



Project Report

Master in Computer Engineering – Mobile Computing

Care4Value – a mobile app as a clinical instruments' feedback collector

Nivedhitha Gowdhaman Malarvizhi

Leiria, March 2019

This page was intentionally left blank

Project Report

Master in Computer Engineering – Mobile Computing

Care4Value – a mobile app as a clinical instruments' feedback collector

Cofinanciado por:



Nivedhitha Gowdhaman Malarvizhi

Project Report developed under the supervision of Professor *Catarina Isabel Ferreira Viveiros Tavares Reis* and Professor *Marisa da Silva Maximiano*, professor at the School of Technology and Management of the Polytechnic Institute of Leiria

This project is on behalf of the Care4Value research group. This project is co-funded by the European Regional Development Fund, under the Portugal 2020 Program, through COMPETE 2020 (POCI-01-0145-FEDER-23248)

This page was intentionally left blank

Acknowledgments

I would like to thank my supervisors Professor Catarina Isabel Ferreira Viveiros Tavares Reis and Professor Marisa da Silva Maximiano for their insight, mentorship, patience, accepting this work, supporting during the course of this development and my graduate studies at Institute Polytechnic of Leiria (IPL).

I also want to express my sincere gratitude to Professor Carlos Grilo for allowing me to work in this project and providing all the necessary facilities to pursue my work and effort to support this project. Lastly, I want to thank my friends and family who have given me constant moral support and sound advice during my graduate academic career at IPL. Your support and belief have been immeasurable, Thank you.

This page was intentionally left blank

Abstract

Mobile healthcare applications are devotedly accepted by the healthcare organizations and by patients. The reasons behind acceptance of mobile healthcare applications are the fact that they are user friendly, reliable, low cost, time efficient, mobility, etc.

In this report, it is presented an integrated architecture for secure mobile healthcare assessment. This mobile application was developed for the android platform. We have made an application that will care about the life of the patient, by making assessment of his/her pain. Also, it gives a review of the type of care the patient needs in each stage.

The application has the ability to predict the need of the patient through reviewing the pain scale that the patient fills from the distinct profiles. This helps the healthcare professional to have an updated vision of the status of the patient and also enables the chance to review the status of each patient along authentication.

Keywords: Mobile applications, Pain assessment, User-friendliness, Mobile devices, caring, Healthcare professional, Hospitality, Patient, Feedback, Pain intensity scales

This page was intentionally left blank

List of figures

Figure 1. Programming Model	8
Figure 2. Android layers	11
Figure 3. key objects in Android app	13
Figure 4. Application launch in Android	14
Figure 5. Architecture of the Application.....	16
Figure 6. Application Launch Screen	17
Figure 7. File structure.....	19
Figure 8. Assets elements	20
Figure 9. Signup page.....	22
Figure 10. LOGIN	23
Figure 11. Authenticating	24
Figure 12. SIGNUP page as professional	25
Figure 13. SIGNUP page as professional	26
Figure 14. Home page	27
Figure 15. Information page	28
Figure 16. Pain scale pain.....	29
Figure 17. Pain questions.....	30
Figure 18. Option select.....	31
Figure 19. After selection	32
Figure 20. Search screen 2.....	33
Figure 21. Result page	34
Figure 22. PDF Names	35
Figure 23. PDF of the survey.....	36

This page was intentionally left blank

Table of Contents

ACKNOWLEDGEMENTS	I
ABSTRACT	III
LIST OF FIGURES	V
Chapter 1	
INTRODUCTION	1
1.1 Motivation	1
1.2 Objective.....	2
1.3 Structure of Document	3
Chapter 2	
BACKGROUND	4
2.1 Market Analysis	6
2.2 Advantage and Disadvantage of Existing System	7
Chapter 3	
APPLICATION DEVELOPMENT	8
3.1 Architecture.....	9
3.2 APPLICATION REQUIREMENTS	12
3.2.1 Functional requirements.....	12
3.2.2 Nonfunctional requirements.....	13
3.3 IMPLEMENTATION	15
3.3.1 Usability.....	21
3.4 TESTING	36
3.4.1 Unit tests.....	36
3.4.2 Mobile Device Space Test.....	37
Chapter 4	
CONCLUSION AND FUTURE WORK	38
REFERENCES	39

This page was intentionally left blank

1. Introduction

All of us are very afraid of pain, because pain, in addition to physical suffering and torments, leads to loss of control - over the situation and over ourselves. Pain destroys our body and destroys self-awareness. More often this happens with old age people. Most likely, the author had in mind the seal of suffering, which changes our inner world. Pain is a common somatic complaint, prompting individuals to seek professional help. A holistic understanding of pain requires the adoption of a bio-psychosocial framework. This is especially true when pain brings about major psychosocial changes and disruptions to the individual's life. Psychological factors play a crucial role in the experience of pain, making chronic pain a complex condition.

Our prototype has been developed on the Android platform. A real-time database is used, which allows the healthcare professional to closely follow the status of the patient. This will happen through firebase and connected all through the cloud.

In our application, we focused on the pain assessment for the patient. After logging in, the patient answers six set of pain scales. Then, the application makes the assessment of the patient's pain and assessment of the ability of the patient to do his/her normal activity through the day. The application can also be used by healthcare professionals. These users can get feedback information about the patient, which is archived in a cloud store.

1.1 Motivation

The healthcare professionals were using the old way to figure out how to make pain management for the patients. This was done through asking the patients about what they feel. Sometimes that was done without collecting or connecting the data of the patient. We thought on making an application that the patient and the healthcare professional can use for pain management, based on the patient answers on six pain scales. Once the patient answers this questions, the application gives the healthcare professional assessment for the pain of the patient and also gives feedback about the situation of the patient, allowing to share the survey through any platform. That was make us think this solution will solve a lot of the problems for the patient and the healthcare professional.

1.2 Objective

We are trying to make a difference in the medical world. We cared that our app handles the caring management for the hospital. So, after using that app, the healthcare professional will know what really the patient needs from the hospital, because we believe that this care is important to the patient, the hospital and the doctor.

Our application provides that one can see how much the patient needs care.

Our application is a mobile app, which has a very clear objective about what the difference we need to do in the medical world:

- Every user will have his own account and can login;
- If any user does not have any account, he/she can register to make accounting;
- Having a database for the hospital, which will include all the patients related to this hospital;
- There are six different pain scales the patient needs to answer to in order to identify what type of caring is needed;
- Every scale has its own values, from which the patient must choose. After the patient fills all scales, the healthcare professional is able to assess how much this patient needs caring;
- High security on the patient data so the patient did not have to worry about his data;
- Submit the result of the scale in the database so after answer the hospital could looking forward to the needed data from the patient;
- Sign out from the system.

1.3 Structure of Document

The rest of this report is structured as follows:

Chapter 2: Background and Related works

This chapter provides the analysis of the application in terms of related applications and to extract the out view for this application.

Chapter 3: Application Development

This section refers to the development idea, technologies used, design discussions, architecture, implementation methods, usability, testing results.

Chapter 4: Conclusion and Future works

This section concludes the prototype developed and enhanced design to be presented in future development.

CHAPTER 2

2. Background

Pain assessment remains a challenge to medical professionals and has received much attention over the past decade. Effective management of pain remains an important indicator of the quality of care provided to patients.

Pain scales are useful for clinically assessing how intensely patients are feeling pain and for monitoring the effectiveness of treatments at different points in time. In this application, a number of questionnaire sets have been developed to assess the specific types of pain. They are mainly used as search tools to assess the effect of a treatment in a clinical trial but may be used in an out-clinic setting. These scales are available for neonates, infants, children, adolescents, adults, seniors, and persons whose communication is impaired.

Our goal is going to track the patient status, relying on the pain scale and the ability to take survey through the mobile application:

- 1) The app has different pain scales to be sure that we have a real information which assesses the pain of the patient. The corresponding pain scales are Zarit, Barthel, Braden, Pain scale, Gijon and Minimental.
- 2) After the questioning part, the app provides a corresponding feedback along with a toast message as, for example, “thank you for answering” and/or, in some cases, apply the results and give a final score to the respondent.
- 3) The purpose of the model is the same as that of the basic model. The pain management model keeps track of the measurements for one patient. There are recordings of pain, sleep, mood, stress, activity.
- 4) The user of the application enters measurements for each of these. All the six measurements are checked separately for worrying trends and a corresponding warning-level for every interpretation of final score.
- 5) In the pain assessments, the app allows the healthcare professional to view and manage the patient’s history.

- 6) The application allows the healthcare professional to view and manage the patient's information as, for example, the profile of the patient and which patients have recently been updated. It is also possible to edit a patient's information, archive a patient and search for a patient.
- 7) The application has the ability to connect with the network and store the data on servers on the cloud. The app provides the updates of the assessment of the patient and the doctor can review that easily from any Android device from the entire world.
- 8) Finally, with using the application, it is possible to track the ability of the patient and knowing in what and when he/she needs a care and how much he/she needs that care.
- 9) We will have the ability to follow his progress - if patient's pain is going to get better or worse and how can the patient react with the surrounding environment in his/her normal daily activities.

2.1 Market Analysis

Name	Usage	Categories	Type of Pain data	Data Share/Stored
Pain Dairy	High	Track, Recall. Analysis the pain	High level pains	Yes
Manage my pain	High	Track, Recall. Analysis the pain	High end	Yes
Catch my pain	High	Pain community, Intelligent application, Pain diary	Migraine, lupus, back	Yes
Manage My pain pro	High	severity pain data tracking	Chronic back pain	Yes
Health Log	High	Flexible logging system/tracking any health event	Migraine, nausea, stress	yes
Pain relief	High	In terms of frequencies TBSW to melt pain through music waves	All kind of pains	yes
Essential pain management	High	Diagnoses and suggest drugs to be taken using record logbook	Assessing up to severity level pains	yes
Pain tracker/diary	High	Create comprehensive picture of pain/track/map/monitor	Complex regional pain syndrome chronic	yes

2.2 Advantage and Disadvantage of Existing Systems

Most of the existing applications are simple, minimally intrusive tests that are effective and easy to administer and score, in most cases. However, verbal, numerical, and visual analogue scales cannot be used in all patients. They may be ineffective in patients who have cognitive or motor problems, in patients who are unresponsive and in young children and elderly patients.

One major disadvantage of pain rating scales is that pain often cannot be accurately described and measured on the basis of severity alone. To describe pain only in terms of its intensity is like describing what we can see only in terms of light or dark, without regard for colours, patterns, or textures. But still these pain scales have to be standardized so that they can be used in all environments, not only in clinical setting.

CHAPTER 3

3. Application Development

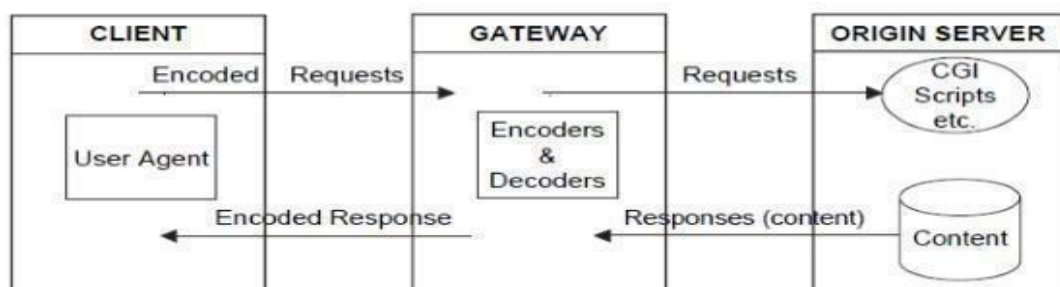
This application should be a way for the pain management which should treat the patient pain on time. Our application manages the pain through assessing the pain of the patient by pain scales. Then, the doctor or other healthcare professional can decide on the type of care needed for the patient.

This application is developed under the platforms

- **Native Android development with JAVA**
- **Gradle version-4.6**
- **Minimum SDK version-22**
- **Fire store -Firebase authentication**

IDE (Interface Development Environment)

The IDE has a lot of source code models that uses the Android IDE, Android virtual device (Emulator) to run and debug apps in the android studio, frameworks and samples, compilers etc. This Android IDE uses template-based wizards to create common Android designs and components, where the developer can layout and design the user interface of the application by adding views such as buttons, table views, and text views onto scenes. It has also a rich layout editor that allows users to drag and drop UI components, options to preview layouts on multiple screen configurations and, also built in support for Google cloud platform, enabling integration with firebase cloud messaging and google app engine. The Android IDE has Lint tools to catch performance, usability, version compatibilities.



Programming Model

Figure1. Programming Model

Design Criteria and Discussion

During the development of the application, in order to achieve adequate and usable prototypes, we have followed some criteria, as discussed in this section. We have focused on the following points:

- Find existing technological applications about pain management;
- Make a plan for the application and identify the limitations and make risk management and timeline for the project;
- Going through the implementation of the application through making the layout with the Android IDE, build the fire database and connect it with the application, then connect it with the server and all the results must follow the specification and meet the objectives.

3.1 Architecture

The Pain Management application is a Mobile application, made for caring about the patient. The application was made as assessment for the patient pain and how that pain affects the patient life's activity. The application has been developed on the Android platform. It includes new information sources and services to make the system highly reliable.

The pain management application gives the healthcare professional the chance to review the assessment of the patient and the healthcare professional has the chance to archive the patient information.

The real-time database has been done through Firebase, which provides instant upload for the patient's pain scale answers.

List of features

- Allow the user to login and signup to the survey in the app;

- After successfully logged in in the app, the user can make entry to the app by clicking new entry;
- Scales Available: The patient will be provided with the set of scales like Zarit, Barthel, Braden, Abbey Pain, Mini Mental and Gijon;
- Show feedback: after one user completely answers all available questions, the user will be provided with the final score and feedback.

Firestore

In the pain management application, Firestore let us store the patient data and to update the pain management assessment. Also, it provides the synchronization of data between the users in real-time and this turn easy for the users of the application to access the data and it helps the users to collaborate with one another and whenever users (patients) update data in the real-time database. It stores the data in the cloud and simultaneously notifies and all the users which connected together, like the patient and the healthcare professional. Also, the real-time database is also optimized for offline use. Whenever a user loses the connection, the database SDK uses a local cache on the device to serve the user and store the changes of the data for the patient/professional.

Also, patients' information is secure because Firestore provides database security rules, which will be used to secure the data of the patient. These rules are used to specify who has access to what pieces of data and how the database should be structured for that. The security rules are securely stored with the real-time database on the servers. Also, the Firestore real-time database is hosted in the cloud.

Android language

Android apps can be written using Kotlin, Java, and C++ languages using the Android software development kit (SDK), while using other languages is also possible. All non-JVM languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support.

Some languages/programming tools allow cross-platform app support.

Android Layer Architecture

Android is a mobile operating system, based on a modified version of the Linux kernel and other open source software as shown in Figure 2.

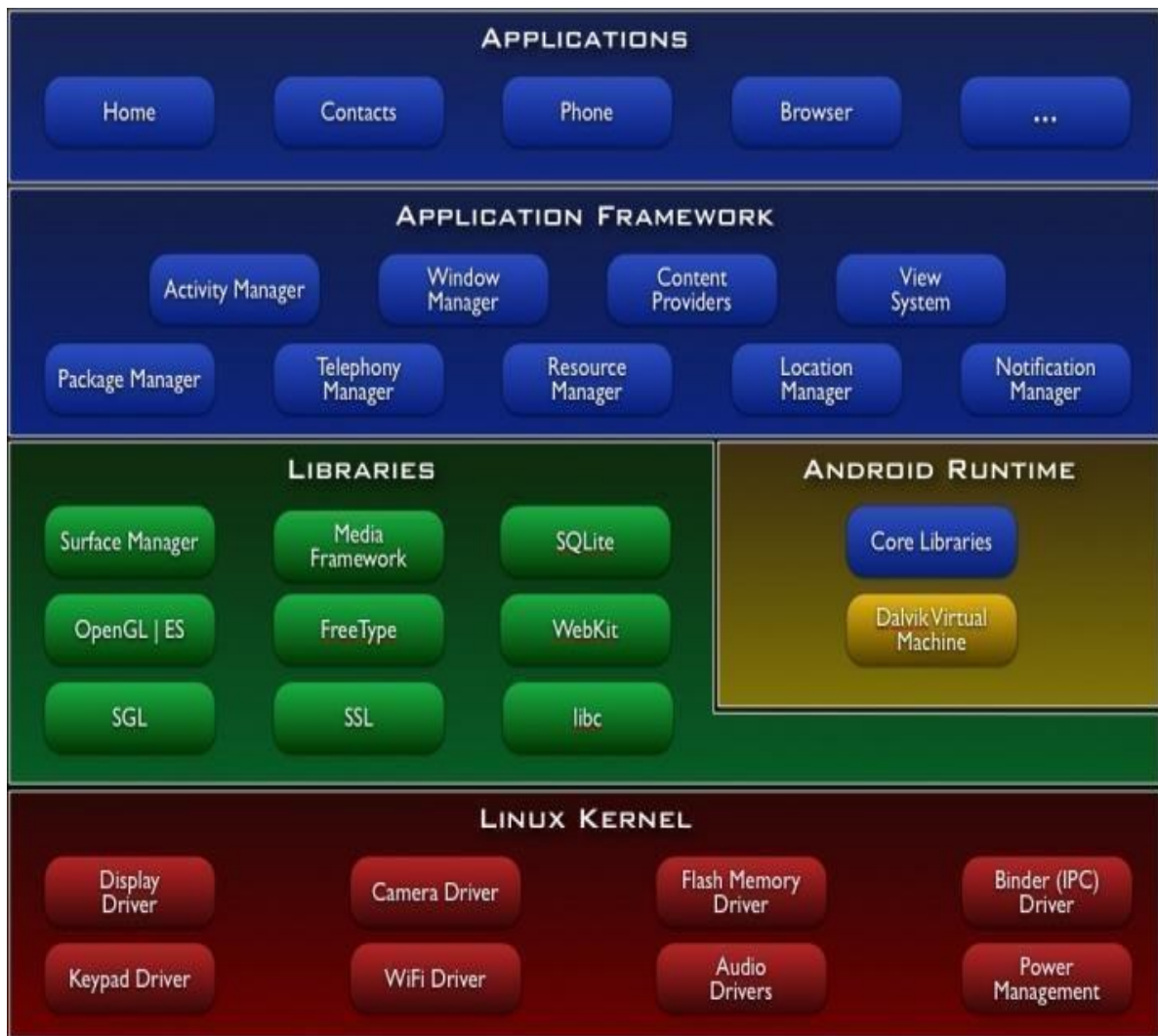


Figure 2. Android layers

The concise architecture of Android can be depicted into four layers: the kernel layer, the middleware layer, the framework layer and the application layer. The Linux kernel is the bottom layer of the Android platform which provides the basic functionalities of operating systems such as kernel drivers, power management and file system. The layer above the kernel is called Android middleware layer, which contains essential elements of Android as a mobile platform.

Within the native components, the Hardware Abstraction Layer (HAL) defines a standard interface to bridge the gap between hardware and software. Compared with the drivers located in the kernel layer, Android HAL holds most of the hardware vendor specific implementation, for example, the APIs of audio device and camera. Other two key component in the native components part are the native libraries and daemons which are written in C/C++.

The native daemons handle all interaction with the system in native level. The native libraries, like SQLite, Web kit, SSL, and OpenGL, could greatly enrich the functionality and compatibility of Android platform for the development purpose (Figure 3). The Android runtime system contains the core libraries and runtime environment.

3.2 Application Requirements

3.2.1 Functional requirements

- Allow the user to answer the pain scales and give the total of the pain;
- Allow the user to have a new entry; the user can make entry to the app by clicking new Entry;
- Allow the user to update a new pain assessment and archive the old pain assessment.

Information retrieval in Functional Requirement

- User can retrieve a total assessment for his/her answers on the pain scale;
- After filling the total assessment of pain, the user gets always feedback.

3.2.2 Non-functional requirements

- Authorization for sending notification for the phone/authorization for using email;
- Internet access is needed for upload of the data of the patient;
- Internet access is needed to review the update of the pain assessment of the patient.

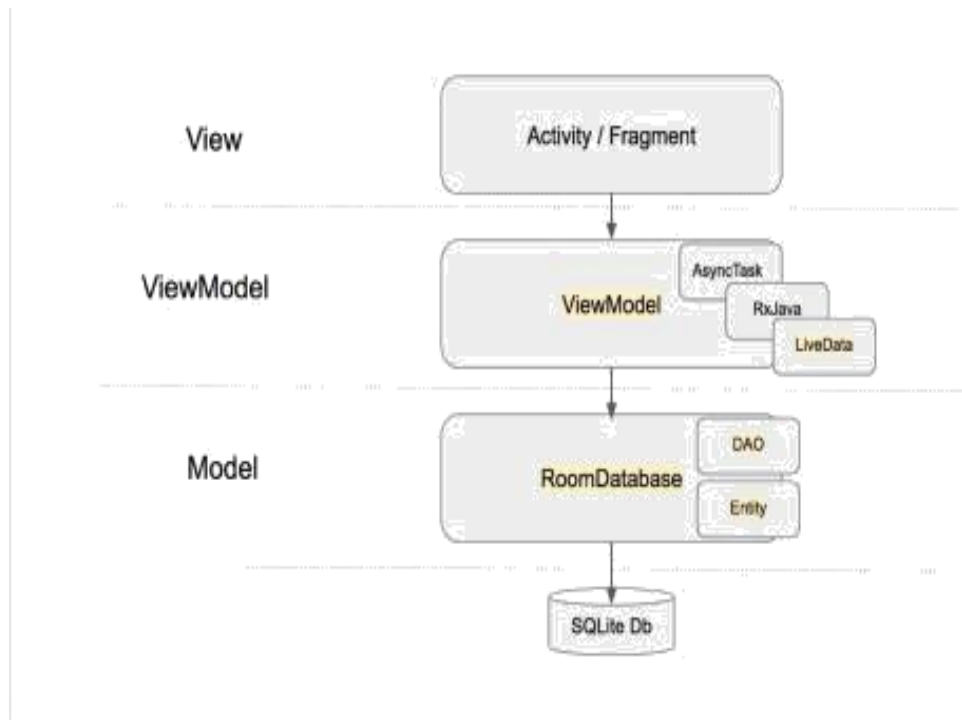


Figure 3.key objects in Android app

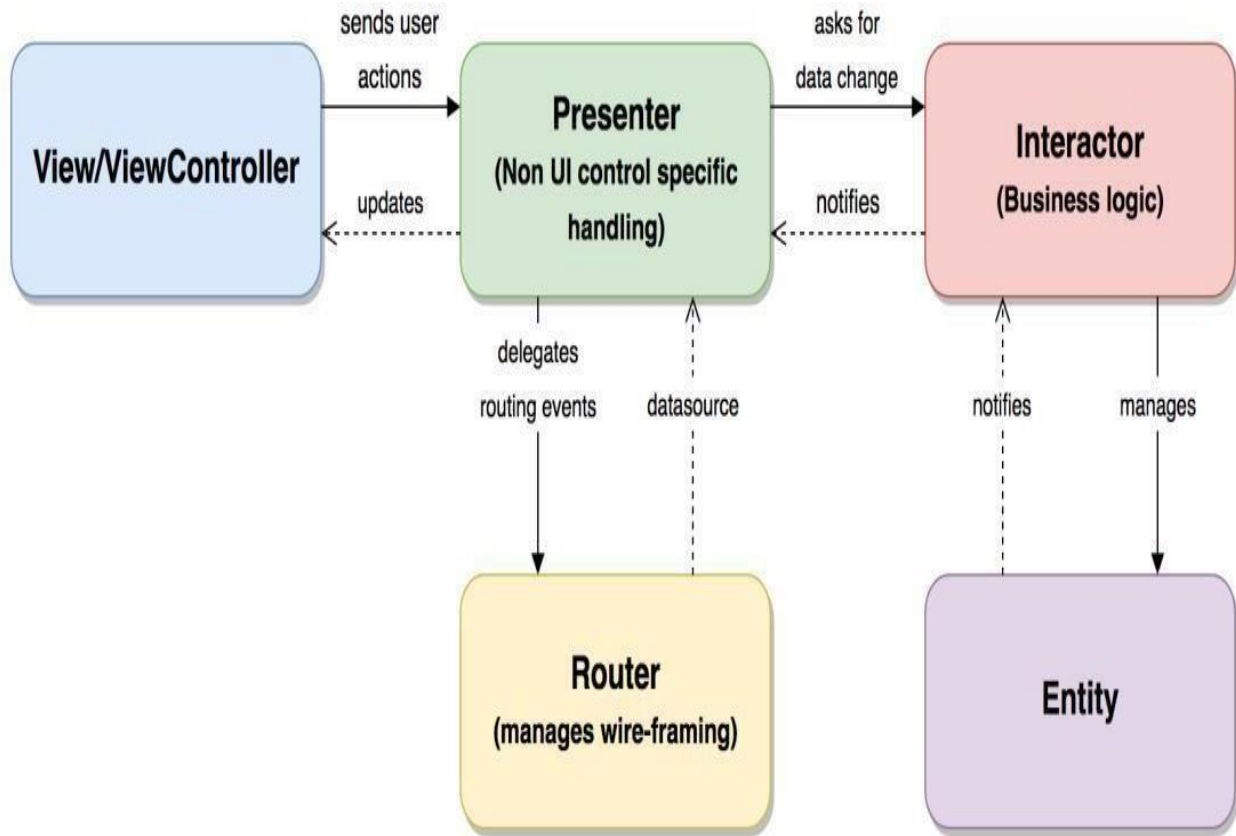


Figure 4. Application launch in Android

3.3 Implementation

In the prototype of the caring application, implementation concerns on making simple design to prove the idea of the concept of the application. Through our simple design, we can make a test between the users to check the interaction of the data between them distinct profiles as caretaker and patient.

There are many different processes in the application that relay on more than technologies. The app needs the network to store the patient's data on the cloud so that the doctor or the healthcare professional can review the data from any place in the world. Also, the Android studio was used to develop the application. We use servers to work with Firebase so that the data is instantly stored. Also, it is used to track the assessment of the patient's pain and to update their pain assessment, so that the doctor or the healthcare professional can take the right decision in the right time.

Modules in Application

- Login and registration;
- Profile of every user to the app and ability to edit the profile;
- Pain assessments according to the assessment of the pain scale that the patient fills;
- Updating the pain assessments and achieving it;
- Search for the patient.

Technology used

Java is the official language of Android development, meaning it is the one that has the most support from Google and the one that most apps on the play store are built with, and has a set of tools to facilitate Android development specially.

The app is compatible with devices that have Android KitKat or higher as an operating system and have been tested for Android phones and Android tablets.

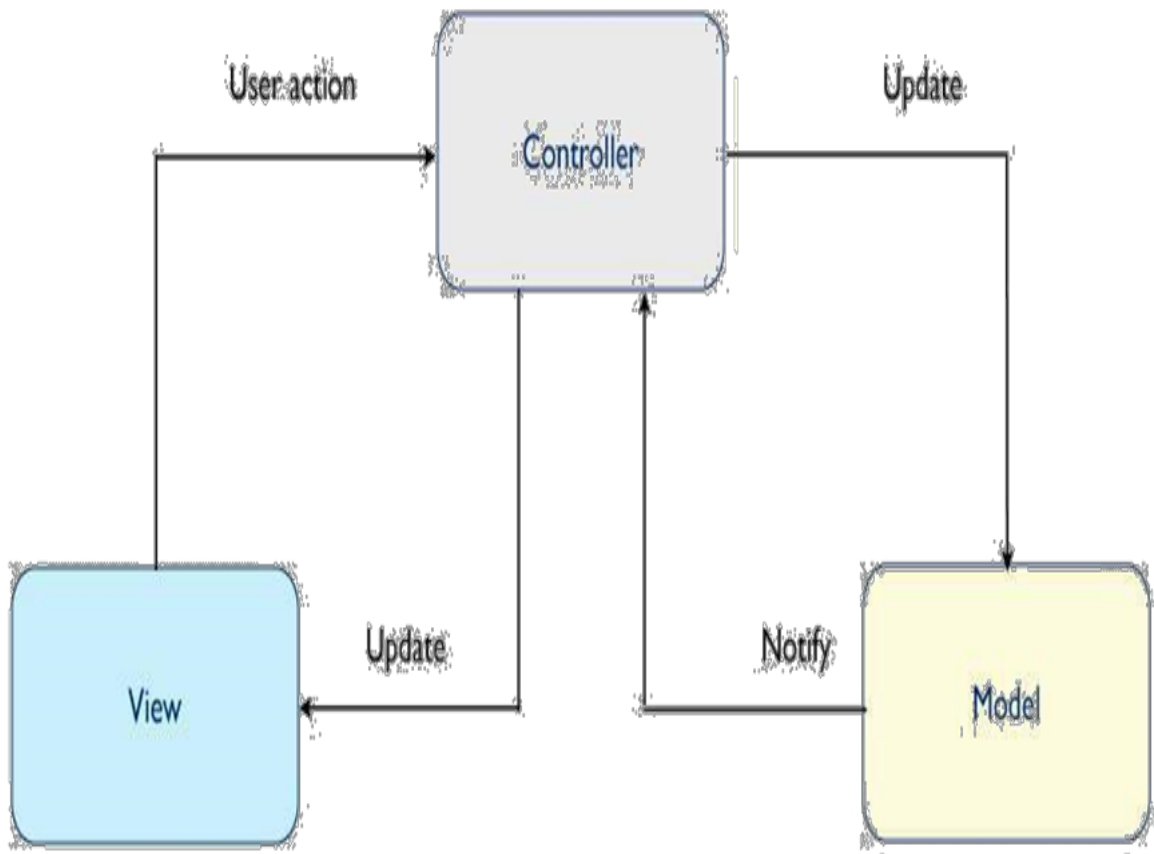


Figure 5. Architecture of the Application

The architecture implemented was the Model View Controller (MVC). This architecture divides the application into three sections (Figure 5).

Model: This is the lowest level of the architecture; in this level the application uses core data to maintain all the data of the application.

View: This level shows all the displays that interact with the user. The applications have the Update of the information of the pain assessment of the patient, Search and Detail.

Controller: In this level, all the communication of the application with the Model and View Controller is done. In this view the application manages all the API, frameworks and classes that are needed on the application.

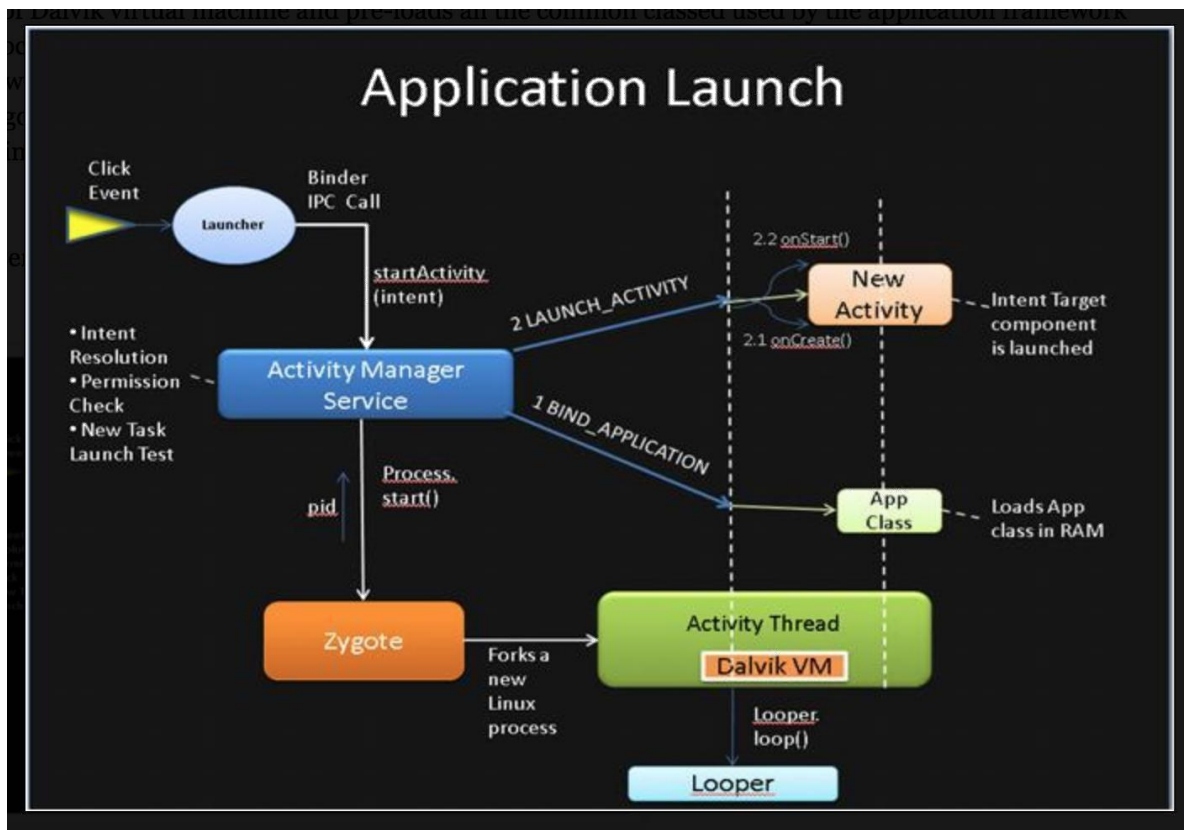


Figure 6. Application Launch Screen

- Main Storyboard - It is primarily responsible for the management of the entire interface used and define the different views of the application system and the navigation between the controllers;
- Modules - Using these classes, the structure of the application is defined, which serves to store all the information and generates content presented to the user side;
- View Controllers (UI builder) - This contains the classes that are associated with each of the views and the classes logic;
- Protocols - It contains the protocols of classes which can be implemented in the various controllers;
- Data Providers - Classes that integrate the online services with the device and that are also responsible for sending and receiving data from the web and for saving and retrieving the data within the devices;
- Parsers - This contains the classes of parser files and their names and convert this data to lists, according to the application data model;
- Supporting Files - Are a set of files and classes that support the project implementation.

File structure

To better understand the implementation of the application, the following structure of files in Android studio will be described as follows (Figure 7):

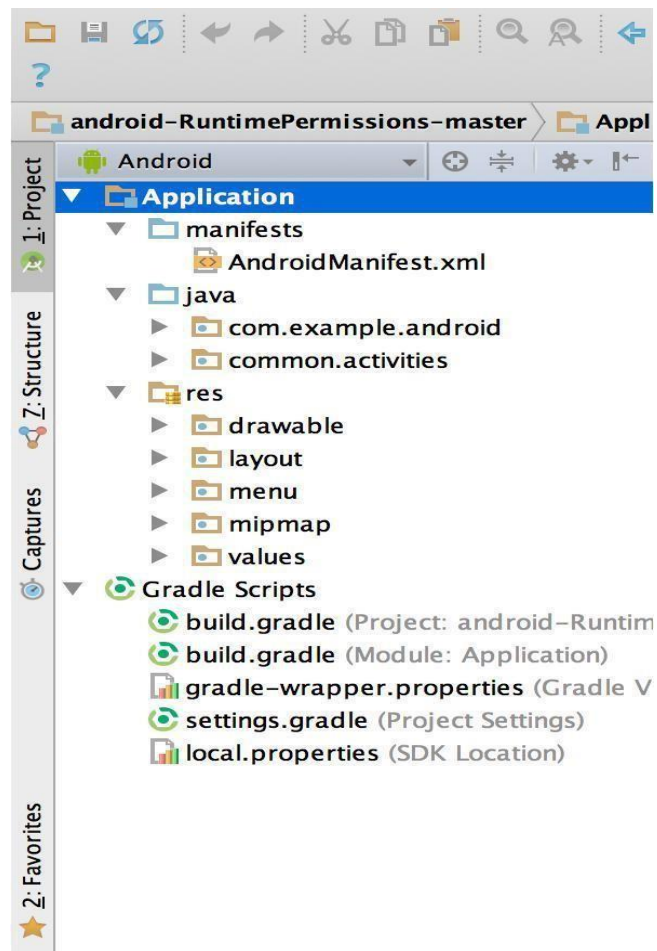


Figure 7. File structure

Gradle file – Used to manage dependencies and discover other libraries to use.

Main storyboard - This is composed of different sequence of scenes, each one representing a view controller and its view scenes. In addition, the storyboard enables us to connect a view to its controller object and to manage the transfer of data between view controllers.

Frameworks - It is also called as framework bundle and its content can be accessed using Core foundation bundle services or the NS bundle class. This provides a library that can able to perform specific tasks.

App delegate – Object that receives notifications during the runtime of the application and that ensures that the app interacts with the system (Figure 8).

Assets - It is the catalogue of different elements combined in Android Studio.

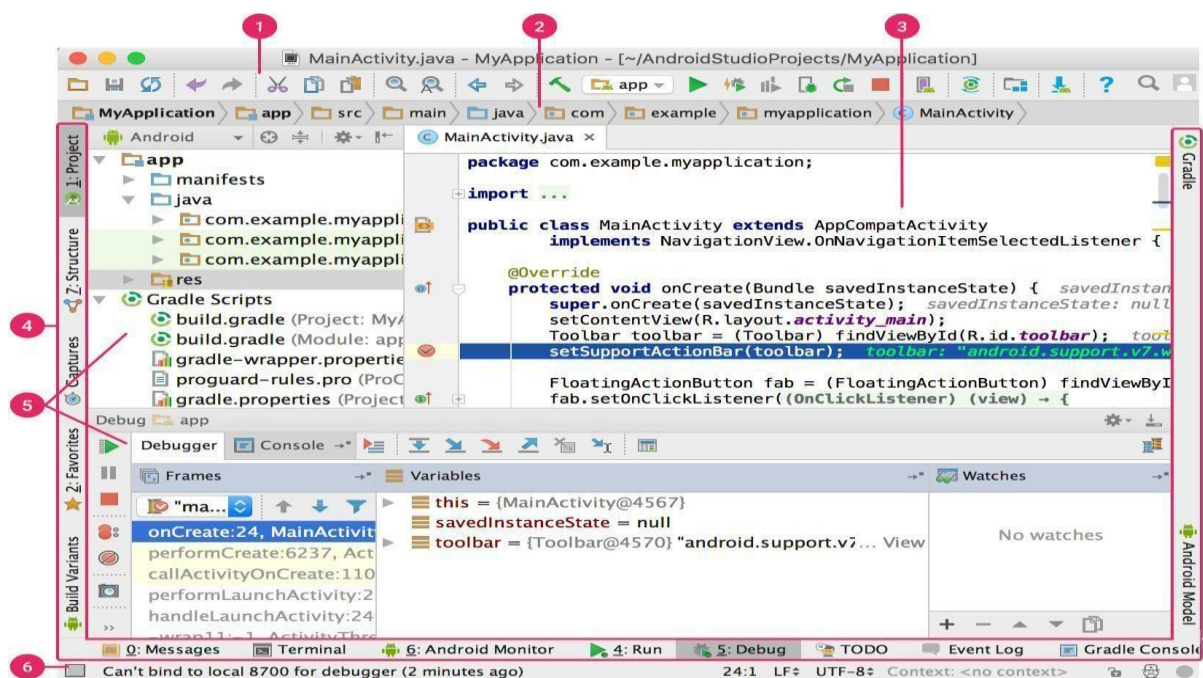


Figure 8. Assets elements

Android Emulator

Android Emulator is a software which can emulate or run the Android operating system in the desktop or laptop for the users who want to use Android or for developers.

The Android emulator is typically used for deploying apps that are developed in the IDE without actually installing it in a device. Android emulators such as Blue stacks can run android apps wherein which emulators like AVD.

3.3.1 Usability

Login

It has an email form and password form so the user can enter the application.

It has the button *forget password*, so that the user can recover the password.

It has a *Login* button, so that when the user fills the email and password and they exist in the database, the user can make a login inside the application to enter the second stage.

It has two tabs, one to login as professional and one to login as patient for the application.

Also, it has a button called Signup. If the user has no email and password in the database, he/she must signup first (Figure 9).

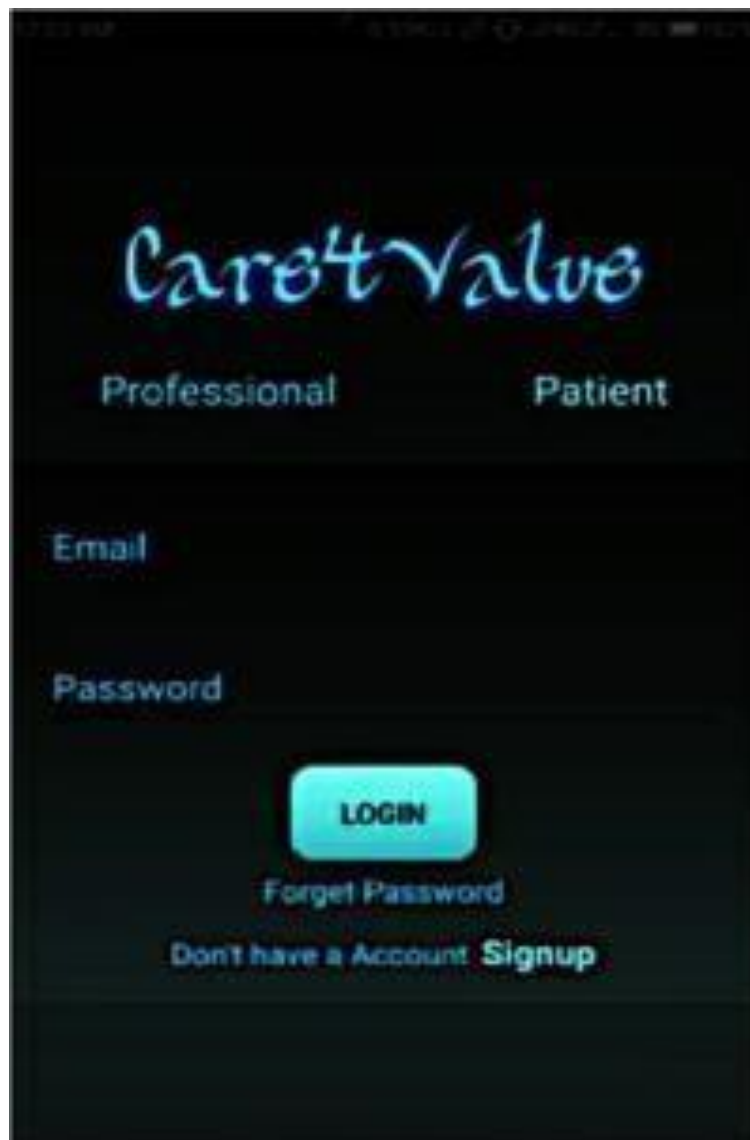


Figure 9. Login/Sign up

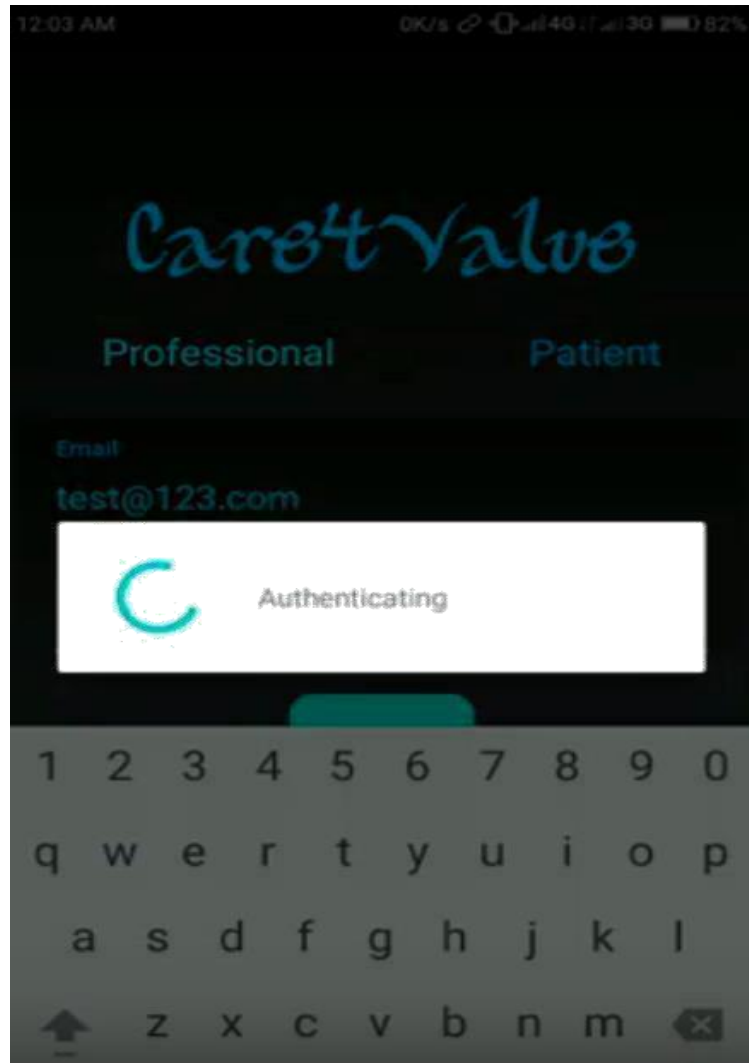


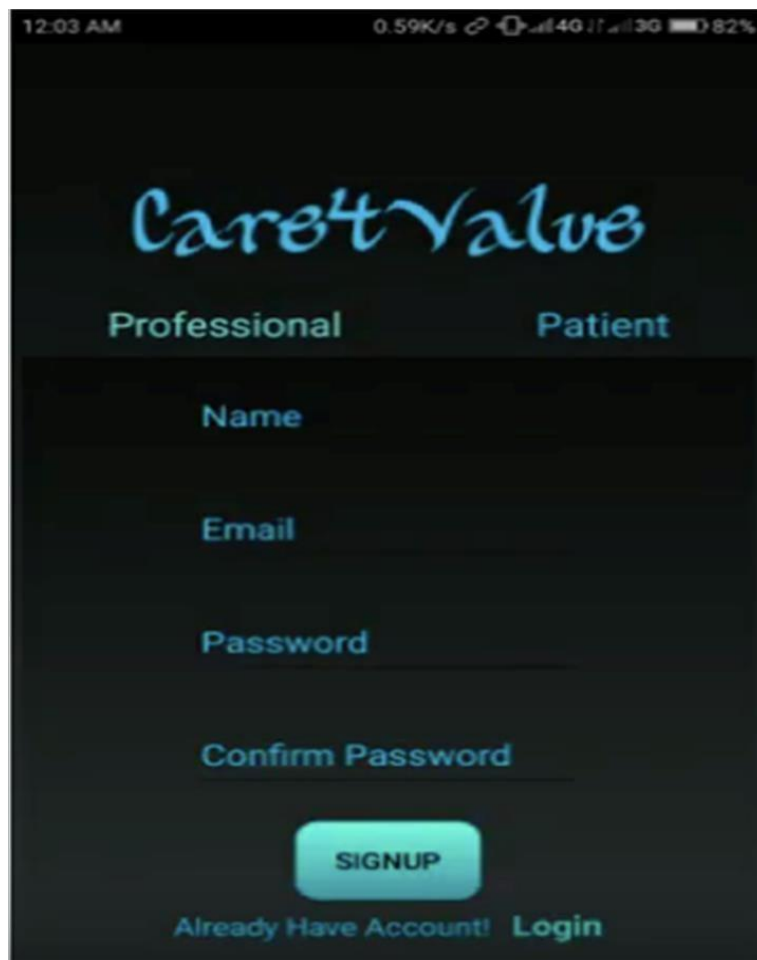
Figure10. Authentication

After the user enters the email and password, he/she will enter the next intent or process. The application will make the authentication of the data (Figure10).

Signup

This page gives the user a way to be a user in the application. The user enters the data, which is stored on the database. After this, the user can enter from the login page (Figure 11).

It has four text filling columns while the user is signing up as healthcare professional. The first form is for name, email, password and password confirmation. If the user is signing up as a patient, he/she needs to add the name, birthday, email and password.



12:03 AM 0.59K/s 40% 30 82%

Care4Value

Professional Patient

Name

Email

Password

Confirm Password

SIGNUP

Already Have Account! Login

Figure 11. Signup page for professional/patient

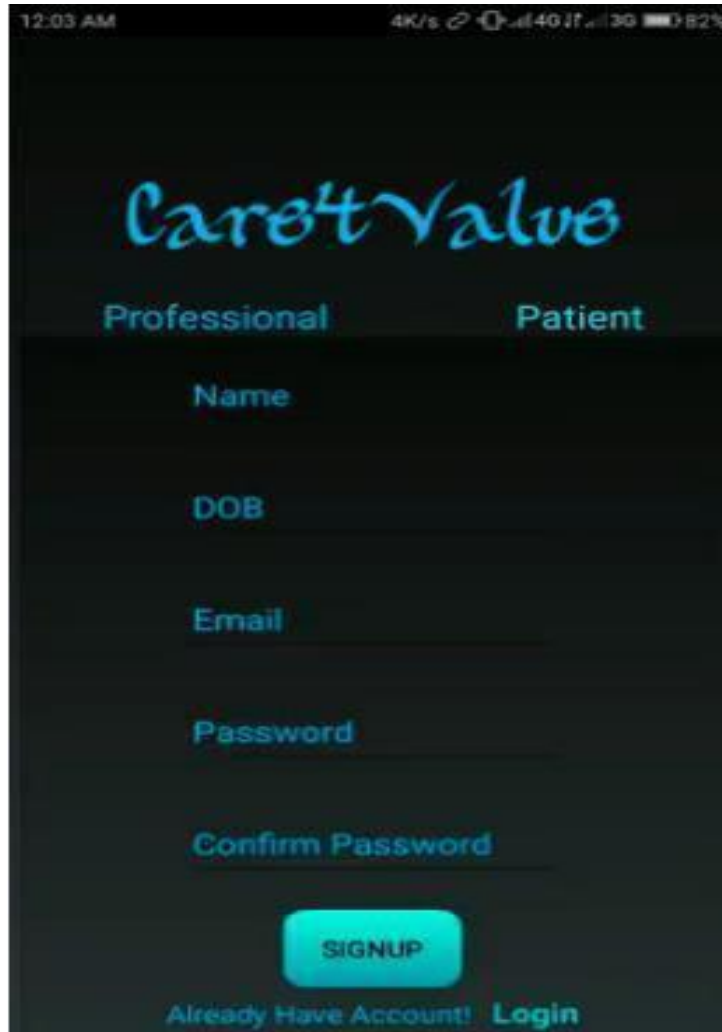


Figure 12. Sign up page as professional

Home screen

This is the most important screen in the application (Figure13). After the user makes a successful login, on this page he/she will get a small toast message notifying a successful login.

On the top right, there is a small button for logout. After that, the user finds a button called “New Entry “, where the user enters information to make the survey progress.

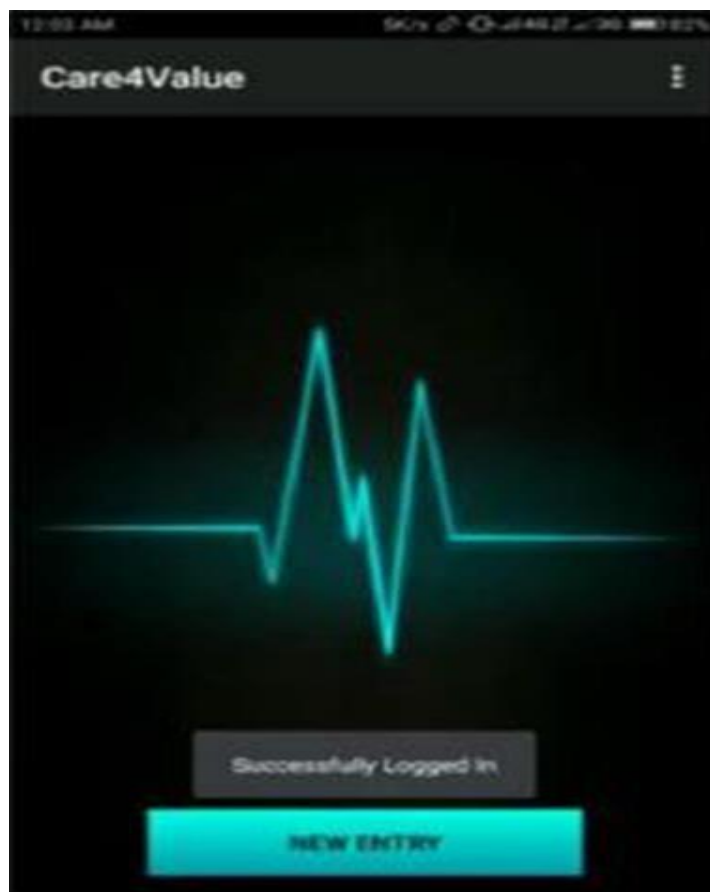


Figure 13. Home page

Information page

This page has text fields to be entered, which are used to track the specific patient with a unique ID as reference (Figure14), so that the doctor or the professional user can access the patient details.

There is a button “*Proceed*” to further access, after filling up the information needed on the application.



Figure 14. Information page

Pain scale page

This screen has six different set of pain scales as the visual rating scale so that the application can assess the results of pain for the patient through which the user can make a self-pain management.

We have six buttons corresponding to scales Zarit, Barthel, Abbey Pain, Braden, Mini Mental and Gijon (Figure 15). Every button has its more specific questions and every page has its list of questions in drop down style.



Figure 15. Pain scales

Pain's question

This page includes a list of options and the patient needs to select from this option. He/she needs to choose from this selection related to pain. For example, can they dress alone or need support? In the end, the application calculates the total score of the survey. Also, this page has two buttons.

The first button is called "Previous" and is needed to return to the previous question if the user wants to edit an answer. The second button, called "Submit", is needed to finish answering this page (Figure 16).

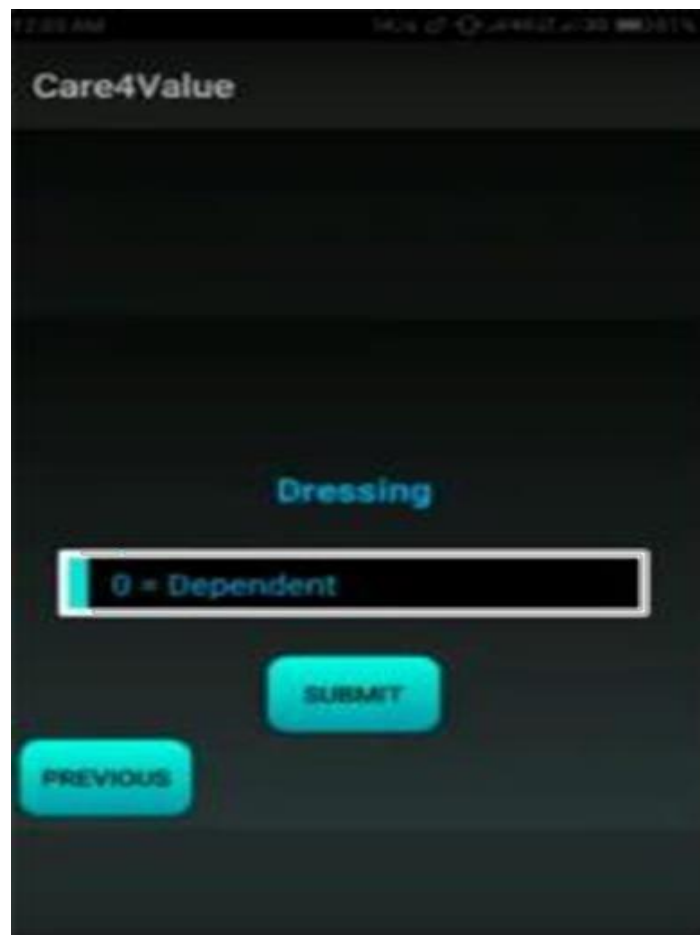


Figure 16. Pain questions

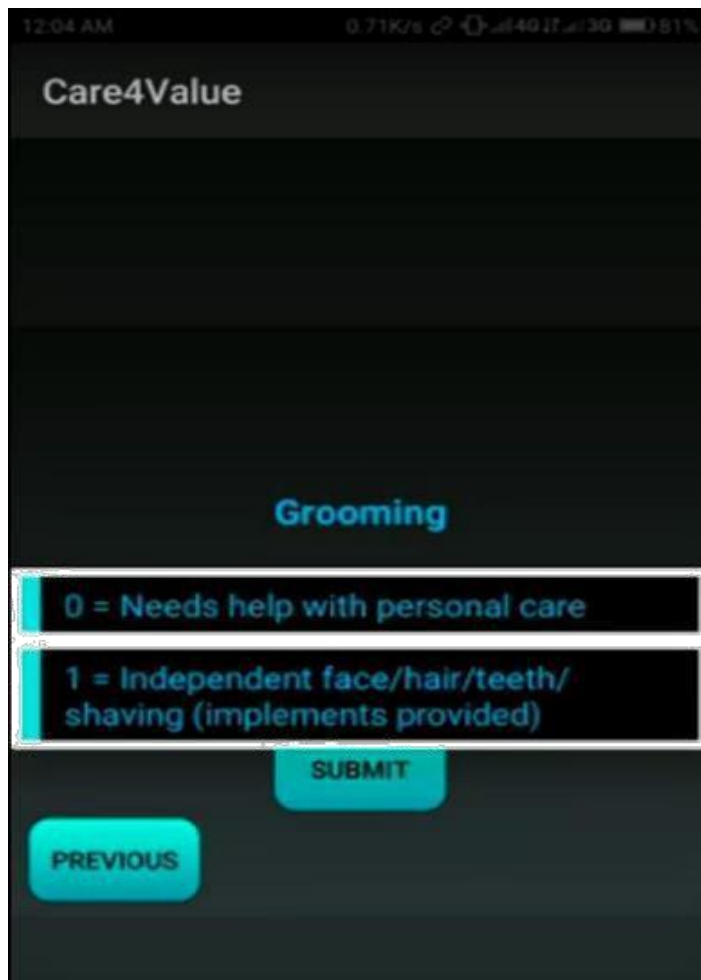


Figure 17. Option selects

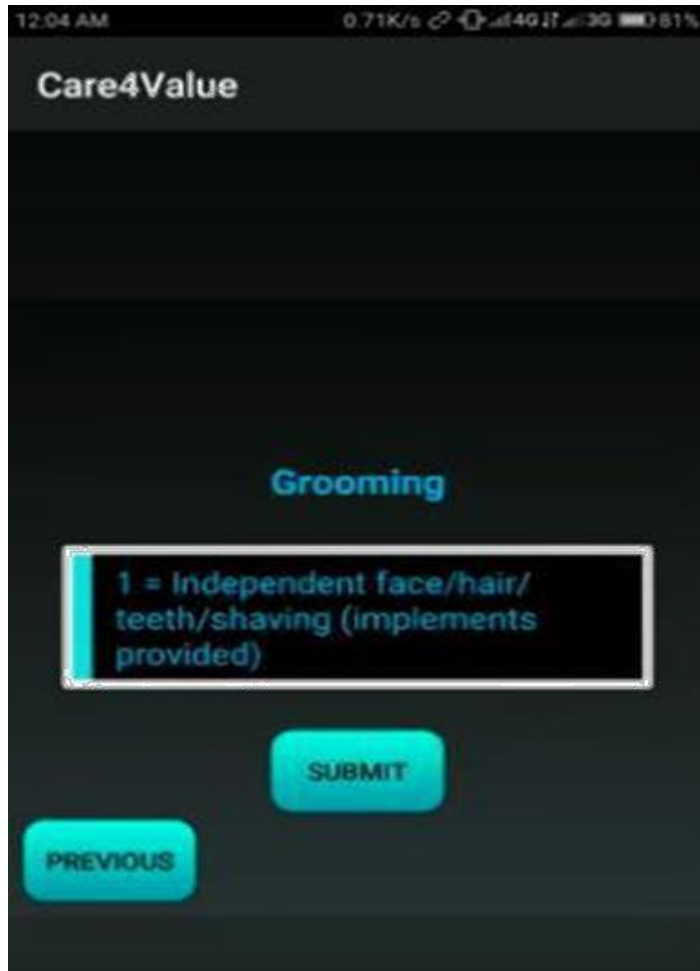


Figure 18. After selection

Special information

At the end of the survey, the patient gets a message, so if they need to add any special feedback or specific treatment to be given.

After the user fills the form, he/she is able to click *continue* to enter the last page with an interpretation of score (Figure 19).

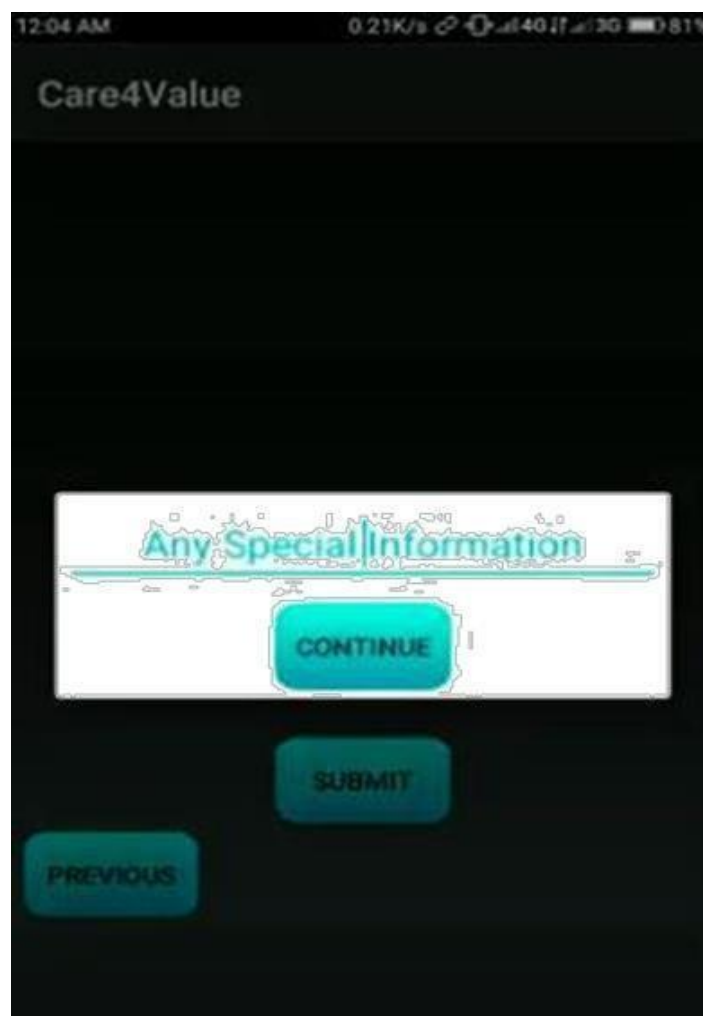


Figure 19. Feedback screen

Result page

This page presents the result of the survey filled by the user. The application assesses how much the user needs a healthcare professional. Firstly, we can find a small toast message as “Thank you” for filling the survey (Figure 20).

This page provides the user with the name of the patient, then the assessment of the scale along with the total score of the patient. Then, it shows the description for that score: low/moderate/high level pain. With the “Share” option enabled (Figure 21), it is possible to share the user results. But this feature is not found in the patient user login.

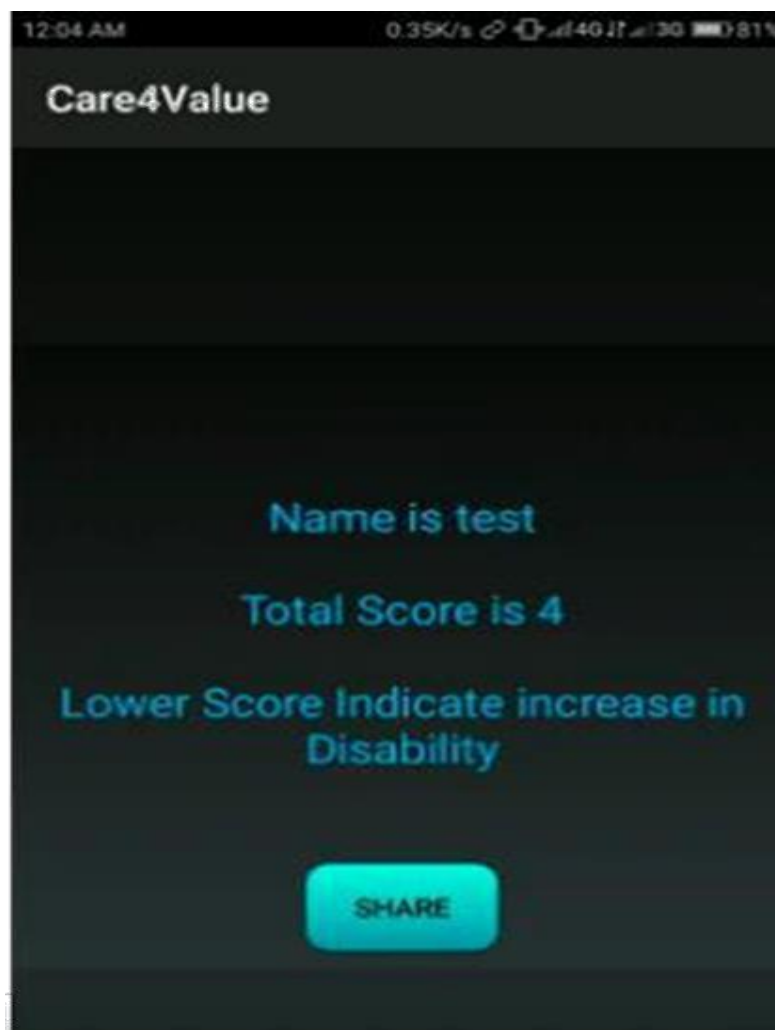


Figure 20. Result page

Sharing with social networks

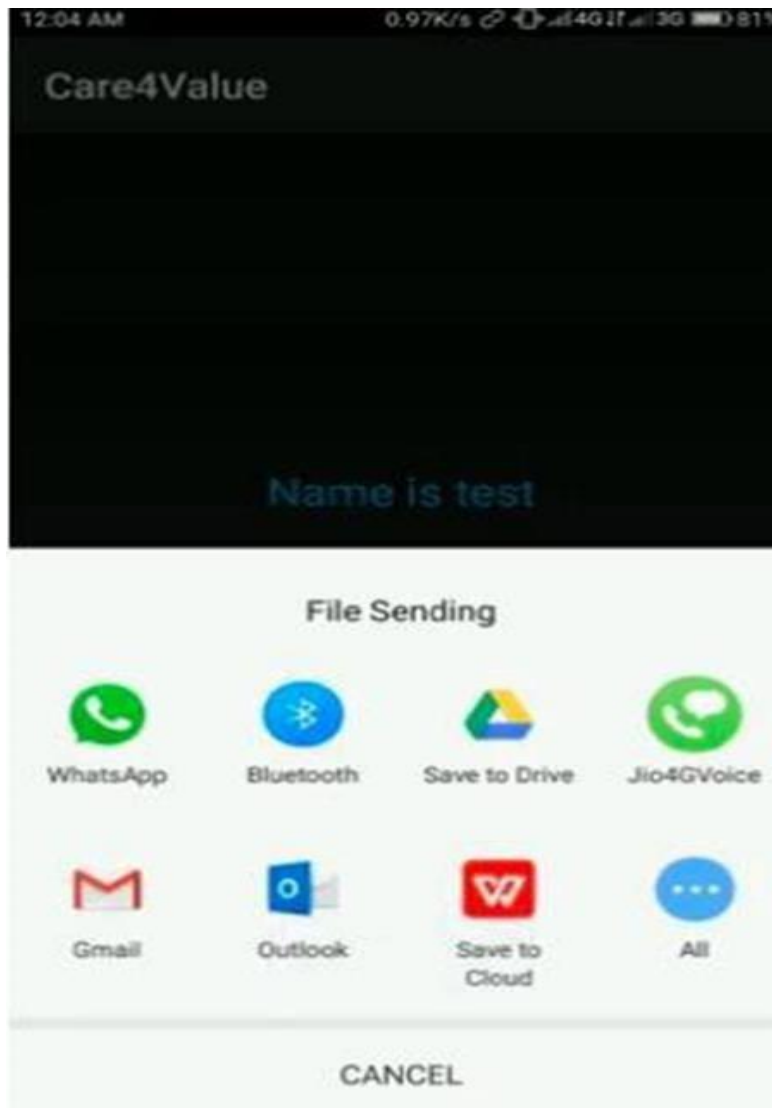


Figure 21. Sharing with Social media

Pdf Export

After the professional user completed the survey, he/she is able to export the survey as a PDF file. Also, the PDF file will have the name of the patient and all details.



Figure 22. PDF export



Figure 23. PDF of the survey

3.4 Testing

During development until completion of the application, several tests were performed in order to eliminate errors found and guarantee the desired functionalities. The tests performed also served to evaluate if the application meets the requirements specified.

It also includes some pattern of user testing with potential health professional users with caregiving burden and few nonprofessional users.

All tests were performed with tools provided by Android Studio (API 28) as target version, Android emulator, and with Android phone.

3.4.1 Unit tests

The unit tests were done during the development of the application by the working group elements. These tests were always done as the functionalities were being implemented.

These tests were used for error detection and application optimization to ensure maximum software quality, usability and compatibility with the size of different android enabled devices.

The Care4 value application is ready to work on Android phones and Tablets.

3.4.2 Mobile Device Space Test

One of the limitations of mobile devices is the available storage space. The memory of devices compatible with this application can vary from 8, 16, 32, 64 or 128 Gb.

Because many users store photos, documents, movies, and applications on their devices, often taking their devices' storage capacity to the limit, one of the factors to consider is the size of the application.

In order to avoid excessive use of storage space, some care has been considered.

The space occupied by installing the application on a mobile device is about 6.96MB and exported pdf data will share 1.8MB.

CHAPTER 4

4. Conclusion and Future work

The target of the proposed Android application is to assist the user in the managing his/her pain. The user inputs several different measurements in different intervals of time, which helps to check the trends that warrant concern and, when a trend is detected, a notification or warning is sent to the patient indicating that some action of care must be taken in response to that alert information.

This application has a set of pain scales that assists the user in the effective management of pain, by providing timely feedback to sufferers and improving self-management strategies.

In future, we can able pain coaches to add custom reminders with specific time and day to notify patients to log in the app to have a regular survey. Thereby, we can be able to have track of records and can also try to enhance this app with a dashboard feature so that the patient's progress can be tracked with greater ease and accuracy.

References

- [1] Market Analysis: pain diary is one of the biggest pains for the pain management in USA, <http://mypaindiary.com/>
- [2] Kumar P, Tripathi L. Challenges in pain assessment: Pain intensity scales. *Indian J Pain* 2014; 28:61-70
- [3] The pain management and the treatment: <https://www.webmd.com/pain-management/guide/pain-management-treatment-care>
- [4] Portelli P, Eldred C. A quality review of smartphone applications for the management of pain. *Br J Pain*. 2016;10(3):135
- [5] Vardeh DV, Edwards RR, Jamison RN, et al. There's an app for that: mobile technology is a new advantage in managing chronic pain. *Pain* 201
- [6] Rosser B, Eccleston C. Smartphone applications for pain management. *J Telemed Telecare* 2011; 17:308
- [7] Patrick K, Griswold W, Raab F, Intille S. Health and the mobile phone. *Am J Prev Med* 2008; 35:177–81
- [8] Plante TB, Urrea B, MacFarlane ZT, et al. Validation of the Instant Blood Pressure Smartphone App. *JAMA Intern Med*. 2016
- [9] Blodt S, Pach D, Roll S, Witt CM. Effectiveness of app-based relaxation for patients with chronic low back pain (Relax back) and chronic neck pain (Relax neck): study protocol for two randomized pragmatic trials. *Trials*. 2014