps to do what the user wishes, the application becomes more difficult and sometimes leads the user to give up on using the application (Buchanan, Farrant, et al., 2001).

The comparison between the applications CatchIt and Contacts highlighted the relation between the number of steps and the ease of use. Overall, bookmarking a moment was easier than adding a new contact because of several reasons; one of them was as the object (P12) said, “CatchIt is easier because it has fewer steps to bookmark a moment”. In fact, the user of CatchIt needs to press only one button to bookmark the moment.

Similarly, in the same comparison, Contacts was sometimes easier to insert the optional data because of the number of steps the participants had to follow to do so. The object (P9) for instance said, “All of the optional data in Contacts are in one page but in CatchIt, I need to visit a particular window to insert my data”. In fact, in some cases, the user of CatchIt should moves through several windows to do one task such as saving a new person’s name in order to tag that person in the target moment.

Limitations

There were various limitations in using the applications of the study. The most important limitation of capturing further data, the optional ones, of the target moment

in CatchIt was the ways of inserting the optional data. Users of CatchIt were forced to capture the details of the target moment (in case of capturing by photos, video or audio) at the time of the moment. As a result, the user could not refer to the old captured data or the late captured data to the target moment. The object (P7) was in hurry and she wished to add more photos of the target moment later but they could not. She said, “I added a few photos because I had to move on after bookmarking a moment”. Another situation happened with the object (P2). She said “I took photos of one of my interesting moment and saving them in my device. Days later, I wished to bookmark that moment and added these photos to the bookmarked moment but CatchIt did not offer that for me”.

The participants were constrained by other limitations such as connecting to the Internet to tag a place and retrieve the list of the tagged places in Hansel, although the most significant feature of using the smart phones such as iPhone is to connect to the Internet (Allalah & Hinze. 2011). The object (P10) said “I do not want to be limited by the Internet connection to use a mobile application”. As a result, CatchIt was used much more than Hansel.

In fact, this constraint, must connecting to the Internet, prevented us to appropriately recognize if the user prefers to interact with an application that offers various optional data, such as CatchIt, or with application that offers only the most important optional data (the label of the target), such as Hansel. For instance, the object (P4) said “There is no constraint to use CatchIt. I can bookmark the moment any time and from any location then I can change them whenever I want”. Therefore, further investigation is requested to know the user needs in terms of the quantity of the optional data.

Another limitation was the inability of changing the required data of Hansel, the time and location contexts. Some participants changed the required data of CatchIt because they forgot to bookmark the moment from the right place such as what happened with the object (P12). Additionally, when other participants did not change the required data, the participants found doing so very important in some cases. The object (P6) said “I did not need to change my location but I think it is useful if I could change the location especially when it is hard to the GPS to read my current location”.

6.3.1.2 The Ease of Remembering and the Ease of Use (in CatchIt)

CatchIt captures the time and location contexts automatically once the user bookmarks the moment. Also, the user is allowed to manually add the people context and manually capture the contents of the bookmarked moments by taking notes, taking photos, recording videos and recording audio. As a result the participants’ needs of remembering the bookmarked moment and the ease of using CatchIt were for manual adding.

People Context

The names of people present the people context in CatchIt. Overall, the names of people are less likely to be remembered than the faces of people. However, in our study the people names were whether for well known people to the user or for new faces to the user. Well known peoples’ names were familiar enough to be remembered for 7 out of 9 participants, who could distinguish the target person’s contact details by using the person’s name in Contact easier than they distinguished the target moment in CatchIt. That leads to say that the names of well known people to the user is a reasonable query term to retrieve the related information, which supports what

Ahlberg and Shneiderman (1994) believed when they said that people is one of the useful cues for retrieving.

On the other hand, the names of new faces might not help in retrieving due to the difficulty of remembering those names (Allalah, 2011). The object (P11) said “I cannot remember the names of the new people I have met recently”. That might be because remembering names of new people are less likely to be remembered than their faces (Burton & Bruce, 1992). Associating the name of the new people with the image of their face to be as a combined cue might solve this problem as two of the participants suggested: tag people in the token photos (more investigations are required).

Effective Capturing Way: Note

In the initial questionnaire of the study taking notes presented the best way of capturing the important notes of the lecture, capturing the shopping list and capturing the meeting with the supervisor/advisor. Also, taking notes was one of the preferred capturing ways to capture the information about the interesting products and to record moments of the participants’ lives. In all previous situations, taking notes on paper was preferred more than taking the notes using text message (keypad). It is similar to the results of Allalah (2011) study when most of the participants depended on pen and paper to write their notes. In one week of using CatchIt, when the only way of taking notes is by the keypad, 9 out of 12 participants took notes of their bookmarked moments and they found adding the notes either *very easy* or *easy*.

However, although of the importance of taking notes in order for recalling is significant, taking notes wastes the user's time and makes the hand busy with typing, as the object (P7) said, "taking notes as text message will take too long because the keypad is small". Additionally, while walking, taking notes is more difficult (Crossan, Murray-Smith, Brewster & Musizza. 2009) and leads to decrease in accuracy of the notes (Brewster, 2002). Consequently, further investigation is requested to find an alternative input way.

Effective Capturing Way: Photos

Photo is one of the visual cues which helps with information retrieval. All participants in our study preferred to take photos of birthdays, products and moments in their lives to remember the events of those moments or to remember some required information. These findings are similar to the study that Carter and Mankoff (2005) and Allalah (2011) did. Most of the participants in those studies used photos to capture events.

In our study, all participants took photos of their moments by using CatchIt. The ease of doing so was almost ranged between *easy* and *very easy*. In addition, one of the participants (P11) found the ease of taking photos by CatchIt covers his needs for his best way to remember, photos. However, tagging the names of the people in the photos was requested from two of the participants to make recalling the moment easier. This suggestion was made because most of photos that helped to remember are of people, similar to case that happened in Lee and Dey study (2005), when the most of the photos taken by participants were of people.

Furthermore, the textual information was most captured target by photos in Allalah (2011) study and in our study all the participants preferred to take photos of the textual information about the target product. On the other hand, as mentioned, typing the textual information using keypad is relatively difficult and leads to insertions of inaccurate data. Therefore, using optical character recognition (OCR) software to transfer the textual contents of the taken photos might be a better alternative solution for taking notes, by typing the textual information by the keypad. That because inserting the target text by OCR is easier than the manual textual entry in terms of the time consuming (Cushman, Ojha & Daniels, 1990). This suggestion needs further investigation.

Effective Capturing Way: Video

Video is the second visual cue of CatchIt. For the participants the video record was one of the most preferred capturing ways in birthdays and in similar significant moments in the participants' lives. The object (P8) commented, "I love recording video but I prefer taking photos because videos have lots of information that I do not need, not same as photos where I can take one photo with enough of information".

The use of recording video in our study, using CatchIt, was between *very easy*, *easy* and *neither easy nor difficult*. This result might lead to the need for changing the user interface of CatchIt to meet more of the user needs in terms of ease of use.

Effective Capturing Way: Audio

For the participants recording audio was less popular cue to remember and less capturing way used in CatchIt. Although recording audio was the second popular capturing way in two scenarios, recording important notes from a lecture and

recording a meeting with supervisor/advisor, recording audio was not used at all to record events in birthdays, the shopping list, information about a product and moments of the participants' lives.

The results of Allalah (2011) study show that only 8 out of 30 participants depended on recording the information that they wished to remember in audio format. Likewise, in our study just 5 out of 12 participants recorded some details of their moment by audio recording. In addition, 7 out of 12 participants suggested leaving out this capturing way from CatchIt. The object (P7) explained, "I do not usually use recording audio", and the object (P12) also said, "It does not make any sense to me". Furthermore, listening to the audio recording was more difficult than reading the taken notes in order to extract the target information for the object (P7). Object (P7) said, "It takes less time to find an item with text than checking images or listening to audio record".

On the other hand, according to Lai, Mitchell, Wood & Lee (2002) speech recognition technology (speech-to-text) is more useful and is easier input form than the textual input form especially with busy hands and eyes. Supportively, the five participants who did use the audio recording found using the audio recording in CatchIt between very easy and easy. Moreover, as we mentioned the difficulty and inaccuracy of taking notes and also the ease of finding the target information from the text form of data, a combination alternative capturing way is requested (audio to insert and text to output). Speech-to-text might be one of the alternative input ways of the textual input (further investigation is requested).

6.3.2 A Leap Forward in Best Feeding

CatchIt is a mobile application, which bookmarking the selective moment (the mobile context and contents) is the main goal of the application. The selective bookmarked moment then will be transferred to the desktop application (Digital Parrot) which depends on the textual format of data to store, recognize and retrieve the stored data. Nevertheless, the aim of this thesis is to find the easiest ways of interactions with the user interface of CatchIt to feed Digital Parrot without concentrating on how the actual feeding will be done. Additionally, light will be shed on the best way of recalling the stored information.

Accordingly, CatchIt should put dealing with the textual output as well as it should cover the user needs discovered in the study, as explained in Section 6.4.1. Moreover, as CatchIt is a mobile application, the number of features of the application should be considered. According to Nielsen (2011) number of features of the mobile application should be reduced to match exactly the user need because of the small size of the screen.

6.3.2.1 Cover the User Needs

The participants determined three of the usability principles: visibility, efficiency and limitation. These principles will be in our account when we aim to improve the usability of using CatchIt in terms of the ease of use to feed Digital Parrot.

Visibility

The visibility of the main function of Hansel was the best way of displaying the main window of the application. The biggest button of the main window of Hansel is the

tagging button beside the most important optional data, the label of the tagged place. Hence, CatchIt should follow the same displaying way of its functions. The main window might include one big sized button for bookmarking a moment (the main function of CatchIt) as well as the text field for inserting the label of the bookmarked moment.

Efficiency

CatchIt feedbacks the user when the user bookmarks the target moment by popping up a message box and also by moving the user to another window (*Moment Details*). Instead of that, CatchIt should have one of those interaction ways to let the user know about their actions. As a result the number of steps will be decreased by one step. So, the user either (1) presses *OK* button of the popped up message box or (2) presses *Back* button of the *Moment Details* window to return to the main window. The second interaction way might be more appropriate and lead to reducing the number of steps as well in case the user wishes to insert further details once the user bookmarks the target moment.

Another important way to reduce the number of steps the user should follow to execute a particular task is using the tab bar buttons. Hansel has tab bar buttons that allows the user to reach a particular function with few numbers of steps. For example, the user can drop tag, see the tagged places on a map and see the tagged places on the list from any window of the application. In contrast, the user of CatchIt should move between (1) the main window, (2) *Moment Details* window, (3) *Tag People* window, (4) *Select People* window, and then (5) *Person* window to add a the name of the requested person. The user then should comeback window by window to *Moment*

Details as such to do another task. Consequently, CatchIt should have tab bar buttons, which are located on all windows in order to facilitate executing the user task.

Constraints

Participants were limited by the Internet connection in Hansel to tag a place and to retrieve the tagged places whether on the list or on the map. In contrast, CatchIt does not depend on the Internet connection to bookmark the target moment and to retrieve the stored data. As a result, CatchIt should maintain non-reliance on the Internet connection to do its functions.

Also, the user of Hansel cannot change or delete both the required data and the optional ones; that limited the user to insert their data without any mistake. If the user commits an error while using Hansel, they should re-tag the place with no mistake. On the other hand, CatchIt allows the user to do any change they request or delete the undesired data as well. So, CatchIt should keep these options.

Nevertheless, CatchIt constrains the user to not using other resources to capture the optional data such as the stored data in the portable device. For instance, Contacts allows the user to take a photo of the target person whether by the built-in camera or by choosing one of the stored photos in the device. As a consequence, CatchIt should extend the choices of capturing and should not limit the user to using just the capturing ways of the application.

6.3.2.2 Number of Features

The number of features that CatchIt has should be reduced to meet the user needs in terms of ease of use, on the first hand and to feed Digital Parrot appropriately, on the other. The user of CatchIt is able to add the people context and to add some details of the bookmarked moment by taking notes (textual input and textual output), taking photos, recording video and recording audio.

However, despite the importance of recording the names of people who were with the user within the target moment, adding those names might be by alternative and easier way such as recording the names in an audio format. This alternative solution should be investigated.

Moreover, although the importance of taking notes of the bookmarked moment and use those notes as cues to remember, inserting the textual data using the keypad is difficult and leads to inserting inaccurate data while movement of users. So, CatchIt should disband with taking notes. Recording video was a popular way to remember but it took long time to capture the target and it has rich information that the user might not really wish to retrieve. Thus, we suggest, recording video should be taken off CatchIt.

On the other hand, photos and recording audio might be helpful when they continue to be used in CatchIt because of the ability to using some technologies that assist to convert the text contents of the taken photos to text (OCR) or to convert speech to text (speech recognition) as well. Therefore, the user can insert the target textual notes easily and can get the benefit of taking notes to recall the past experience. Further

investigations are required regarding the usability and the usefulness of such alternative capturing ways in CatchIt.

6.4 Summary

Evaluating of the usability of CatchIt was the central concern of this chapter. The evaluation was obtained by a user study. Three goals were addressed before embarking on the study to be achieved at the end of the study: (1) explore more of the user needs in terms of ease of using the mobile application and then (2) improve the user interface of CatchIt to (3) feed Digital Parrot in an appropriate way. To do that, 12 participants were involved into the study and also two applications were involved (Hansel and Contacts) in addition to CatchIt.

Initial questionnaire was handed to the participants to learn more about the preferred capturing ways to capture the target information of different situations in order to recall it in the future. Then, the participants tested CatchIt and Hansel in two weeks (one application per week). The last step of the study involved interviewing the participants in order to understand more of their needs in terms of the ease of using the mobile applications.

Participants were asked to answer the initial questionnaire in order to learn more about the preferred capturing ways to capture the target information of different situations in order to recall it in the future. They were then asked to test CatchIt and Hansel in two weeks (one application per week). Finally, the participants were interviewed. The questions of the initial questionnaire and the guided interview were

reformulated and the estimated time of the guided interview was determined as the results of the pilot study.

The interview had five categories of questions: (1) the ease of using CatchIt, (2) the ease of using Hansel, (3) comparison between the ease of use CatchIt and Hansel, (4) comparison between the ease of using CatchIt and Contacts and (5) the feeling of using CatchIt.

The results of the study were analyzed and divided to four groups. The first group was concerned with the demographic information of the participants. The second group was for the results of the preferred ways capturing the surrounded information. The third group was for the results of the usability of CatchIt, and the last group was for the results of the feelings of using CatchIt and suggestions for improving CatchIt.

After analysis, the results were discussed from the perspective of the goal of the thesis, developing an easier way to bookmark the target moment and then to find an appropriate way to feed the desktop application, Digital Parrot. Consequently, the user needs in terms of easing interacting with the user interface of the mobile applications were discussed. Accordingly, some aspects of CatchIt were kept, others left out or adapted to make CatchIt feeds Digital Parrot appropriately.

The next chapter will recommend new methods that could assist in adjusting the user interface and the aspects of CatchIt in order to conclude the research with some suggestions to feed Digital Parrot in an appropriate way.

7. Conclusion and Further Work

This concluding chapter summarizes this research and also outlines the steps that were undertaken to reach the objectives of the research. In the final section of this summary chapter suggestions for further research work are offer.

7.1 Summary

The research aid to design and implement a semi-automatic feeding system to a human memory aid system (Digital Parrot) in a way that would be most appropriate for the users. Because Digital Parrot is a desktop application, it requires an appropriate mobile application to capture the mobile-contexts of the user. Consequently, *CatchIt* was designed, implemented and evaluated.

CatchIt is a mobile application that bookmarks selected moments semi-automatically in order to feed Digital Parrot. It captures the time and location contexts automatically; however the people context and the further details of the moment are captured manually.

In order to develop *CatchIt*, we surveyed in Chapter 1 the two approaches (PIM and CARPE) which support the management of the personal information and the recalling of this information. We provided a brief overview of Digital Parrot which is an augmented memory system that is characterized by combining the approaches PIM and CARPE for better recalling. We discussed the Digital Parrot's needs for a capturing application that feeds Digital Parrot with appropriate cues. Developing this feeder application was then defined as the aim of this thesis.

In Chapter 2 we introduced a scenario to illustrate the augmented memory area. We explained the types of human memory and how particular information, *cues*, flow among various types of human memory. As our research focuses on capturing the cues, based on the scenario, we explained the following: types of cues, the methods of capturing the cues, and ways of capturing the cues.

In Chapter 3 we compared our related works. In particular, *Hansel* and *Contacts* are mobile applications that capture some cues for different purposes. Hansel is used to tag the target places to revisit these places, whereas Contacts is used to add contacts into phone. On the other hand, the *bookmarking a moment* approach aims to bookmark selective moments of the user. This approach also introduced the requirements of a mobile capturing application. After explaining the related works, Chapter 3 concluded with specifying the requirements of our mobile capturing application, *CatchIt*, based on both results- from the *bookmarking a moment* approach and from our scenario.

In Chapter 4 the conceptual architecture of CatchIt was presented. This included the main three steps of CatchIt: bookmarking a moment, revising, and synchronizing with a desktop application, such as Digital Parrot. Then we presented the design of a user interface prototype of CatchIt. For better results and to improve its overall design, an expert walk-through method was applied on this prototype. As a result of this method the user interface prototype was finalized with the improvements.

In Chapter 5 the suitable portable device (iPhone) and the target user group were determined. Then the database, the overall architecture and the user interface of

CatchIt were implemented. The database stores the captured data in the portable device from the user interface and retrieves the stored data to the user interface. The final prototype is fully functional in capturing cues from its users in a semi-automatic way: text and pictures are entered manually and automatically assigned to location and time. We also introduced an initial idea to transfer the data from CatchIt to Digital Parrot: An XML file which contains the captured cues of the bookmarked moments is sent by email from CatchIt to Digital Parrot.

In Chapter 6 we evaluated the usability of CatchIt. In our study we compared CatchIt with two existing mobile applications *Hansel* and *Contacts*. The comparisons between CatchIt and Hansel and between CatchIt and Contacts explored more of the user needs for the mobile application that captures their moments. We found that some users wished to make the user interface of CatchIt easier to use and to make the functions of CatchIt more effective and therefore to feed Digital Parrot in the most appropriate way. As a result, we made suggestions for the improvement of the CatchIt design so as to make it more appropriate to serve as a feeder service for Digital Parrot. Overall, the results of the user study were encouraging, as expressed by participants.

7.2 Further Work

This thesis raises several issues to be researched in more detail. Improving the user interface and building the entire aid system for the human memory are the main research work that requires further investigation. Additionally, we discuss the future use of CatchIt and Digital Parrot for people with memory impairments (such as caused by Traumatic Brain Injury).

Improvement of the User Interface of CatchIt

We develop the first version of the user interface of CatchIt which bookmarks the selective moments. Further versions should be implemented to cover more of the user needs that this thesis discovered. The new user interface should be easier to use and more effective to remembering. The new user interface should cover the requirements listed below:

- The user interface should display the main function and the main optional data of the application (label) distinctively in one window.
- The user of the new users interface should be able to do their tasks with fewer numbers of steps.
- The users should be able to use the application and change the required and the optional data whenever and wherever they wish.
- The capturing ways of CatchIt should be reduced to include only the easiest capturing ways (taking photos and recording audio). As a result, these capturing ways should have further features, such as transcription of the texts in the captured photos to textual notes and also transcription the speech to textual notes.

Moreover a further study should be applied to investigate the quantity of the optional data that the user prefers to deal with in the user interface of the moment capturing application.

Pieces to Complete the Augmented Memory System (Digital Parrot)

A synchronization framework should be located between Digital Parrot and CatchIt for synchronizing data. In this framework the user can review the list of the bookmarked moments on a big screen and then revise them. Also, the user can transfer the required moments to Digital Parrot to store those moments permanently and to take the benefits from using such powerful aid human memory to organize and retrieve the user experience. This framework could be a part of Digital Parrot so that the user will not have to use other software, which might require more effort and time to become familiar with.

Another benefit of introducing this synchronization framework is the increased possibility of enhancing the recollection of stored experience in the human memory. This possibility is enhanced because the user participates in a process being **revising bookmarked moments** with *target moments*.

As a result, the entire system will be restructured to be as follows:

1. CatchIt: bookmarks the selective moments and captures the surrounded information of the target moments in the most appropriate way and then transfers the bookmarked moments to the mid framework.
2. Synchronization framework: receives the bookmarked moment and allows the user to revise the bookmarked moments in a big screen. This process will insert the user in a long loop with the past experience, which might lead to better recalling. At the end of revising, the revised moments will transfer to Digital Parrot.

3. Digital Parrot: stores the revised moments permanently, organizes them semantically, and then retrieves them according to the user search term.

CatchIt and Traumatic Brain Injury (TBI)

Forgetfulness is one of the major difficulties that people with TBI suffer from (Thompson, 1996). Consequently, these people are likely to use additional tools to help them recall their everyday events (Hart, Hawkey & Whyte, J, 2002). As mentioned in Section 5.2, one of the target users of CatchIt is people who need some help to remember their past experience. However, our current implementation and user study did not focus specifically on participants with TBI.

As CatchIt is a feeder mobile application to Digital Parrot, further research is necessary to explore how people with TBI behave and interact with CatchIt as a capturing application. In addition, further studies are required to investigate if people with TBI would receive benefits of efficient recalling by using CatchIt and Digital Parrot together. Research regarding aspects of using CatchIt and Digital Parrot for people with memory impairments is planned for our next research project.

References

- Ahlberg, C., & Shneiderman, B. (1994). Visual information seeking: Tight coupling of dynamic query filters with starfield displays. *Proceedings of the SIGCHI conference on Human factors in computing systems: Celebrating interdependence*, (pp. 313-317). Boston, Massachusetts: ACM. doi:10.1145/191666.191775.
- Alallah, J. (2011). *Capturing situational context in an augmented memory system*. (Master's thesis). University of Waikato, Hamilton, New Zealand. Retrieved from <http://hdl.handle.net/10289/5915>.
- Alallah, J., & Hinze, A. (2011). Feeding the digital parrot: Capturing situational context in an augmented memory system. *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, (pp. 1-10). Canberra, Australia: ACM. doi:10.1145/2071536.2071537.
- Apple. (2011a). *Core location framework reference*. Retrieved from http://developer.apple.com/library/ios/#documentation/CoreLocation/Reference/CoreLocation_Framework/_index.html#//apple_ref/doc/uid/TP40007123.
- Apple. (2011b). *Map kit framework reference*. Retrieved from http://developer.apple.com/library/ios/#documentation/MapKit/Reference/MapKit_Framework_Reference/_index.html.
- Apple. (2012a). *Contacts*. Retrieved from <http://www.apple.com/iphone/built-in-apps/contacts.html>.
- Apple. (2012b). *Developer tools*. Retrieved from <https://developer.apple.com/technologies/tools/>.
- Apple. (2012c). *iPhone*. Retrieved from <http://www.apple.com/iphone/>.
- Atkinson, R., & Shiffrin, R. (1971). The control of short-term memory. *Scientific American*, 225(2), 82-90.
- Brewster, S. A. (2002). Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing* 6(3), 188–205.
- Broadbent, J., Marti, P. (1997). Location aware mobile interactive guides: Usability issues. *Proceedings of the 4th International Conference on Hypermedia and Interactivity in Museums*, 88-98. Doi. 10.1.1.40.3065
- Buchanan, G., Farrant, S., Jones, M., Thimbleby, H., Marsden, G., & Pazzani, M. (2001). Improving mobile internet usability. *Proceedings of the 10th international conference on World Wide Web*, (pp. 673-680). Hong Kong, Hong Kong: ACM. doi:10.1145/371920.372181.

- Burt, C., Mitchell, D., Raggatt, P., Jones, C., & Cowan, T. (1995). A snapshot of autobiographical memory retrieval characteristics. *Applied Cognitive Psychology, 9*, 61–74.
- Burton, A., & Bruce, V. (1992). I recognize your face but I can't remember your name: A simple explanation? *British Journal of Psychology, 83*(1), 45.
- Carter, S., & Mankoff, J. (2005). When participants do the capturing: The role of media in diary studies. *Proceedings of the SIGCHI conference on Human factors in computing systems*, (pp. 899-908). Portland, Oregon, USA: ACM. doi:10.1145/1054972.1055098.
- Crossan, A., Murray-Smith, R., Brewster, S., & Musizza, B. (2008). Instrumented usability analysis for mobile devices. In J. Lumsden, (Ed.), *Handbook of Research on User Interface Design and Evaluation for Mobile Technology* (pp. 927-944). Hershey, PA: Information Science Reference.
- Cushman, W., Ojha, P., & Daniels, C. (1990). Usable OCR: What are the minimum performance requirements? *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, (pp. 145-152). Seattle, Washington, United States, ACM. doi:10.1145/97243.97267.
- DeWitt, S. (2007). The effects of note taking and mental rehearsal on memory. *Journal of Undergraduate Psychological Research, 2*, 46-49.
- Domnitcheva, S. (2001). Location modeling: State of the art and challenges. *In Workshop on Location Modeling for Ubiquitous Computing*. Atlanta, GA, USA. doi:10.1.1.65.7125.
- Elmasri, R., & Navathe, S. (2003). *Fundamentals of database systems* (4th ed.). Boston, MA: Addison-Wesley.
- Ericsson, K., & Kintsch, W. (1995). Long-term working memory. *Psychological Review, 102*, 211–245.
- Gemmell, J., Bell, G., & Lueder, R. (2006). MyLifeBits: a personal database for everything. *Communications of the ACM, 49*(1), 88-95. Doi:10.1145/1107458.1107460.
- Markus Luczak-Rösch. (2012). *Hansel*. Retrieved from http://www.markus-luczak.de/ladmich/hag_app.zip
- Hart, T., Hawkey, K., & Whyte, J. (2002). Use of a portable voice organizer to remember therapy goals in traumatic brain injury rehabilitation: A within-subjects trial. *The Journal of Head Trauma Rehabilitation, 17*(6), 556-570.
- Jaimes, A., Omura, K., Nagamine, T., & Hirata, K. (2004). Memory cues for meeting video retrieval. *In: CARPE'04: Proceedings of the 1st ACM workshop on Continuous archival and retrieval of personal experiences*, (pp. 74-85). New York, NY: ACM. doi:10.1145/1026653.1026665

- Lalanne, D., & Hoven, E. (2007). Supporting human memory with interactive systems. *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it, 2*, 215-216. Lancaster, UK: British Computer Society.
- Lai, J., Mitchell, S., Viveros, M., Wood, D., & Lee, K. (2002). Ubiquitous access to unified messaging: A study of usability, and the limits of pervasive computing. *International Journal of Human Computer Interaction, 14* (3-4), 385-404.
- Lamming, M., & Flynn, M. (1994). Forget-me-not: Intimate computing in support of human memory. *Proceedings of FRIEND21_94, the International Symposium on Next Generation Human Interface*, Magufo Jagoen, Japan, 2-4.
- Lansdale, M. (1988). The psychology of personal information management. *Applied Ergonomics, 19*(1), 55-66.
- Lee, M. L. & Dey, A. K. (2007). Providing good memory cues for people with episodic memory impairment. *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, (pp. 131-138). Tempe, Arizona, USA: ACM. doi:10.1145/1296843.1296867.
- Nielsen, J. (2011). *Mobile usability update*. Retrieved from <http://www.useit.com/alertbox/mobile-usability.html>.
- Polyn, S., Natu, V., Cohen, J., & Norman, K. (2005). Category-specific cortical activity precedes retrieval during memory search. *Science, 310*(5756), 1963-1966. doi:10.1126/science.1117645.
- Schweer, A. (2011). *Augmenting autobiographical memory: An approach based on cognitive psychology*. (PhD thesis). University of Waikato, Hamilton, New Zealand.
- Sellen, A. J. & Whittaker, S. (2010). Beyond total capture: A constructive critique of lifelogging. *Communications of the ACM, 53*(5), 70-77. doi:10.1145/1735223.1735243
- Shiffrin, R. (1993). Short-term memory: A brief commentary. *Memory & Cognition 21*(2), 193-197.
- Smith, S., Glenberg, A., & Bjork, R. (1978). Environmental context and human memory. *Memory & Cognition, 6*(4), 342-353. doi:10.3758/bf03202732.
- Sommerville, I. (2001). *Software engineering* (6th ed.). Harlow, England: Addison-Wesley.
- Tan, D., Stefanucci, J., Proffitt, D., & Pausch, R. (2001). The Infocockpit: Providing location and place to aid human memory. *Proceedings of the 2001 workshop on Perceptive user interfaces*, (pp. 1-4). Orlando, Florida: ACM. doi:10.1145/971478.971526.

Thompson, S. B. (1996). Practical ways of improving memory storage and retrieval problems in patients with head injuries. *British Journal of Occupational Therapy*, 59, 418–422.

Westfall, L. (2006). *Software requirements engineering: What, why, who, when, and how*. Texas: The Westfall Team. Retrieved from http://www.westfallteam.com/Papers/The_Why_What_Who_When_and_How_Of_Software_Requirements.pdf.

Zhang, J. (2004). Memory process and the function of sleep. *Journal of Theoretics*, 6(6), 1-7.

Appendix A (Materials for the Usability Study)

This appendix includes the materials that were used through the user study.

Ethical Approval

Computing and Mathematical Sciences
Erewhito me ūā Pātaka Māngoroa
The University of Waikato
Private Bag 3105
Dunedin
New Zealand

Phone +64 7 338 4021
www.scms.waikato.ac.nz



THE UNIVERSITY OF
WAIKATO
Te Ūāka o Waikato

21 February 2012

Hasan Alyamani
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Hasan

Request for approval to conduct an experiment study involving human participants for your Masters Degree

I have considered your request to conduct an experiment for your research project involving human participants, *CatchIt: Bookmarking a Moment to Feed the Digital Parrot*, commencing 1st March this year. The purpose of the research is to test us usability and functionality of 'CatchIt' in order to compare your system with another.

The procedures described in your request are acceptable.

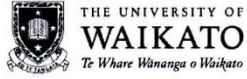
I note your statement that no names of participants will be disclosed in any resulting publications and information will be stored in the FCMS data archive for three years and then destroyed.

The research participants' information sheet, consent form, and questionnaire meet the requirements of the University's human research ethics policies and procedures.

Yours sincerely,

Mike Mayo
Human Research Ethics Committee
Faculty of Computing and Mathematical Sciences

Initial Questionnaire



Participant ID: _____

CatchIt: Capturing Cues of Bookmarked Moment to Feed Digital Parrot

We would appreciate it if you would complete this questionnaire for us to learn more about your preferred ways to recall information.

1. Age group

- Under 20 20-30 31-40 Over 40

2. Gender

- Male Female

3. Which way would you prefer to capture the events in your friends' birthday party? (tick all applicable)

- Taking notes (on paper) Taking notes (text message) Taking photos
 Recording video Recording audio Other ways:
-
-

Any comments

4. Which way would you prefer to capture the important notes in a lecture? (tick all applicable)

- Taking notes (on paper) Taking notes (text message) Taking photos
 Recording video Recording audio Other ways:

Any comments

5. Which way would you prefer to capture what you should buy from the supermarket? (tick all applicable)

- Taking notes (on paper) Taking notes (text message) Taking photos
 Recording video Recording audio Other ways:

Any comments

6. Which way would you prefer to capture the details of the meeting with your supervisor/ advisor? (tick all applicable)

- Taking notes (on paper) Taking notes (text message) Taking photos
 Recording video Recording audio Other ways:

Any comments

7. Which way would you prefer to capture the interesting information about a product? (tick all applicable)

- Taking notes (on paper) Taking notes (text message) Taking photos
 Recording video Recording audio Other ways:

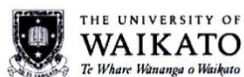
Any comments

8. Tick the best two ways that you use to record moments in your life?

- Take notes (on paper)
- Take notes (text message)
- Take photos
- Recording videos
- Recording audios
- Other ways:

Any comments

Questions of the Guided Interview



Participant ID: _____

CatchIt: Capturing Cues of Bookmarked Moment to Feed Digital Parrot

Guided interview was carried out when the participants completed using both applications, CatchIt and Hansel. Each participant was interviewed individually. In the interview, participants were asked five categories of questions.

The first category of questions concerned the participants' experiences of using our system, CatchIt. The second category of questions concerned the participants' experience of using the system Hansel. The third category of questions aimed to compare the two applications, CatchIt and Hansel, according to the participants' experience in using them. The fourth category of questions designed to compare between CatchIt and Contacts. The fifth category of questions aimed to extract the participants' feelings and interests using in applications.

The aim of all these groups of questions is to find out which is the best application to feed the desktop application, the Digital Parrot and to find out the way to make our system easier to use.

9(b). How easy is it to change the location of that bookmarked moment?

- Very easy Easy Neither easy nor difficult Difficult Very difficult

Any comments:

10. How easy is it to go back to the previous window?

- Very easy Easy Neither easy nor difficult Difficult Very difficult

This is the end of the first category of questions

The Third Category of questions (CatchIt Vs Hansel)

The following questions are to compare between the applications *CatchIt* and *Hansel*.

		Systems			Reasons
		CatchIt	Hansel	Same	
Criteria	Which application is easier to deal with regarding bookmarking a moment in CatchIt, or tagging a place in Hansel?				
	Which application is easier to distinguish the target: the bookmarked moment in CatchIt or the tagged place in Hansel?				
	Which application is easier to insert the required data of the application (time and location for both application CatchIt and Hansel) to the bookmarked moment in CatchIt or to the tagged place in Hansel?				
	How many moments did you bookmark?				
	How many places did you tag?				

This is the end of the third category of questions

The Fourth Category of questions (CatchIt Vs Contacts)

The following questions are to compare between the applications *CatchIt* and *Hansel*.

		Systems			Reasons
		CatchIt	Contacts	Same	
Criteria	Which application is easier to deal with regarding bookmarking a moment in CatchIt or adding a new contact to the contact list in Contacts?				
	Which application is easier to distinguish the target: the bookmarked moment in CatchIt or the contact in Contacts?				
	Which application is easier to insert the required data of the application: time and location in CatchIt or the contact's first name, last name, phone number, etc (one of them is enough to save the new contact)?				
	Which application is easier to add the optional data, photo, to the bookmarked moment in CatchIt or to the contact in Contacts?				

Which application is easier to add the optional data, textual data, to the bookmarked moment in CatchIt or to the contact in Contacts?				
Which application is easier to add the optional data, people, to the bookmarked moment in CatchIt or to the contact in Contacts?				

This is the end of the fourth category of questions

The Fifth Category of questions (CatchIt Feelings)

The following questions are to know more about your feelings of using *CatchIt* and to get your suggestions to improve *CatchIt*.

1. What languages did you use while you bookmarked the moment?

2. Did you find supporting of different languages in *CatchIt* useful?

Very useful Somewhat useful Undecided Not very useful Not at all useful

3. What capturing ways do you think that “*CatchIt*” should have?

4. Which capturing ways in *CatchIt* did you like and would use in future?

5. Which capturing ways in *CatchIt* would you not use and think can be left out?

6. Can you think of another application that is similar to *CatchIt*?

7. Are there any features in that application that you missed here?

Appendix B Participants' Expressions

This appendix contains the participants' expressions of both the initial questionnaire and the guided interview.

Participants Expressions of the Initial Questionnaire

The initial questionnaire was used in the study to evaluate of the usability of the user interface of CatchIt. The goal of the questionnaire was to know more about preferred capturing ways in different scenarios (situations). The following table shows expressions of the participants in this questionnaire. The expressions are ordered firstly by the situation then by the participants' numbers.

Situation	P#	Expression
Friend's Birthday Party	P7	Capturing events is better with images.
Setting in a Lecture	P7	Taking notes as text message will take too long because keypad is small.
	P9	I type messages of the lecture if the notes are very important and too short at the same time.
Shopping from Supermarket	P7	It takes less time to find an item with text than checking images or listening to the audio record.
	P8	I take photos from the cooking book to buy the contents.
Meeting with a Supervisor/Advisor	P7	Recording audio is a useful way to remember details.
Interested in a Product	P7	It is easier to capture and review the product details with photos.

	P8	I like taking photos by phone because I always carry my cell phone with me.
Recording Moments in Life	P7	I like to add notes but if it is describing a picture or video, such as in the album of 'Facebook'.
	P8	I love taking photos because it has lots of information.
	P8	I love recording video but I prefer taking photos because in video there are lots of information that I do not need, not same as photos where I can take one photo with lots of information.
	P8	Sometimes I do not recognize my hand writing when I write my notes very quickly.
	P8	Sometimes, I take more time to find a paper and pen, so I do not prefer taking notes.
	P8	Using my cell phone is the best way to save my important information.
	P8	Taking photos and capturing videos mean less required time.

The Participants Expressions of the Guided Interview

The guided interview accrued after two weeks of testing CatchIt and Hansel (one week for each). This interview consisted of five categories of questions. The aim of these questions to understand more of the user needs for the user interface of the mobile application that captures the users' moments. The participants' expressions are show in the following five tables. The expressions in the first four categories will be grouped by the task then by the participants' numbers whereas the expressions in the last table are grouped by the aspects of CatchIt and then by the participants' numbers.

The Ease of Using CatchIt

Task	P#	Expression
The ease of adding photos in CatchIt	P2	I took photos of one of my interesting moment and saving them in my device. Days later, I wished to bookmark that moment and added these photos to the bookmarked moment but CatchIt did not offer that for me.
	P5	The process is quite long.
	P7	I like the camera icon. It tells me the purpose of it.
		I added a few photos because I had to move on after bookmarking a moment.
Added video	P4	I like using photos rather than videos.
Added audio	P5	I use audio just in lecture and I had not any lecture to record.

	P7	I do not usually use recording audio.
	P8	I did not need to add audio although I like recording audio.
	P12	It does not make any sense to me.
The ease of tagging people in CatchIt	P7	I like it because I add the name once then I can use it again and again.
	P8	I like the idea of the two lists of people. One for the current moment and one for people in all moments.
Ease of changing the time	P5	I bookmarked the moment after it happened. So, I changed the time.
Changing the location	P5	I could not recognize it although I needed to change the location.
	P6	I did not need to change my location but I think it is useful if I could change the location especially when it is hard to the GPS to read my current location.
Ease of changing the location	P3	I should know the address of the target location.
	P12	It was easy to change the location because I used the address format.
	P12	I changed the location because I forgot to bookmark the moment from the right place.
Overall	P1	I love CatchIt because it takes the accurate time for me. I always depend on the time when I write my notes. So, I am not worried about the

		time anymore with CatchIt.
	P3	When I read the labels of moments in the moment list, I laughed because I remembered those moments although they were not my target moment.
	P6	CatchIt is handy, flexible and useful.
	P11	CatchIt covers my best way to remember, photos.
	P11	I found CatchIt very nice and useful.

The Ease of Using Hansel

Task	P#	Expression
Labeling the tagged place	P12	There is no information says that I should put a label.
The ease of labeling the tagged place	P6	I was not sure if the label was stored.
Going back to the previous window	P5	There is no back button in the all windows.
	P7	I have to really know about the previous window to go.
Overall	P9	I did not recognize that the tagged place is on the map- there is no sign.
	P9	Not all icons are important - setting button is useless.

CatchIt Vs Hansel

Task	P#	Expression
Easier to deal with regarding bookmarking a moment in CatchIt or tagging a place in Hansel.	P4	There is no constraint to use CatchIt. I can bookmark the moment any time and from any location then I can change them whenever I want.
	P5	Simply, there is one screen in Hansel which has a label and one button.
	P6	Hansel is a bit confusing. There is no label on it. I tried to tag a place many times and I did not realize that the place was already tagged.
	P9	I thought that I should insert a label in Hansel to tag the place.
	P12	The name of the button is clearer in CatchIt than Hansel.
Easier to distinguish the target	P1	CatchIt has two kinds of information to distinguish the moment. If I forget the label there is the time in default while if I do that in Hansel there is nothing to remind me.
	P3	In CatchIt, when I did not put a label to the moment, there is the time for the moment.
	P7	Hansel uses less information that makes me find my place clearly. However, if I did not write a label for the place, it would be too

		difficult for me to know which place I am looking for.
	P12	I prefer remembering by label so Hansel is easier to distinguish the required place. However, if CatchIt replace its way of showing the list of bookmarked moments to be by label then by time, then CatchIt would be easier than Hansel.
Easier to add the required data	P9	I like CatchIt; I can delete and change the data, which I could not do with Hansel.
	P10	I do not want to be limited by the Internet connection to use a mobile application.
	P12	CatchIt gives me more options to deal with inserting the required data.
Overall compressions	P4	CatchIt shows the target location once I want to look at the map whereas I should find the target location on Hansel myself.
	P5	CatchIt is much more accurate in detecting the target location
	P9	I can depend on different ways of capturing in CatchIt to remember the moment, which I could not do with Hansel.

CatchIt Vs Contacts

Task	P#	Expression
Easier deal with regarding bookmarking a moment in CatchIt or adding a new contact in Contacts.	P4	I can bookmark any moment while I am driving where I could not add a contact.
	P12	CatchIt is easier because it has fewer steps to bookmark a moment.
Easier to distinguish the target	P7	I have to remember one out of three pieces of information in order to distinguish the target moment- time, label, or time and label.
	P11	I cannot remember the names of the new people I have met recently.
	P12	Find a particular contact from the contact list is easier because I am more familiar with my contact list.
	P12	There is no search option in CatchIt. So, it is hard to find a particular moment.
Easier to add the required data	P12	To add a new contact I need to insert more information whereas I did not need to insert any information to bookmark a moment.
Easier to add the optional data- photo	P9	All of the optional data in Contacts are in one page but in CatchIt, I need to visit a particular window to insert my data.
Easier to add the optional data- notes/ textual	P4	I do not like to be limited by where I add my data so that I like CatchIt.

information	P11	I could write emails in the notes of CatchIt with less time.
	P12	In CatchIt, I can add any text in one place.

CatchIt Feelings

Aspect	P#	Expression
Supporting multi-languages	P5	It would be easier to remember moment if I used my first language.
Capturing ways to leave out	P1	It is useless to capture people by using "People" while I can add the photo or video of them.
	P11	When I remember events from time, location, some photos and the name of the event, I could remember lots of information about the event.