

Faculdade de Engenharia da Universidade do Porto



Location Based Services for Everyone

Diogo Dias Júnior

Master in Informatics and Computing Engineering

Supervisor: Jorge Alves da Silva (PhD)

Co-Supervisor: Luís Filipe Pinto de Almeida Teixeira (PhD)

28th June, 2010

Location Based Services for Everyone

Diogo Dias Júnior

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Rui Pedro Amaral Rodrigues (Invited Auxiliar Professor)

External Examiner: Maximino Esteves Correia Bessa (Auxiliar Professor)

Supervisor: Jorge Alves da Silva (Auxiliar Professor)

31st July, 2010

Abstract

During the last decade, mobile devices have been gaining customers at a large scale. Today's mobile devices enable users to do more operations than with a computer. However, typical users are not very skilled and usually they want to access the information they seek with the minimum number of actions. Users should be able to easily access the information without worrying about technological details.

Currently there is much information and content available on the network. However, only more experienced users can find this information in the vast network that is the Internet. Over the past years people have tried to associate meta-tags to content to be possible through automation the deduction of knowledge implicit in the content. One of these examples are the local services and particularly the availability of these services on the Internet. When these services are made available, there is an intrinsic attribute - its location. Thus, it is desirable that these services are indexed by their location and made available by an aggregator with tools that enable categorization and search. However it is imperative that this system can be used by everyone.

With all these characteristics in mind, the goal is to create a system with current technology that allows people to easily create a service based on your location. Since the most important thing with this kind of services is always the content and the way it is shown, people should have the ability to create and edit the content of the service as they wish.

From the consumer side of the service, there must be a total consistency between the created service and the consumed service, and it is also necessary that the services are presented in a categorized and filtered way. The system developed is called MoLand and is a system that uses the mobile device and the web browser to allow the creation of location-based services by everyone in an easy and intuitive way. With MoLand, the users are also able to consume these services using an augmented reality application with a highly intuitive and flexible interface.

MoLand works on a client-server platform where clients (mobile devices) are connected to the central server to know what are the landmarks and their services.

Location-based services and how they are created in a graphical way are key issues in this project.

Resumo

Durante a última década, os dispositivos móveis têm vindo a ganhar consumidores em larga escala. Os dispositivos móveis dos dias de hoje permitem aos utilizadores executar mais operações do que um computador. No entanto, os utilizadores normais não têm muita agilidade ao manipular a tecnologia e normalmente querem aceder à informação que procuram com o número mínimo de acções. Os utilizadores devem ser capazes de aceder à informação sem se preocuparem com detalhes tecnológicos.

Actualmente há muitos conteúdos com informação valiosa disponibilizada na rede. No entanto apenas utilizadores mais experientes conseguem encontrar essa informação na grande rede que é a Internet. Ao longo dos últimos anos tem-se tentado associar *meta-tags* ao conteúdo para que seja possível através de processos automáticos a dedução de conhecimento implícito existente no conteúdo. Um desses exemplos são todos os serviços locais e particularmente a disponibilização destes serviços na Internet. Um atributo intrínseco de um serviço local é a sua localização. Assim, é conveniente que estes serviços sejam indexados pela sua localização e disponibilizados num agregador com ferramentas que possibilitem a categorização e pesquisa. No entanto, é obrigatório que este sistema possa ser utilizado por todo o tipo de utilizadores.

Com estas premissas evidenciadas, surge então a necessidade de criar um sistema com tecnologias actuais que permita aos utilizadores criar serviços baseados na sua localização de uma forma fácil. Como neste tipo de serviços a parte mais importante é sempre o conteúdo e a forma como ele é mostrado, é imperativo que se permita às pessoas terem a possibilidade de criar e editar o conteúdo do serviço.

Do lado do consumidor de serviços é necessário que exista uma consistência total entre o serviço criado e o serviço consumido, assim como é também necessário que os serviços lhe sejam apresentados de uma forma categorizada e filtrada.

MoLand é um sistema que usa um dispositivo móvel com sistema operativo Android para permitir a criação de serviços baseados na sua localização por toda a gente de uma forma fácil e intuitiva. Com o MoLand, os utilizadores podem usufruir de uma aplicação de realidade aumentada com uma interface intuitiva e flexível para consumir os serviços.

O MoLand actua sobre uma plataforma cliente-servidor onde os clientes (dispositivos móveis) se ligam ao servidor central para saberem quais são as *landmarks* existentes e os seus serviços.

Serviços baseados na localização e forma gráfica como os serviços são criados, são as palavras-chave deste projecto.

Acknowledgements

I would like to thank everyone at Fraunhofer Portugal Research for the motivation and support they had shown. My special thank to my co-supervisor Luis Teixeira and to my supervisor Jorge Silva, for their patience and availability. I would also like to acknowledge my family, especially my mother who has been giving me support throughout my life, and my girlfriend and friends for their comprehension during this journey.

Diogo Dias Júnior

Table of Contents

List of Figures.....	xii
List of Tables	xiv
Abbreviations.....	xvi
Chapter 1	1
Introduction	1
1.1 - Motivation	1
1.2 - Problem Description.....	1
1.3 - Objectives.....	2
1.4 - Document Structure	2
Chapter 2	3
State of the Art	3
2.1 - Introduction.....	3
2.2 - Geographic Information Systems	3
2.3 - Geographical Location on Mobile Devices	4
2.4 - LBS Components	6
2.5 - Application Examples	8
2.6 - Augmented Reality on Mobile Devices	8
2.7 - Using Geographical Location of Objects	9
2.8 - Similar Projects Regarding Augmented Reality	9
2.9 - Similar Projects Regarding Location Based Services	16
Chapter 3	21
MoLand Specification	21
3.1 - Problem Overview.....	21
3.2 - Problem Specification	21
3.3 - Requirements Specification	22
3.4 - Use Cases Model	24
3.5 - Conceptual Data Schema	26
3.6 - Physical System Architecture	27

3.7 - MoLand Modules	28
3.8 - System Communication Overview	29
3.9 - Summary	30
Chapter 4	31
Implementation.....	31
4.1 - Overview	31
4.2 - Implementation Details Overview	32
4.3 - Service Creation.....	33
4.4 - Web Server Implementation Details	39
4.5 - Mobile Application Implementation Details	40
4.6 - Summary	41
Chapter 5	44
Evaluation and Validation.....	44
5.1 - Overview	44
5.2 - Usability Tests	44
5.3 - Interface	49
5.4 - Summary	55
Chapter 6	56
Conclusions and Future Work	56
6.1 - Fulfilling the objectives	56
6.2 - Future Work	57
Appendix A	60
Appendix B	62

List of Figures

Figure 1 - GPS with Geographic Location 23°00'00" South 49°00'00" West (retrieved from [11])	5
Figure 2 - Google MyLocation Structure retrieved from (3)	5
Figure 3 - Example of Augmented Reality (retrieved from (8))	9
Figure 5 - Layer Casa Sapo (retrieved from (13))	10
Figure 4 - Layer Brightkite (retrieved from (38))	10
Figure 6 - Wikitude mobile application retrieved from (14).....	11
Figure 7 - junaio mobile application retrieved from (15)	11
Figure 8 - Tagwhat Mobile application (retrieved from (16))	12
Figure 9 - World Surfer Mobile Application (retrieved from (17))	13
Figure 10 - IBM Seer Mobile Application retrieved from (18)	14
Figure 11 - Google Maps application running on Android, RIM and Windows Mobile (retrieved from (19))	16
Figure 12 - Gowalla Mobile Application (retrieved from (20)).....	17
Figure 13 - Foursquare mobile application (retrieved from (21)).....	18
Figure 14 - Yelp mobile application (retrieved from (23)).....	19
Figure 15 - Buuuk mobile application (retrieved from (24))	20
Figure 16 - Mobile User Use Cases	24
Figure 17 - Web User Use Cases	25
Figure 18 - System Class Diagram	26
Figure 19 - System Architecture	28

Figure 20 - Sequence Diagram of the System Communication	30
Figure 21 - Layout Editor Paper Prototype	34
Figure 22 - Example of services that users would create	35
Figure 23 - Final version of Services Editor.....	36
Figure 24 - User dragging a button to editor.....	37
Figure 25 - User finishing the drag of a button inside the editor	37
Figure 26 - User resizing a component	38
Figure 27 - User editing the text of a button	38
Figure 28 - User setting or changing the image of a Image component.....	39
Figure 29 - Landmarks disposed on the Mobile Application	41
Figure 30- MoLand welcome page interface	50
Figure 31 - MoLand friends interface	50
Figure 32 - MoLand landmarks interface.....	51
Figure 33 - MoLand services interface	52
Figure 34 – Interface where user is creating a new landmark.....	52
Figure 35 – Interface with landmark info saved and with a preview of the service initially associated	53
Figure 36 – Interface where user is creating a service for a Portuguese restaurant ...	54
Figure 37 - Mobile interface with the Landmarks around.....	54
Figure 38 - Service rendered on mobile device	55

List of Tables

Table 1 - Summary of functionalities found in current AR applications.	15
Table 2 - Times and errors for Task 1.....	63
Table 3 - Notes taken for Task 1	64
Table 4 - Times and errors for Task 2.....	65
Table 5 - Notes taken for Task 2	66
Table 6 - Times and errors for Task 3.....	67
Table 7 - Notes taken for Task 3	68
Table 8 - Times and errors for Task 4.....	69
Table 9 - Notes taken for Task 4	70

Abbreviations

3G	Third Generation
AR	Augmented Reality
A-GPS	Assisted Global Positioning System
Cell ID	Cell Tower Identification
DHTML	Dynamic HyperText Markup Language
GIS	Geographic Information System
GPS	Global Positioning System
JPEG	Joint Photographic Experts Group
JSON	Javascript Object Notation
KM	Kilometer
LBS	Location Based Services
PNG	Portable Network Graphics
SMS	Short Message Service
SNIPPET	A Small Piece of Something
URL	Uniform Resource Locator
WEB	World Wide Web
WIFI	Wireless Fidelity
WLAN	Wireless Local Area Network
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 - Motivation

In recent years, the technologies available for mobile devices are getting better constantly and today people can find mobile devices that resemble real computers. Due to the habit of using computers, people are getting faster doing tasks related to computers and technology. However, there are certain types of tasks made by users on mobile devices that can't be accomplished fast due to the small size and some restricted capabilities of those devices. Solutions that attempt to overcome these problems can introduce new forms of interaction with mobile devices and can even introduce new paradigms at the level of visualization and interaction. The motivation for this work relies on the opportunity to create new ways of interacting with mobile devices and new ways of sharing knowledge between people.

1.2 - Problem Description

The problem that this project tries to solve is the creation of a platform and a framework that is able to store locations and the objects that are present on those locations. Using a mobile device with an augmented reality application, the user can interact with an object and check which services it provides, even if the user is away from the object. Objects are basically landmarks. But these landmarks can represent everything that exists in the real world, events, spaces, etc. Associated with these objects, there could be services that they provide. These services can be provided by third-parties, such as big companies or even a single person. For example, one transport company can add each bus stop to the system and when a passenger on his mobile device selects a specific bus stop, he can see the arrival time of next bus. That is only one of the services that this landmark can provide, because it can provide any number of services that the company wants. This service is entirely provided by the company, and could be discovered by users of the augmented reality application available in their phone. The landmarks can also be created by anyone. The user can also configure the landmark services that it provides. For example, the owner of a bar can introduce the landmark of his bar and associate a service to it allowing people to see the DJ that will be performing each day of the week.

In addition, this work is about the creation of a community, where each of the members creates, edits and shares their landmarks and the services that each landmark provides. With

this information created by the community, it will be possible to index almost everything that exists around us. People will be able to search everything in the world and even "see" through buildings! The system will be created and filled with static information iteratively. This static information can be provided by systems that are actually indexing some of the locations of popular monuments, places or cities. Some information systems were used like Wikipedia and Wikimapia. With this static and valuable information the users feel more enthusiastic when using the system for the first time, when there are few locations indexed in the system. There are many other use-cases that can help the users of the system. For example, imagine that one user wants that all his/her friends are able to know the location of his/her birthday party. To accomplish this task, the user only needs to create a new landmark on the system with the appropriate configurations and then share it with his/her friends. That type of use cases will represent the main difference to the users. And that is also the main difference from existent applications in the market, because on the existent applications the user can only search and see the landmarks and not create landmarks.

1.3 - Objectives

The main objective of this work is the creation of a framework that allows the creation of services based on their location. These services are referenced by landmarks that the user can also create in the framework. There is a lot of information around us that we cannot see through our simple, yet powerful eyes. Using technology, people can find information, some of which hidden, and make it available to other people. The second objective is the creation of a platform where users can edit the services provided. The third objective is the creation of a library for mobile devices that renders the services on mobile devices. Finally, the fourth objective is the creation of an augmented reality application for mobile devices. That application shows through the camera the existing landmarks and allows the user to see the service provided by landmark.

1.4 - Document Structure

This dissertation is divided in six chapters.

In this first chapter a description of the problem's context and an explanation of the main goals of this project is presented. The second chapter presents the state of the art, where all the technologies involved and some similar projects are analyzed. The third chapter defines a solution for a system that provides location-based services for everyone. This chapter has the standard content that is usually presented in a software engineering project. The fourth chapter presents the implementation details and the fifth chapter presents the evaluation and validation of the system proposed in the fourth chapter.

Finally, in the sixth chapter a summary of the developed work and a description of some directions of future work are presented.

Chapter 2

State of the Art

2.1 - Introduction

The main purpose of this chapter is to present the existing technologies related to the subject of this work. Beyond the technologies, the chapter has also some references and descriptions of some similar projects. The following systems/technologies are described:

- Geographic Information Systems
- Geographical Location on Mobile Devices
- Location Based Services
- Augmented Reality
- Mobile Devices

2.2 - Geographic Information Systems

A Geographic Information System (GIS) is a system that allows to store, manage and present information related to a location. This information, typically stored in a geo-database, can present many layers of information that a map is made of. With all the information present in layers, the GIS is able to respond to interactive queries made by users. In the simplest way, GIS is the merging of cartography with database technology. Geographic Information Systems are used by many types of organizations, such as Emergency Services, Environmental, Medical, Industries, Businesses, Defense, Natural Disasters, Education and Government.

GIS has normally three associated views:

- Database View
- Map View
- Model View

2.2.1 - The Database View

The database view shows that GIS works as the database of the world. The whole world is saved in a database. It is an Information System for Geography. The system is based on a structured database that stores and describes the world in geographical terms.

The geo-database characteristics are:

- Storing a rich collection of spatial data in a centralized location
- Applying sophisticated rules and relationships to the data
- Defining advanced geometric relational models
- Working in an environment with sophisticated aspects provided by databases(e.g., multi-user access and editing, integration, interoperability, scalability, customization, distribution)

2.2.2 - The Map View

In addition, the map view shows that GIS is a set of maps with interesting capabilities to show features on the Earth's surface. The GIS stores maps with the basic geographic information to support the queries, analysis and edition of the information.

2.2.3 - The Model View

The model view shows that GIS acts as an integrator of data and relates any data with a spatial component that does not depend on the source of the data.

For example, one can combine the location of mobile workers, located in real time by GPS devices, in relation to customers' homes, located by address and derived from the customer database. GIS maps this data, giving dispatchers a visual tool to plan the best routes for mobile staff. This allows people to save time and money (1).

2.3 - Geographical Location on Mobile Devices

2.3.1 - Global Positioning System

Nowadays, anyone can know what is his/her exact location using a GPS (Global Positioning System). The GPS provides the latitude, longitude and altitude of any place on Earth. This objective is achieved with the help of 24 satellites distributed in 6 orbital planes equally spaced, at an altitude of 20,200 Km approximately. This configuration guarantees that, at least 4 GPS satellites are visible from any point on land surface. These satellites send a signal and, on land, the GPS device receives the signals. The position can be calculated in 3 dimensions (latitude, longitude and altitude), since the GPS on land receives the signal from at least 4 satellites

(2).Today it is common to find a mobile device with a GPS receiver. This allows the developers of mobile applications to do innovative applications that use the GPS to know the location of the user and then interact with him/her.



Figure 1 - GPS with Geographic Location 23°00'00" South 49°00'00" West (retrieved from [11])

2.3.2 - Cell tower ID

Another way that a mobile device is able to find the user's actual location is using the technology called Cell ID. This solution is implemented by the majority of mobile device manufacturers and is supported by the infrastructure of mobile communications providers. It works in the following manner. Any mobile device needs to be connected via one cell tower. The phone knows the ID of the cell that it is using. If there is a system that knows the localization of the cell tower, the phone can use this system using the 3G/WIFI network to ask what is the location of its current cell ID. Google has a system that provides this service. The location of cell towers is achieved when a mobile device, that has GPS, sends to the Google service the current Cell ID and the GPS' accurate position. After the user communicates with this service and sends the cell ID and the location of cell ID, Google knows with a controlled accuracy, the cell tower's location. So, when a mobile without GPS queries the service, Google translates the cell ID provided into a geographic location (3). The following picture shows the stages and the flow of data related with the two use cases mentioned before.

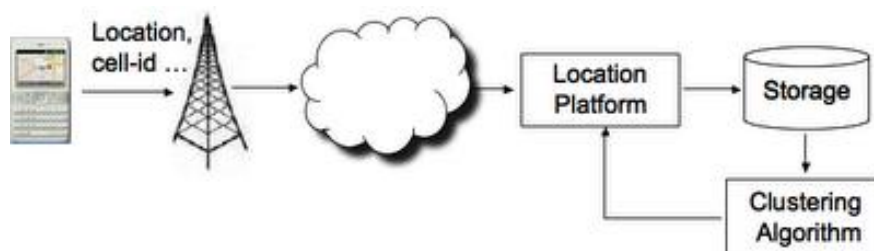


Figure 2 - Google MyLocation Structure retrieved from (3)

2.3.3 - Location Based Services

Location Based Services (LBSs) are business and consumer services that give users a set of services based on the geographic locations of the client. These services offer the possibility to users or machines of finding or locating other people, machines, vehicles, resources and also location-sensitive services. The request for location can be originated from the client himself or from another entity such as an application provider or the network.

Normally, the LBSs include two actions:

- Obtain the location of a user
- Use this information to provide a Service

LBSs can automatically be triggered when the client is at a specific location. Alternatively they can originate from the client himself, in order to provide location-based requests such as information needs, finding points of interest, finding persons or even checking the traffic conditions.

2.3.4 - The relation of GIS and LBS

GIS and LBS have some particular similarities. Such common features are the handling of data with positional reference and spatial analysis functions that give answers to questions like:

- Where am I?
- What is near by?
- How can I go to?

But LBS and GIS have different origins and different user groups. Whereas LBSs were born quite recently by the evolution of public mobile services, GISs have been developed during several decades on the basis of professional geographic data applications. With respect to user groups, GISs can be seen as traditional "professional" systems, intended for experienced users with wide collection of functionalities. Furthermore, GISs require extensive computing resources. In contrast, the LBSs are developed as limited services for large non-professional user groups. LBS applications operate with the restrictions of mobile computing environment like low computational power, small displays or battery run time of the mobile device.

2.4 - LBS Components

If the user wants to use a LBS, different infrastructure elements are necessary. The basic components and their connections are the tools for the user to request the needed information (4). The results can be given by speech, using pictures, text and so on. Possible devices are PDAs, Mobile Phones, Laptops, and so on, but the device can also be a navigational unit of a car or a toll box for road pricing, in a truck.

The second component is the mobile network that transfers the user data and service request from the mobile terminal to the service provider and then the requested information back to the user.

For the processing of a service usually the user's position has to be determined. The user position can be obtained either by using the mobile communication network or by using the GPS. Further possibilities to determine the position are WLAN stations, active badges or radio beacons. The latter positioning methods can be especially used for indoor navigation like in a museum. If the position is not determined automatically the user can also specify it manually.

The service provider offers a number of different services to the user and is responsible for the service request processing. Such services offer the calculation of a position, finding a route, searching the yellow pages with respect to position or searching specific information about objects interesting to the user (e.g. a bird in wild life park) and so forth.

Service providers will usually not store and maintain all the information that can be requested by users. Therefore geographic base data and location information data will be usually requested from the maintaining authority (e.g. mapping agencies) or business and industry partners (e.g. yellow pages, traffic companies).

2.4.1 - LBS types

There are three types of Location Based Services:

- Pull
- Push
- Tracking

In the case of a pull service, a client makes himself a request for a LBS. When making this request he gives the permission for his position to be given because, without that location information, the request for the service can not be completed (5).

An example of pull-type service is when a customer sends an SMS to one service number for available local pharmacies. To offer this service, a Service Provider has to know from where the request was made. Based on the customer's location information, the Service Provider will send a reply with the available pharmacy in the area from where the request was made.

Push services differ from Pull services since the request for the service is not technically made by the customer but by the Service Provider. In these cases the customer must give permission to the Service Provider to send information to his mobile phone. An example of a push-type service is when the client has registered himself to a pharmacies service and set up a profile on which he requests details of available pharmacies every morning at 10:00AM. To send that information of the right area to the client, the Service Provider has to know where the customer is at 10:00AM. Because the customer has allowed sharing his position every morning, at 10:00AM with the pharmacies service, the operator will tell the service provider where this customer is at the time of service request. After getting the position of the customer, the Service Provider can send the right information to the client informing the nearest pharmacy that is open.

Tracking services work in the following manner. The rationale behind that type of service is that someone (person or service) asks for a location of the mobile terminal (person, vehicle, etc...). As in the pull and push cases the assumption is that the customer has given the permissions

which allow particular persons or services to track him. An example of a tracking service is a friend finder. A group of friends has signed up for the service to better keep in touch with each other. Every member of the group has given permission to be tracked by the other members. When the group combines a meeting, each one might choose optional routes to get to the meeting. Those who are first at the meeting point would like to know where the others are at that moment. They may set up a connection to the tracking service and make the request of the positions of those members who are not yet there. Many of the services, like those mentioned above, are quite easy to implement when the operator is also a Service Provider and is offering these services to its own clients.

2.5 - Application Examples

According to (4), LBSs are used by Emergency Services, Navigation Services, Information Services, Tracking and Management Services, Billing Services, etc. Relatively to the last topic in (4), the author mention that in the next decade, researchers plan to pull graphics out of the phone or computer display and integrate them into real-world environments. This technology, called augmented reality, will further blur the line between what is real and what is computer-generated by enhancing what we see, hear, feel and smell. Unlike in virtual environments, in augmented reality, the user can see the real world around him, with computer graphics superimposed or composed with the real world. Instead of replacing the real world it is supplemented. So called "see-through" devices, usually worn on the head, overlay graphics and text on the user's view of his or her surroundings.

2.6 - Augmented Reality on Mobile Devices

Augmented Reality (AR) is essentially the fusion of actual and virtual reality, where graphic objects are blended into real footage in real time. AR creates the illusion that virtual, computer-generated objects exist in the real world (6). In industry, however, AR has been in use for quite some time. As consumers and individuals, we are only beginning to see the capabilities of AR in our day-to-day use. In the military field for example, AR has been used for a number of years in the windscreens or headgears of soldiers that provide information about their surroundings. Military strategists have been using navigation systems to position friendly and enemy units, and often AR allows the soldier to spot more details than the human eye is capable of. In medicine, AR has been used in several areas including surgery simulations, ultra-sound projections, visualization of anatomical joints in motion, surgical education and healthcare (7). Gamers have been developing AR games to push the boundaries of user's imagination. In Figure 3 it is possible to visualize an example where real time information was added to a scene captured from a camera. The University and the Newsstand landmarks are located in the image in the correct place.



Figure 3 - Example of Augmented Reality (retrieved from (8))

2.7 - Using Geographical Location of Objects

Nowadays, more and more applications of augmented reality for mobile devices begin to appear. There are several applications that use the mobile device's camera to capture the real world and then add virtual information. Those types of applications are fun and useful for users (9). These applications use the compass, camera, and GPS system of the mobile device. The result is that the user can see things like the location of Twitter users and local restaurants in the physical world, even if they are kms away.

There are other types of applications that run on mobile devices using AR tags to identify objects in the real world. This type of applications are commonly developed using a toolkit called ARToolKit that was ported to mobile operating systems (10). For example for Android there is the NyARToolkit (11) that is a port from ARToolKit (available for the C language) to Java language.

2.8 - Similar Projects Regarding Augmented Reality

2.8.1 - Layar

The Layar Reality Browser shows what is around the user by displaying real time digital information on top of the real world as seen through the camera of the mobile phone. Layar augments the real world as seen through user mobile phone, based on its location. Layar works by using a combination of the mobile phone's camera, compass and GPS data to identify the

user's location and field of view, retrieve data based on those geographical coordinates, and overlay that data over the camera view (12).

Users can choose a filter or “layer” to overlay on top of their real-world view — Wikipedia entries, bars and nightlife nearby, Twitter users nearby, etc. — and point the camera at its surroundings. The user gets a visual overlay of relevant results close to him/her, determined by a range filter. New types of layers, featuring different types of data, are launching all the time thanks to a strong third-party developer platform. The application helps users find and filter the content by collecting featured and popular layers, as well as providing a search mechanism to find new layers.



Figure 5 - Layer Casa Sapo (retrieved from (13))



Figure 4 - Layer Brightkite (retrieved from (38))

2.8.2 - Wikitude

Wikitude is a mobile application that uses an augmented reality platform. Usually, Wikitude is seen as a location-aware Wikipedia application for mobile devices. Current applications of Wikitude, such as Wikitude World Browser and Wikitude Drive, run only on mobile devices. These applications can only be used for iPhone, Android, and Symbian software platforms as travel guides and personal navigation devices. Future applications of Wikitude can be developed for military, city modeling, and shopping (14).

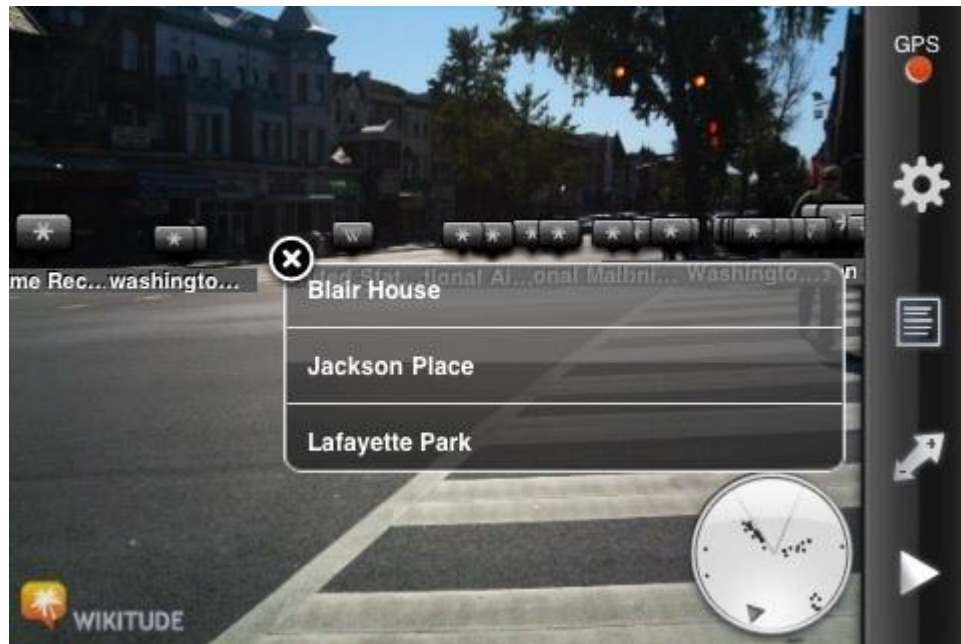


Figure 6 - Wikitude mobile application retrieved from (14)

2.8.3 - junaio

junaio aims to be the next generation mobile augmented reality browser, designed for camera equipped devices. It provides users with interactive web based information and services wherever they are. Users can point the camera, click and view the history of a monument, the nearest affordable hotel, the menu at the restaurant next door, what movies are showing, or even the last train home. Inserting graphics and 3D objects to user's location or posting recommendations makes junaio a social experience.



Figure 7 - junaio mobile application retrieved from (15)

junaio has extended its capabilities beyond the usual location-based internet services. Not only may the user obtain information on nearby POIs such as shops, restaurants or train stations, but also the camera's eye is now able to identify objects and "glue" object specific real-time, dynamic, social and 3D information onto the object itself (15).

2.8.4 - Tagwhat

Tagwhat is a free network where the user can create and share location based messages and content on-line or in mobile augmented reality.

Tagwhat aims to make augmented reality technology more useful and interactive by allowing users to place digital tags on anything of interest that other users can then see and comment on. Tagwhat's strong suit is its interactivity, along with the fact that users have the power to ask about locations, tag places, and even upload their own photos and associate them with a particular place for other users to stumble onto. Instead of using a passive database like other augmented reality apps, Tagwhat allows users to participate and build a rich directory of destinations and items of note in their area. (16)



Figure 8 - Tagwhat Mobile application (retrieved from (16))

2.8.5 - World Surfer

World Surfer allows users to point their phones in a particular direction to search for retailers, restaurants and other points of interest. Using World Surfer users can “click” on the real world with their phones to access information. World Surfer is a directionally-filtered location browser which helps users to zero in on places in the real world which interest them, simply and efficiently. After locating a point of interest, World Surfer users can visit the location's website, get reviews, place a phone call, check for pictures or YouTube videos, search Google, see the location on a map, and be guided there by an interactive arrow. They can also bookmark their favorite spots, hotel or parking garage to return after some exploring days (17).



Figure 9 - World Surfer Mobile Application (retrieved from (17))

2.8.6 - IBM Seer

Seer is an Augmented Reality App for Australian Open 2010. Seer users can see points of interest throughout the tennis grounds which have been plotted using GPS. By making use of the T-Mobile G1's digital compass and precise GPS coordinates, the application offers a 'heads up display' to show the users what they are looking at. It augments this with other live data from the scoring systems and IBM scouts reporting from around the grounds, to give the user a comprehensive and dynamic insight into their surroundings. For instance, pointing the camera lens towards a court will not only identify the court number, but also display details about the current and subsequent matches.

Seer Android users can also use the phones' Map view, which pinpoints their location on a detailed map of the grounds, and can be used as a way finder. The Timeline view is an aggregation of news feeds and updates from IBM scouts, and allows users to see in real-time what is happening around the site. A handy 'Radar' function indicates the user's current position and nearby points of interest within range (18).



Figure 10 - IBM Seer Mobile Application retrieved from (18)

2.8.7 - Comparison

Among all the available applications, the Layar application is the application that has more functionalities. The application that has less functionalities implemented is the IBM Seer Application, but this application was developed for a specific event. It should also be highlighted that the user can only create landmarks with Tagwhat and World Surfer. All the functionalities are summarized in Table 1.

Functionalities	Applications					
	Layar	Wikitude	junaio	Tagwhat	World Surfer	IBM Seer
Has layers/worlds/channels	Yes	Yes	Yes	Yes	Yes	No
Possibility of changing range	Yes	Yes	No	Yes	No	No
Has search	Yes	Yes	No	Yes	No	No
Has layer/world/channel categories	Yes	No	Yes	Yes	No	No
Users can add layers/worlds/channels	No	No	No	No	No	No
Developers can add layers/worlds/channels	Yes	Yes	No	No	No	No
Has List visualization	Yes	Yes	No	Yes	Yes	Yes
Has map visualization	Yes	Yes	No	Yes	Yes	Yes
Has paid layers	Yes	No	No	No	No	No
Has favorites/bookmarks	Yes	Yes	Yes	No	Yes	No
The information of landmark is viewed on	Browser	Browser	Browser	Application and Browser	Browser	Application
Augmented Reality visualization has a radar	Yes	Yes	Yes	Yes	No	Yes
Users can add landmarks	No	No	No	Yes	Yes	No

Table 1 - Summary of functionalities found in current AR applications.

2.9 - Similar Projects Regarding Location Based Services

2.9.1 - Google Maps

Google Maps is a mobile application that is available on most platforms (19). With the Google Maps application, users are able to determine their current location, get driving transit directions and get phone numbers and addresses for local businesses. Besides these main functionalities, the application has others such as:

- Layers – users can see layers of geographic information (for example Wikipedia, Transit).
- Latitude – users can see their friends' location and status messages.
- Buzz – users can see what is going on nearby, post a message tagged with their location and users can also follow their friends.



Figure 11 - Google Maps application running on Android, RIM and Windows Mobile (retrieved from (19))

2.9.2 - Gowalla

Gowalla is a mobile web application that allows users to check-in to locations that they visit using their mobile device. This is achieved either through the use of dedicated applications available on Google Android, iPhone, Palm WebOS and BlackBerry, or via m.gowalla.com using the mobile browser.

Spots and Trips can be featured by Gowalla. Featured spots tend to be local landmarks such as Buckingham Palace in London, whilst featured trips are chosen for being unique and exciting.

Upon checking-in, users may receive items as a bonus and these can be swapped or dropped at spots. Users become founders of a spot by dropping an item there. Items form a key feature

within the game and each user has a vault into which he/she can place items he/she wants to keep for ever (20).



Figure 12 - Gowalla Mobile Application (retrieved from (20))

2.9.3 - Foursquare

Foursquare is a web and mobile application that allows users to connect with friends and update their location. Points are awarded for "checking in" at venues. Users can choose to have their Twitter and/or their Facebook accounts updated when they check in. Users can also earn badges by checking in at locations with certain tags, for check-in frequency or for other patterns such as time of check-in. Users will be able to add their own custom badges to the site in the future. If a user has checked-in to a venue more than anyone else, on separate days, and he/she have a profile photo, he/she will be crowned "Mayor" of that venue, until someone else earns the title by checking in more times than the previous mayor. Users can create a "To Do" list for their private use and add "Tips" to venues that other users can read, which serve as suggestions for great things to do, see or eat at the location.

Foursquare currently has iPhone, Android, webOS, Windows Phone 7 and BlackBerry applications (21).

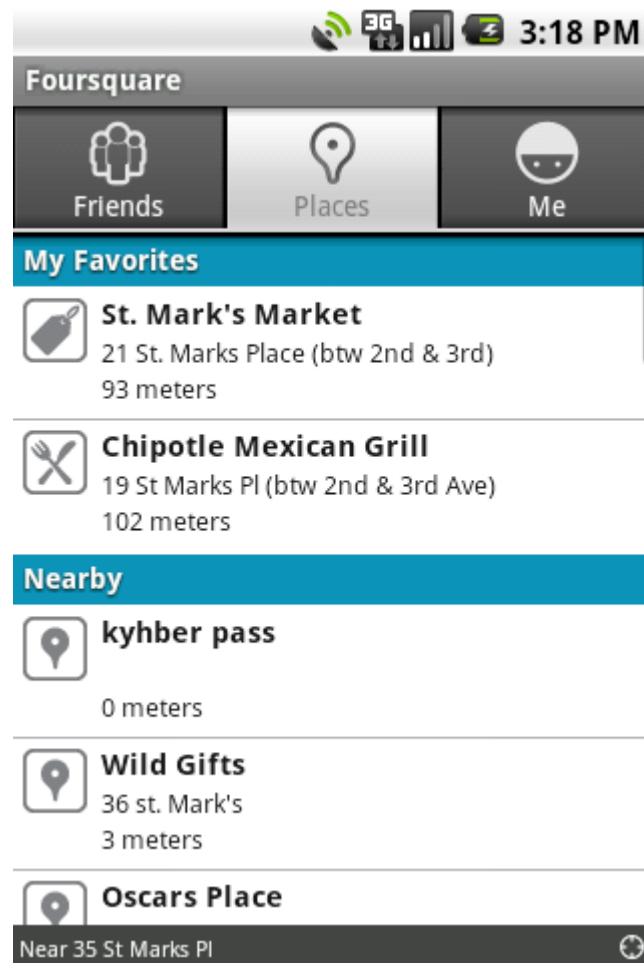


Figure 13 - Foursquare mobile application (retrieved from (21))

2.9.4 - Yelp

Yelp provides online local search capabilities for its visitors. A typical search includes what the user is seeking (e.g. a barber shop) and the location from which the search is to be performed, entered as a specific address, neighborhood, city/state combination, or zip code. Each business listing result contains a 5-point rating, reviews from other site visitors, and details such as the business address, hours, accessibility, and parking. Site visitors can aid in keeping the business listings up to date, with moderator approval, and business owners can directly update their own business' listing information (22). Listings and related content are organized by city and a multi-tier categorization system. Content and listings can also be discovered through categorized reviews or via Yelp member profiles and their review lists. Maps leveraging Google Maps show reviewed businesses to further aid in business discovery. Information can be accessed by web or mobile browsers.

Yelp is also available on Mobile devices. **Yelp mobile application** determines the user's current location (either using built-in GPS or a Wi-Fi-based location) and then lets user search for nearby businesses, read reviews of establishments and access a moveable Google map that can help user easily redefine its search based on locale (23).

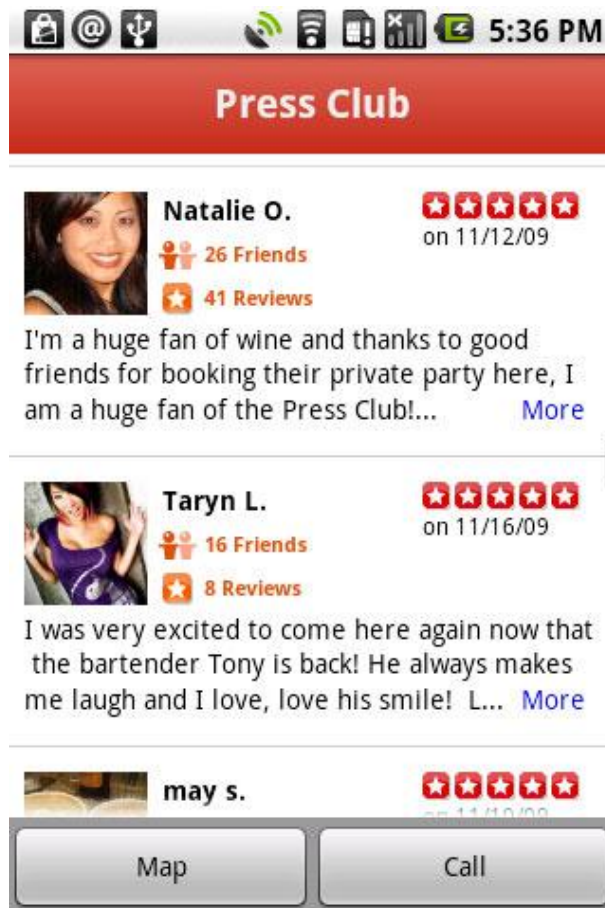


Figure 14 - Yelp mobile application (retrieved from (23))

2.9.5 - Buuuk

Buuuk provides location based restaurant information to the consumers on Android and iPhone. The Buuuk application is able to search for restaurants where they are located (using their GPS location aware feature), check out the offers available for Buuuk users (or credit card offers), check top rated restaurants and pubs, submit reviews, etc. In addition, users can also see the restaurants on the map, get direction to the restaurants, calculate distance as well as time it takes if users take a car, train or bus to the restaurant (24).

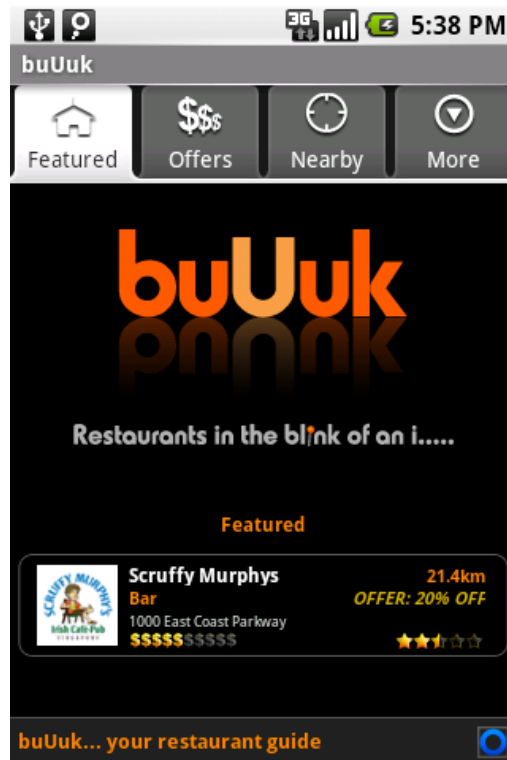


Figure 15 - Buuuk mobile application (retrieved from (24))

Chapter 3

MoLand Specification

3.1 - Problem Overview

As an introduction to the description of the problem, it is necessary to describe the context of location-based services. Location based services are services that, as its name suggests, are associated with a location. In fact, everything that exists or happens in our planet has an associated location. There is therefore an infinite number of use cases that can be implemented by taking advantage of this condition: services have a location.

Nowadays there are many systems that allow their users to enjoy services indexed by the user's location. However, these services are normally created and are made available by companies or developers who have an above average technical knowledge. Thereby, common people who have services to provide can not do it because they do not have the required knowledge to do that task.

This restriction is one of the main problems that this project is solving.

3.2 - Problem Specification

Beyond its major theme – location-based services –, the project also focuses on other topics such as interface creation, and system usability.

The main objective is: the creation of a platform that allows any user to manage their landmarks and services associated with them. Taking into account the accessibility required for the use cases for managing the services, it is necessary to create an accessible interface that allows any user to create and manage the services that he/she wants to provide or make available to other users.

In a high-level perspective, this project takes information that has associated a geographical context and offers it to the user, in a simple, effective and fun way. This problem has three main

entities that need special attention: Users, Data and Context. Users are the final entity that will consume the Data and the Context is shared by the User and the Data.

3.3 - Requirements Specification

In this section all the requirements are enumerated and described.

3.3.1 - Functional Requirements

3.3.1.1 - Web Platform

Users Management

- Users can register and create an account
- Users have a profile and can edit it
- Users have friends that can be added and removed

Landmarks Management

- Users can add landmarks to the system
- The landmark can be edited or removed
- All the landmarks have a latitude and longitude
- When adding a landmark, the user can search by a place to know the latitude and longitude

Layers Management

- Users can create layers in order to filter the landmarks
- Users can add/remove layers from landmarks

Services Management

- Users can add one service for each landmark
- All the content available on service can be added/edited/removed by users
- Landmarks have a preview of the service that shows to the user what will be seen on a mobile device
- The services editor has content templates that users can add
- The services editor has basic components such as buttons, textboxes, images, etc. that user can add and then edit the content
- The basic components have properties such as colors, sizes, etc. that can be configured by users
- The user can upload an image to fill one image component added to layout
- The services editor should be a WYSIWYG editor

3.3.1.2 - Mobile Application

Landmarks View

- The landmarks must be visible in an augmented reality application when the user points the camera of the mobile device in the direction of landmark
- The landmarks that appear on mobile application are within a radius of interest
- The mobile application user can increase or decrease the radius
- When the user selects the landmark, the landmark service appears
- The user can choose the visible layers in order to filter the landmarks that are shown
- The user can interact with the provided service
- Simple use cases are associated with landmarks such as “take me there”, “call landmark owner”, “rate landmark”, “mark as spam”, etc., when the landmark has data related with this use cases.

3.3.2 - Non-Functional Requirements

3.3.2.1 - Usability

The system should follow the usability standards so that the users can use it easily. The web application should have a clean interface and must also be compatible with most browsers. The mobile application should use the standard graphical components available on the mobile device operating system. Thereby, the user can quickly remember the action associated with the component.

3.3.2.2 - Reliability

As the system relies on the web, it is important that the system information is reliable. The users have to identify themselves with the system and not create a link of distrust. Private user data must be confidential in order to establish a high degree of trust.

3.3.2.3 - Performance

The mobile application must be fast in all the implemented use cases. The rendering of services and landmarks must be especially fast; when the user moves the mobile device, the landmarks should move smoothly on screen device.

3.4 - Use Cases Model

The use case model describes the proposed system functionalities. A use case represents an interaction between a user and the system. The two environments were divided into two different diagrams.

3.4.1 - Mobile User

Figure 16 includes the use cases related to the mobile user.



Figure 16 - Mobile User Use Cases

3.4.2 - Desktop User

Figure 17 has the use for a web user.

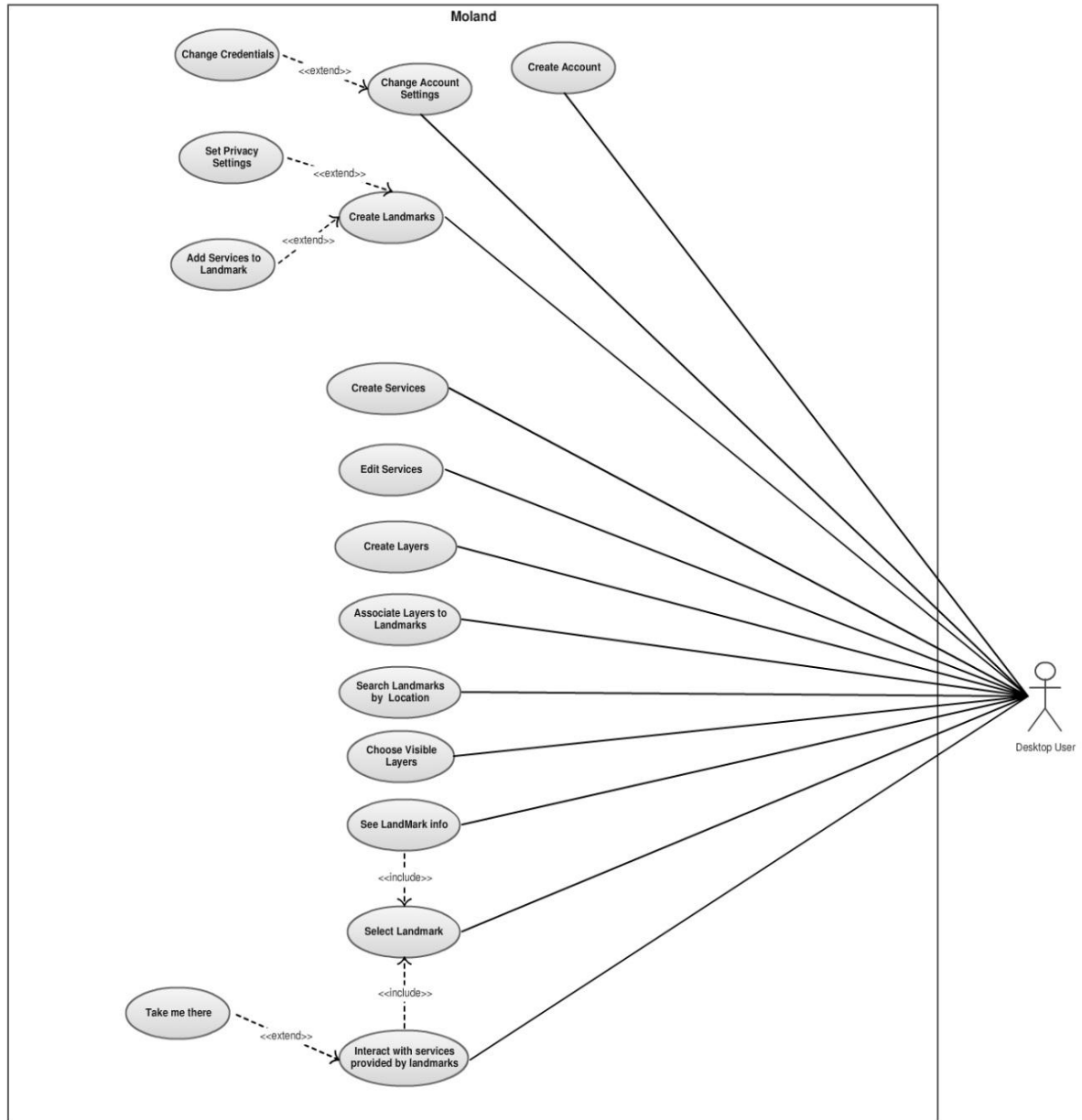


Figure 17 - Web User Use Cases

3.4.2.1 - Actors

The actors of the system are:

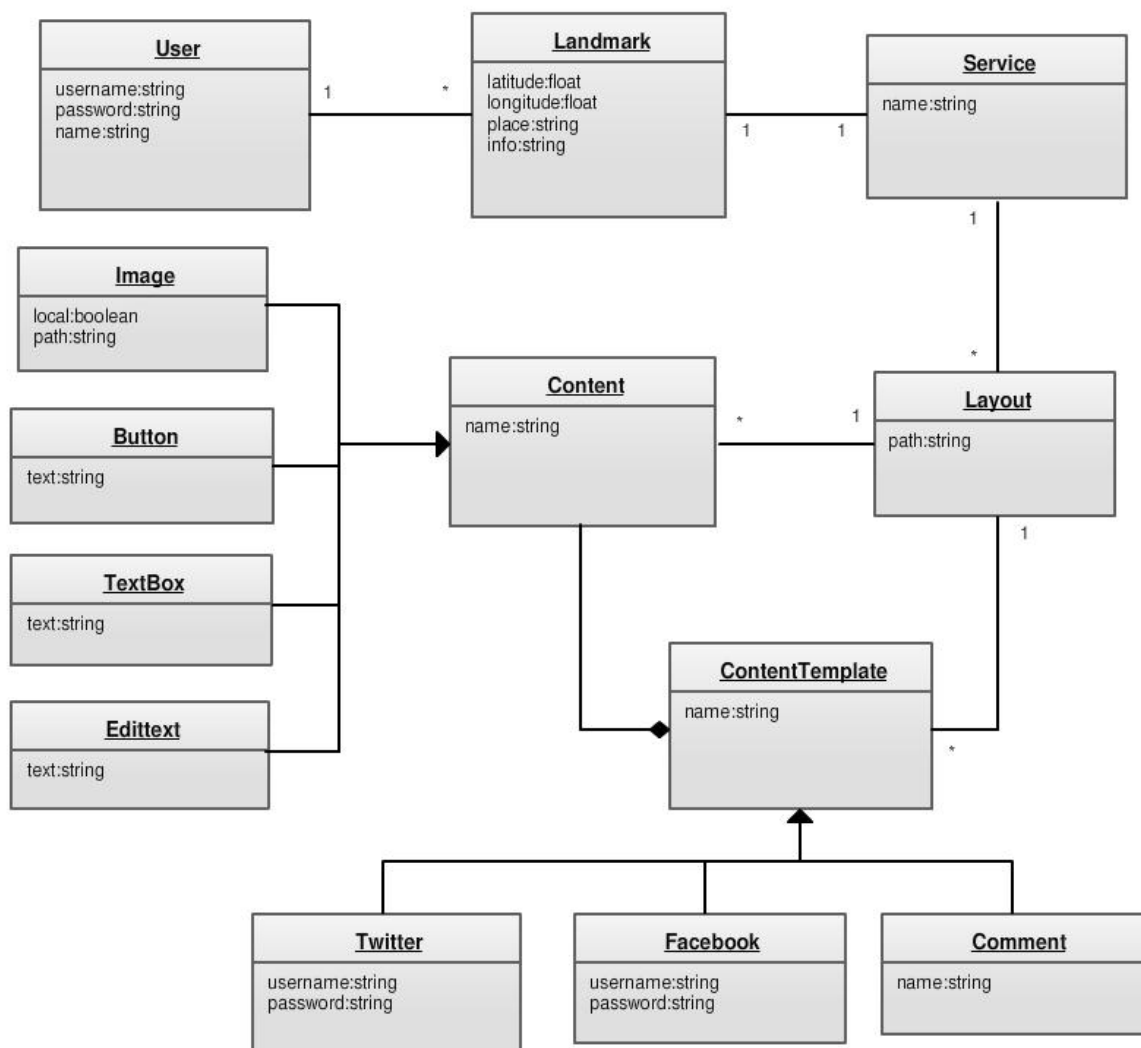
- Mobile User
- Web User

All users of the system have the same privileges.

The user has the Mobile role when he/she is using the system through the mobile application. The role changes when user uses the system through system available on web.

3.5 - Conceptual Data Schema

The conceptual data schema presents a high level view of the system. This schema reflects the organization of entities and their relationships as well as the integrity constraints that system has - Figure 18 illustrates this.



3.5.1 - Class Description

User – The class User represents the users of the system. The user has many landmarks and has at least the attributes username and password for the system authentication. The real name is also associated with the user in order to be easily recognized by friends on system.

Landmark – The Landmark is one point in the World that has one specific latitude and longitude. The Landmark has also the place, corresponding to the landmark address more commonly used by people and also additional information. One Landmark belongs to the User and has one Service.

Service – The Service is provided by one Landmark. A Service represents the information that will be available in one Landmark. The Service has a name and a Layout.

Layout – The class Layout contains the information that is provided by the Service. The Layout is composed by content and has one attribute to store the path where the Layout is saved. One Layout may have many Content elements and may also have many ContentTemplate elements.

Content – The class Content represents the components that are within a Layout. The Content can be one of the following widgets: Image, Button, Textbox, Edittext. One Content belongs to one Layout.

ContentTemplate – One ContentTemplate object is composed by any of the widgets referred before. The ContentTemplates available are: Twitter, Facebook and Comment. The class ContentTemplate has the name of the Template and the other sub-classes have the necessary configurations.

3.6 - Physical System Architecture

The architecture of the system is client-server. The server is where all the information is stored and, on the client side, all the mobile phones connect to the server and make requests to server.

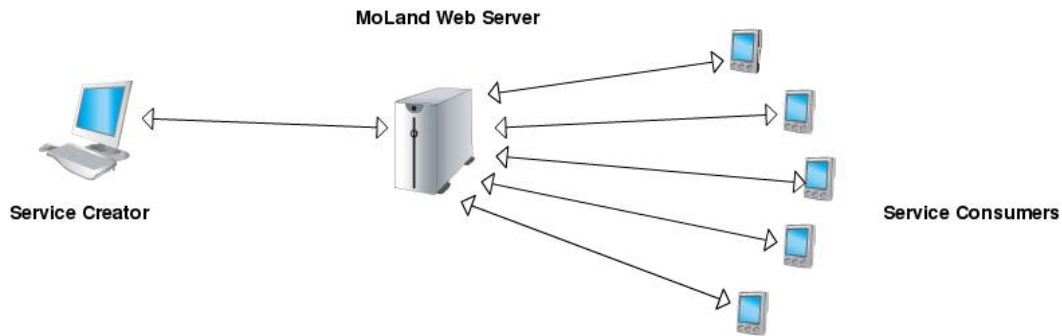


Figure 19 - System Architecture

3.7 - MoLand Modules

The system has different modules on the Web Server and on the Mobile Application. The Web Server has modules mostly related to management and the Mobile Application has modules mostly related to visualization.

3.7.1 - Web Server Modules

The Server has three modules. One module is responsible for attending the requests made by users when they are using the system on the Web Browser. The second module is the management of requests made by the mobile application. The last module is the module responsible for making the service request management.

Web Requests Management – This module implements all the system business logic. It handles the interactions made by users with the system using the Browser.

Basically this module has all the create/update/delete methods for all the classes related with system.

Landmarks Requests Management – This module is responsible for attending the requests made by the mobile application. The mobile application is always asking server for landmarks around the mobile device location. This module handles these requests and responds sending to mobile device the landmarks available on system and near the mobile device.

Services Requests Management – This module is responsible for handling the requests made by the mobile application when the user wants to see the service for one specific landmark. As each landmark has one service, the user can see for each landmark the service provided. But the service is only sent to the mobile device when the user wants to see the service, because this request is time consuming and can decrease the application performance. When user selects the landmark, the service for this landmark is requested to the server rather than all the services being downloaded for all landmarks at start up.

3.7.2 - Mobile Modules

The mobile device has three modules. Two modules are related to the visualization of content on the mobile application, and the other is related to the network requests. The last is basically responsible for downloading all the necessary information from the network not only from the server but also, for example, from the Internet when one service has an image that is available online.

Landmarks Visualization - This module is responsible for managing the position of landmarks on the device screen. As the user can move the mobile device around her, the landmarks need to be positioned on the right position on the screen.

Network Requests – This module is very important because the mobile application without network access is a simple camera application. Without the network the mobile application can not know the landmarks available around the mobile device and can not know also the services available for landmarks. So, this module is therefore responsible for creating the requests and returning the expected content. The content can be one of the following: Images (PNG, JPEG), JSON or XML.

Service Visualization – The service visualization module is responsible for handling the response of the service request made by the Network Requests module. Then this module is responsible for creating all the necessary interface components for rendering the service and the creation of all components that compound the service.

3.8 - System Communication Overview

The system communication is done between the Web Server and the Mobile Application. First, the users can create/edit/remove landmarks using the Web server. Then, on the mobile application, other users and also the landmark owner can see the landmark on the mobile application. These landmarks are only sent to the mobile application after the user is logged in to the system. Then the user can interact with any of the visible landmarks. When the user selects one landmark, the service that this landmark provides is automatically requested to the web server. This flow is depicted in Figure 20.

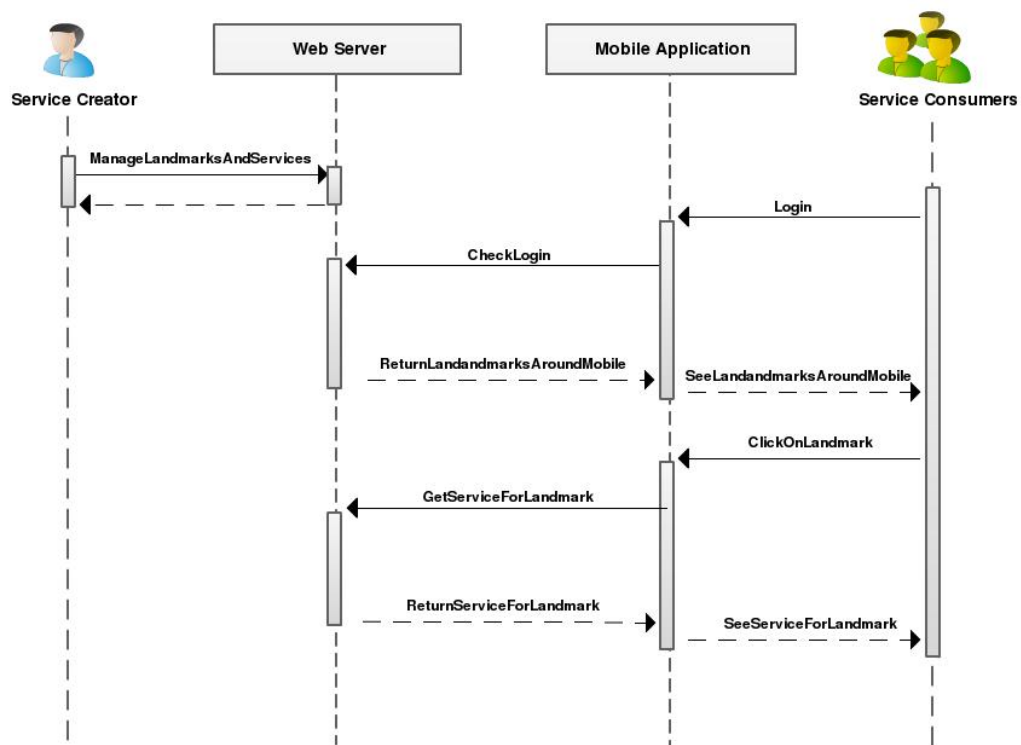


Figure 20 - Sequence Diagram of the System Communication

3.9 - Summary

One of the major goals of this system is to create an online community where everybody can create landmarks and associate services to these landmarks. The system architecture has two environments: the Web Server and the Mobile Application. The most important use cases are the creation of landmarks and services that are stored on the Web Server, and the visualization of landmarks and services on the Mobile Application.

Chapter 4

Implementation

4.1 - Overview

The developed work consists of a system, available to the end user through the Web Interface and through a Mobile Application that can be installed on any device running the Android Operating System(OS).

The Android OS provides the necessary framework and tools to create the Mobile Application and implement all the use cases mentioned before. The Android was chosen because: it is an open source platform, it does not distinguish between “native” applications and third-party applications, it allows the complete integration between applications of various nature, and finally the most important feature, it is a platform that simplifies the process of

developing new applications. The language used to create new mobile applications for Android is Java. This language is one of the most well known among developers.

The *Ruby on Rails* framework provides the necessary methods and tools to implement the Web Server. There are some reasons that were considered to choose this framework: it is intended to be used with an agile development methodology, it is open source, it uses the Model View Controller architecture pattern and offers HTML, XML and JSON as output formats which is especially suitable for the RESTful web services desired in the system.

The *prototype & script.aculo.us* was the javascript framework chosen to implement the necessary functionalities essentially, related with service creation and edition. This framework was chosen because there is already a significant number of plugins that can be reused and easily implemented.

4.2 - Implementation Details Overview

4.2.1 - Web Server

On the server side, the system is composed by one Apache web server that is responsible for attending the HTTP requests made by the Mobile Application. This web server relies on a database where all the information is stored. The database system management responsible for data storage is the well-known MySQL DBMS. The framework that is in charge of making the connection between the database and the web server is Ruby on Rails. This framework is responsible for implementing all the business logic that is necessary to get the system functioning properly. The contents returned by the web server are small snippets of XML and JSON providing all the information necessary to the mobile device application.

JSON is a lightweight computer data interchange format. It is a text-based, human-readable format for representing simple data structures and associative arrays called "objects".

Beyond the support to mobile applications, the server needs to provide support to a web application where users can create, edit, manage and share their landmarks. This is supported by the technologies mentioned above, but the response of the web server is, at this time, DHTML because the recipient is the user's browser. This result is visualized on any web browser. On the server side these are all the technologies that support the entire system.

4.2.2 - Mobile Application

The mobile application was developed to work on devices running the Android operating system. The minimum requirements that this device needs are: one camera and access to the network (WIFI or 3G). The GPS will only be a mandatory requirement if the user wants to have high accuracy in the information provided by the application. The camera is used to capture the video from the surrounding environment that will be used for creating an augmented reality environment; the network is necessary to request to the server the landmarks that surround the user. This request will contain the location of the mobile phone so that the server answer contains all the landmarks actually near the user and available on database.

4.3 - Service Creation

The Service Creation is responsible for allowing users to create services. As mentioned before, one service is composed of basic components such as Images, Text Boxes, Edit Text or Buttons. With these basic components, users can create one service with the information they want. The service creation was implemented with Javascript, therefore the processing of this use case happens on the client side. All graphical web browsers implement Javascript and since this functionality is implemented only with Javascript, there is no need to install other tools or plugins.

4.3.1 - Initial Ideas

Initially the main purpose was to make a layout editor but it was extended to a service editor. This service editor allows users to create the service as they want. The editor needs to be a What You See Is What You Get (WYSIWYG) editor, and needs to be simple and intuitive for typical users. This editor has all the basic components that can be added to a Service. These components (Images, Text Boxes, Edit Text and Buttons) are represented on the web page as an object that can be dragged and dropped into the services editor. The editor is represented by an image of a mobile device in order to the final aspect of the service be easily perceived by the user.

To understand the behaviors that users expect from an editor of this type, a few prototypes were made on paper with the main ideas and were tested with potential users. In Figure 21 and Figure 22 these paper prototypes are shown.

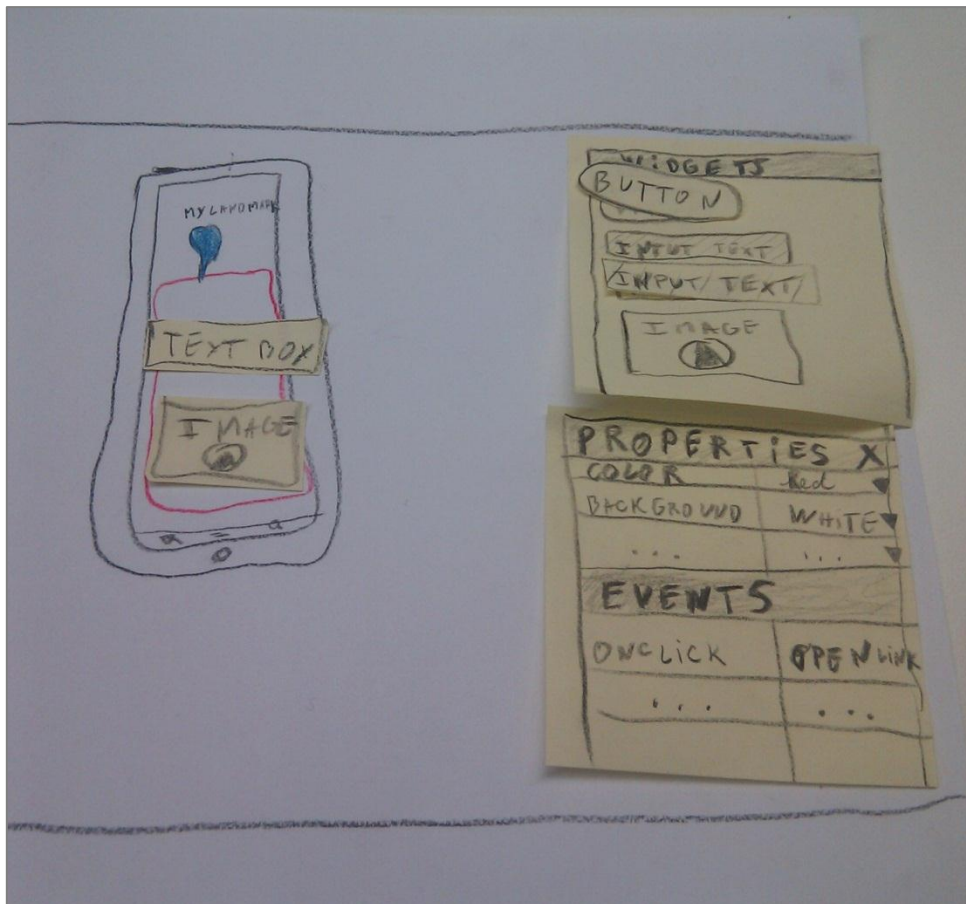


Figure 21 - Layout Editor Paper Prototype

During the test, it was asked to users that they create new services. In Figure 22 two examples of services are shown of services that users were asked to create.

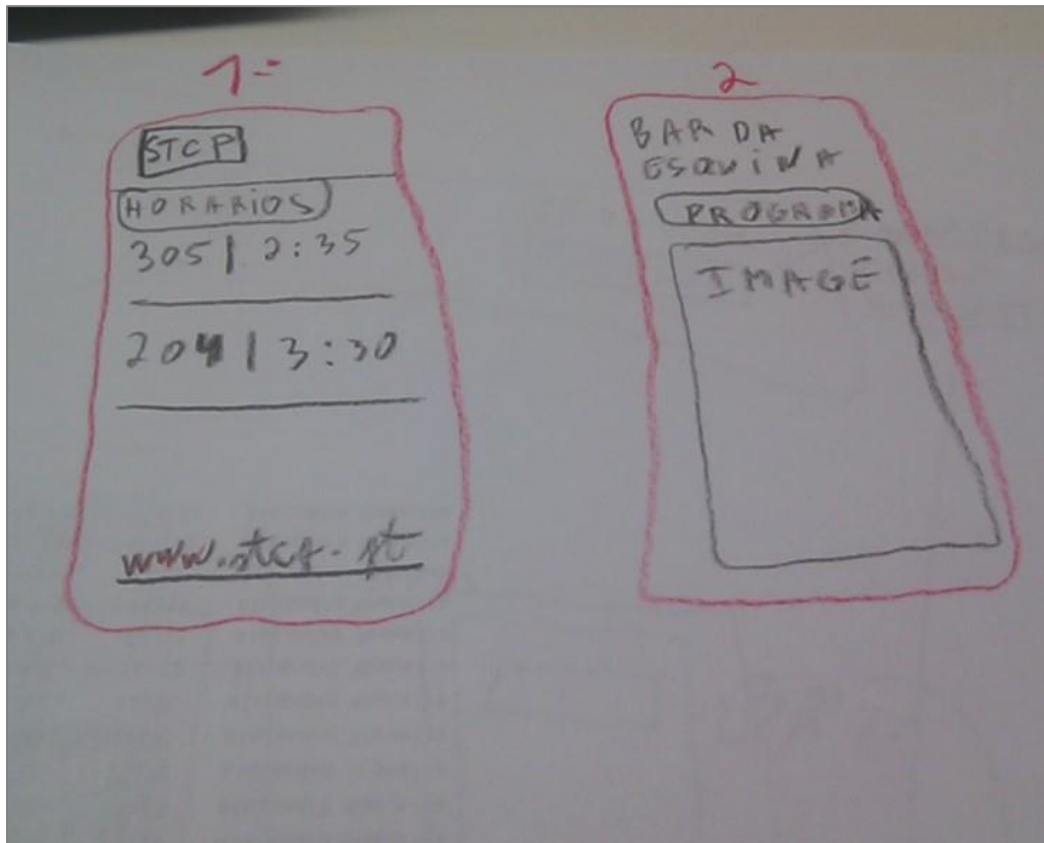


Figure 22 - Example of services that users would create

The first example of service (left side), is a service provided by a bus company where an example of a bus schedule is represented. The second example of service (right side), is a service that can be provided by any bar owner.

The creation of the requested services was easy for most users. When users were asked to create the service, the first impulse that they had, was to drag the components inside the editor. This reinforced the need to create an interface with drag and drop components.

4.3.2 - Final Implementation

During the editor development and after tests with users being made, the interface has been modified. The final version had a few additions of components as shown in Figure 23.



Figure 23 - Final version of Services Editor

In the content menu another type of content was added. This new type of components is a set of basic components. This set of components act as a template that users can easily add to the service. Instead of adding all the necessary components one by one, users can simply add a template and then change the basic components of the template.

4.3.3 - Implementation of Services Creation with Javascript

The services editor has one menu with all the available components such Images, Text Box, etc. These components were implemented with a script.aculo.us functionality called Draggables. After instantiating the object as a draggable, the user can move the object in the browser. All these components are implemented as a div with HTML.

The other missing element is the editor. The editor is implemented as a Droppable object. The editor background is an image of a mobile device. The user can drag the components inside the editor.

When the user drags and drops one component inside the editor, a new equivalent component is automatically created in the corresponding position. This allows users to add more than one component of same type. When the user is dragging the component, the component has a red border; when the user is dragging the component over the editor and the entire component is inside the editor, the border is blue. This indication allows users to know that they can only put the components inside the service editor. This behavior can be seen in greater detail in Figure 24 and Figure 25.

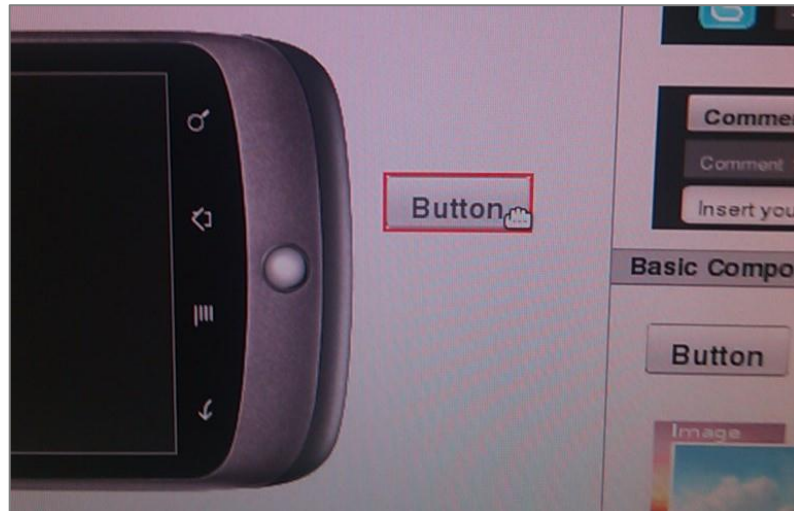


Figure 24 - User dragging a button to editor

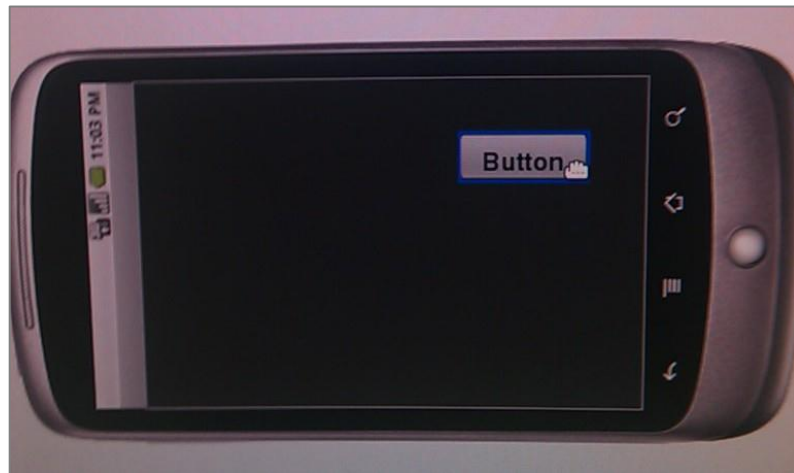


Figure 25 - User finishing the drag of a button inside the editor

After adding components to the editor, the user can change the size of the components. To edit the components inside the editor, the user needs to select them. After being selected, when the user puts the mouse over the borders of the component, the mouse pointer image changes, depending on which border side is near the mouse. This indicates to user that he/she can change the size of component. When the mouse pointer image changes, and if user clicks and drags, the component size can be changed. Figure 26 illustrates this behavior.

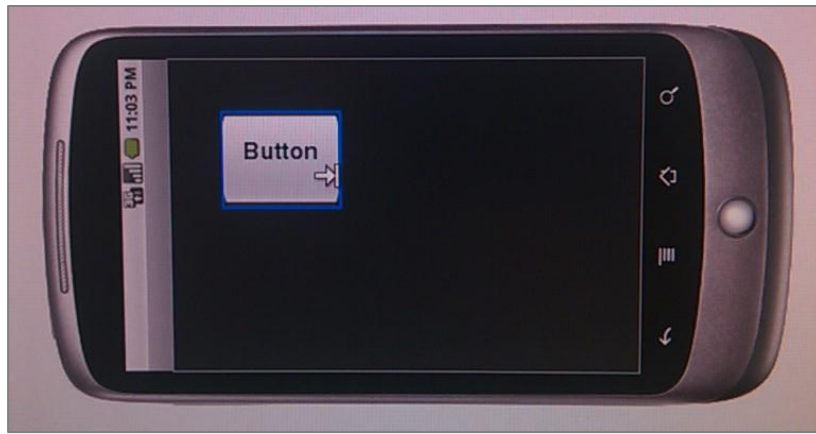


Figure 26 - User resizing a component

The basic components have another functionality. The component content can be easily edited and changed. If the user double clicks on a component, an interface immediately appears where the user can change the content. This behavior works for all components (Images, Text Boxes, Edit Text and Buttons). In the case of a Text Box, a Edit Text or a Button, a text field appears above the component with the actual text that can be changed. When the user clicks out of the component that is being edited, the content is saved. Figure 27 illustrates this process.

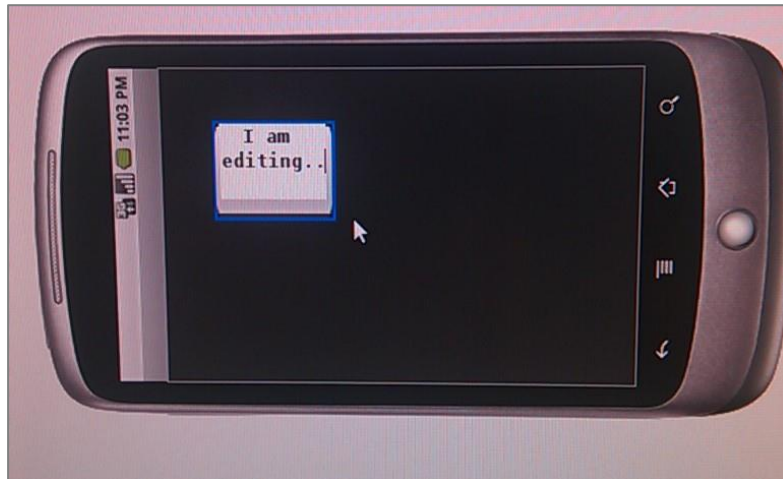


Figure 27 - User editing the text of a button

When the user adds one image to the editor or double-clicks on an image component, this time, the user does not change the Image text; a wizard appears where users can upload an image or even specify a URL for an image available on the web. Figure 28 shows this wizard.

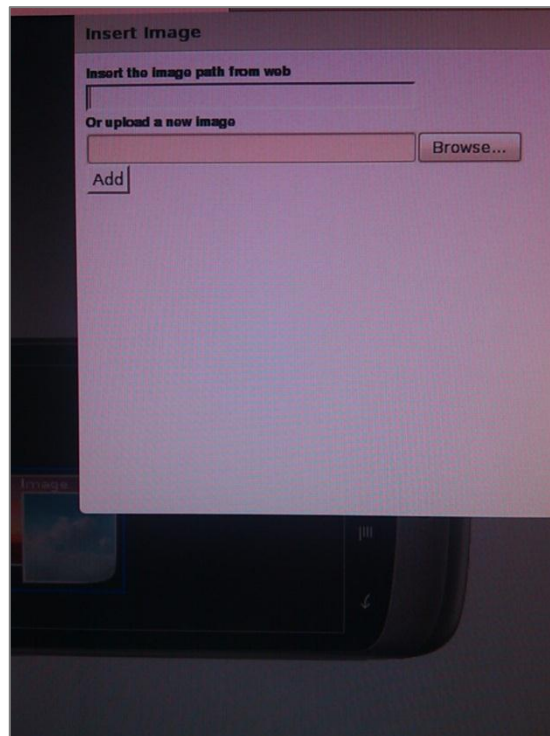


Figure 28 - User setting or changing the image of a Image component

The last two functionalities, the resize functionality and the edit content functionality were implemented using the possibility of the prototype to be extended. A plugin was developed for each feature mentioned. A plugin that allows any object to be resizable and another to allow that any object can be editable.

At the end, when the user saves the service, it is created and sent to the server with all the added components and their characteristics.

4.4 - Web Server Implementation Details

The web server is implemented with the framework Ruby on Rails. This framework uses the Model View Controller (MVC) architecture pattern. Following this pattern a Model was created for each Class existent on the Conceptual data schema. Not all models needed controllers. Only controllers for Landmarks, Services and Users were created. These controllers are responsible for implementing the business logic related with these entities. It was also necessary to add a controller to the Images (to manage the upload) and a controller to the Management. This latter controller is responsible for dealing with mobile application requests. These requests can be one of the following: “give me all the landmarks around me”, or “give me the service for a specific landmark”.

4.5 - Mobile Application Implementation Details

The Mobile Application was created using the Android framework. This application is an augmented reality application. It uses the camera phone to capture the image from the real world, and then add some additional info on top of that image.

One Android application is composed by several directories. The main directories are the src and the res. All the application packages are stored in the src directory and the files with Java classes are stored in the packages. The res directory is composed by several sub-directories. These sub-directories have the following names: layout, drawable and values. The layout directory has all the layouts used on application. The layout is a simple XML file that has all the components and their properties that will appear on the application such as Relative Layouts, Linear Layouts, ImageViews, TextViews etc. The directory drawable has all the images that will be used on the application. The values directory is used to store XML files that have constant values.

The mobile application source code is divided into four packages(net, geo, engine and moland).

The package net is where all the classes related with the network are stored. These classes are responsible for network request and content parsing, namely of the following types of content: Bitmaps, Landmarks on XML structure, JSON snippets and XML snippets. The parsers mostly used are the Bitmaps and Landmarks. The other two were implemented thinking in the future features.

The package geo has the classes related with geo-location. The important class Landmark is implemented in this package and also the LocationUtils class that has helpful static methods. These methods can be used to know the distance and also the azimuth between two locations on Earth. Both methods need to know the latitude and longitude of both the origin and destination to calculate the distance and azimuth. These methods are necessary to know the azimuth between the mobile device location and all the landmarks available on system.

The package engine has all the classes related with services rendering. These classes are responsible for making the services request to the server, parsing the returned content and creating all the service components. This package was developed for working as an engine for services rendering on this application. However this package is independent and can be distributed as a library. The main purpose of this engine is the creation and rendering of objects declared in one XML file. This XML file has objects declared following one specified structure. This structure can be changed and basic components or widgets can also be added that may appear in the next releases of Android. To port this application to another mobile platform it would only be necessary to port and implement the engine. So, it is possible to have the service description in a universal language for various platforms.

The main package is called moland and has the main application classes. One of these classes is the main activity. An activity is a single, focused “task” that the user can do inside one Android application. When the application starts, the initial activity prepares all the necessary components. One of these components is the camera; another is the GPS, if device has one. If the device does not have one GPS, it can also know the location with high accuracy using the MyLocation Google Service [3]. The magnetic sensor is also prepared in order to know to what direction the user is pointing the mobile device. After doing this setup, the activity makes a request to the server asking the landmarks around the mobile device available on the system. Once the application knows the device location and what the surrounding landmarks are, it is

A library responsible for services rendering was created. With this library it is possible to create and render, on the mobile device, the services that were created using the editor, available on the web. The integration between the web application and the mobile application is complete. The services are rendered on the device as they were created in the services editor.

The usability of both applications was one of the aspects that have been taken more into account during their development.

Chapter 5

Evaluation and Validation

5.1 - Overview

In order to evaluate the MoLand prototype, usability tests were made with eleven users. It is important to test the functionalities of each developed component. The user accesses the system through the Web Interface and the Mobile Application. The Web Interface and Mobile Application screenshots illustrate some of the main features of the developed prototype in the third subsection. Both are still under development with the purpose of making them more intuitive and user-friendly.

5.2 - Usability Tests

This section is divided into five sub-sections. First, is described the purpose of each script task. The script used to run the usability test can be found in Appendix A. The used methodology is described in the second sub-section. The third sub-section describes the environment where the test took place. All the results are documented for each task thoroughly in the fourth sub-section. And finally in the fifth sub-section the demographic data is presented. The test results and notes that were taken for each session can be found in Appendix B.

5.2.1 - Test Description

The purpose of the test was to evaluate the usability and ease of navigation of the MoLand components: Web Interface and Mobile Application.

At the beginning of the test, the moderator briefly explained the application and how the test would run. Users were asked to complete six tasks. The tasks were read aloud by the moderator. Each task was composed of subtasks. For each subtask the time and the number of errors that user made to complete it, were measured. An error can be one of the following situations:

- User chooses a wrong option;
- User does not understand how to complete the subtask.

For tasks related with Web Interface, the number of incorrect mouse clicks is indicated.

The objective of the first task was to understand if the users could register and log into the system. The moderator asked the users to register on the system and then to login.

In the second task, the objective was to understand if the users could create a landmark and to understand if users understand how to add a service to the landmark.

The third task was editing the service. Before beginning the task, a background of what the participant could make on this new interface was given. This was just to clarify a little of what the participant could do in the interface and what he could expect in the end. The purpose of the subtasks was to measure if users could drag the components from the available menu and drop them on the editor area. Then it was asked to the participant that he/she adds a personal image to understand if the participant could use the provided wizard. Then the participant was asked to resize the picture added to the service. This test served to verify if the users could see how to resize the components. The resize feature was implemented as it is typically implemented in drawing programs. Besides the drag and drop of the basic components, also the available templates and its concept were tested in the next subtask. At the end, the last task served to see if the user could save the created service.

The objective of the fourth and last task was to test the mobile application. The moderator asked the participant to find the landmark that he/she had just created in the system. To be able to find the landmark created, the participant had to move the mobile phone around until the landmark that was created appears.

To finish the test, users could give suggestions on any aspects of the application. At the end, the valuable collaboration of the users was recognized and a small gift was given to thank the participant.

5.2.2 - Description of Methodology

Each test session was led by one moderator, and also included one participant and a note taker.

The moderator employed a task-based think aloud protocol (25) . Many researchers use this technique in usability testing (26). Nielsen refers to it as "the single most valuable usability engineering method" (27).

The participants were asked to vocalize their thoughts, doubts, critics, compliments and the path they took to do the task. When they had difficulties to complete the tasks, the task was remembered as well as some initial information. Even so, if they could not complete the task, the moderator explained to them how it was supposed to be done. After the task was explained, the users could give their opinion about it. The users were informed that they would not be evaluated by their performance. The only thing that would be evaluated would be the application.

5.2.3 - Test Environment

The test was made in a vertical prototype. All the features tested are already implemented on that prototype. The participant had a computer to test the Web Interface and a mobile device

running the Android OS in order to test the mobile application. The test sessions were made in one room at Fraunhofer AICOS facilities. The environment was quiet and there were no distractions. The note taker and the moderator were both known by the participants. There were a total security and confidence between the moderator who led the conversation and the participants. Each test session had approximately 20 minutes.

5.2.4 - Usability Test Results

Task 1

The subtasks of task 1 were to register the user on the system and after to do login.

Subtask 1.1 and 1.2

All the participants completed the subtasks with success. To register a new user, they took an average of 18 seconds to complete the subtask without errors.

All the participants logged in to the system quickly. To login they took an average of 4 seconds and everyone completed the subtask without error.

One participant suggested that the system should have another field to confirm the password in the registry and another one said that a system for password recovery would be good.

Task 2

Task number 2 was one of the most important to test. This task was also the most difficult for participants. In this task, the users were asked to create a new landmark on the system and then to edit the service associated to the created landmark.

Subtask 2.1

In the first subtask, the participants had difficulties creating a landmark correspondent to their workplace. The participants had a map where they needed to indicate where they wanted to create the landmark. When participants clicked on map, the fields for longitude and latitude were automatically filled. The latitude and longitude fields are necessary to characterize a landmark. The second subtask was easier because the participants only needed to find the link that allows editing service.

All the participants completed the subtask. The participants took an average of 102 seconds to complete the subtask. Only 4 of them did not make errors in the process.

The following errors were made:

- Five participants tried to use the field name to search locations by address;
- Four participants saved the landmark without landmark name;
- Four participants double clicked to zoom in the map;
- Four participants did not read the available help information.

The participants did an average of 1.18 errors.

Subtask 2.2

In this subtask, the participants only needed to identify the option and click on it. They needed to identify the link that allows the service to be edited.

All the participants completed the subtask successfully. They took an average of 6 seconds to complete the subtask. The fastest managed to find the link in just one second and the slowest in 27 seconds. This subtask was evident and everyone managed to complete the subtask without errors.

Task 3

In this task the participants were asked to do tasks related with the service creation use case. To create services, the participants needed to use the available service editor. In this task all the editor features were tested. As this type of use case is not so common on the web, a simple overview was given to the participant about the interface and what he could do.

Subtask 3.1

In this subtask the participant was asked to add a text box to the service containing his/her name. To complete the subtask the participant needed to drag the component identified as a Text Box and drop it inside the editor. Then it was tested if users could change the content of the component. To edit the content of the text box, the participants needed to double click over the text box.

All the participants completed the subtask. The participants took an average of 23 seconds to complete the subtask. Only two participants committed errors. One participant selected a different component instead of the text box. And another clicked on the save button to save the content that finished editing instead of just clicking out of the component.

All the participants started to like what they saw and what they could do with the editor.

Subtask 3.2

In this subtask the users were asked to put their own image in the service that they were creating. This subtask tested if participants could use the wizard to change the image content. Because users did not have photos on the computer where they were doing the test, they needed to search for a photo, online. This time of search is also included in the time measured for task completion. However this time is not intrinsically related with the system.

All the participants completed the task with success. The participants took an average of 40 seconds to complete the subtask. They committed 0 errors and they are so excited that they started doing the next task without the moderator needing to ask. The next subtask was the resize operation for those who would like to see their photo with another size on the service editor.

Subtask 3.3

In this subtask the participants were asked to put the size of their photo in the editor smaller. The objective of this test was to test if users could do the resize operation.

Not all the participants managed to complete the task; one participant failed. The participant who failed was doing the right action, however the mouse sensitivity was too high and he could not see the mouse images changing when the mouse was over the component borders. The other participants completed the task very quickly with an average of 10 seconds.

Subtask 3.4

This subtask served to verify if participants assimilated the concept of templates. The participants were asked to add a template where they could provide their Facebook updates. The participants had only to drag the Facebook template and drop it into the editor.

All the participants completed the subtask. The participants took an average of 8 seconds to complete the task. As this functionality is not yet finished, the participants asked how would it work. The moderator explained them that the implementation of the menu Properties is in the roadmap. The users can then specify various properties associated with the components. The templates will have a property where the user can insert the Facebook account. Then on the mobile application the last user updates on Facebook will appear. The other templates will have the same behavior, such as Twitter and Comments Templates. In the future, other templates will be supported.

Subtask 3.5

This subtask tested if users could easily save the created service. All the users completed the subtask and did not commit errors. They took an average of 3 seconds to complete the task.

After this task, the participants were thrilled with what they could do with the system and the freedom it gave to users. One of the participants gave a good suggestion that could be implemented in the future: a contextual menu for each component that could appear when the user clicks on the component with the right mouse button.

Task 4

In this task the users changed from the web browser to the mobile application. The moderator gave to the participant a mobile device that had the application installed. The objective of this test was to test the usability of the developed mobile application.

Subtask 4.1

To complete this subtask, the participants had to move the mobile device around, until they found the landmark on the screen. When the participants found the landmark, they mentioned it to the moderator.

All the participants completed the task with success. They took an average of 2 seconds to complete the task. It was the first time that some participants saw an application of augmented reality on a mobile device. They did not know what to expect from this experience. But as the landmark was near the participants, they only needed to move a little around to find something unusual on the screen and realize that that was the created landmark. As the landmark was very close to the participant (workplace), the landmark appeared near the bottom of the screen and the participants could not see the information about the landmark. This happens because the nearest landmarks are represented on the bottom of the screen and the farther away are represented in the top of the device and with a small size.

Subtask 4.2

In this subtask it was tested if users knew how to see the landmark service. After founding the landmark, the users only needed to touch on the landmark position to see the service.

All the participants completed the subtask. They realized quickly that they needed to touch the landmark to see the service. The participants took an average of 2 seconds to complete the subtask. The participants were very enthusiastic when they saw the service rendered on the mobile application just like it was created by them, in the web application.

5.2.5 - Demographic Data Results

For this usability test, eleven persons collaborated with ages between 22 and 31 years. All the participants were of the male sex.

All the participants are presently working at Fraunhofer AICOS. Four of the participants are presently finishing their undergraduate courses. Two of these four are attending the Master in Informatics and Computing Engineering and the other two are attending the Master in Networks and Informatics System Engineering.

Among the rest of the participants there are: a designer, an accountant and five researchers. The researchers are conducting research in computer science. The designer does work for the information technology area. And the accountant does not do any work directly related to the information technology area.

Testing the system with a range of users in several areas allowed identifying a greater number of system problems.

5.3 - Interface

When the end user accesses the MoLand web interface, a welcome page is displayed. Then the user can choose between three resulting sections: Friends, Landmarks and Services.



Figure 30- MoLand welcome page interface

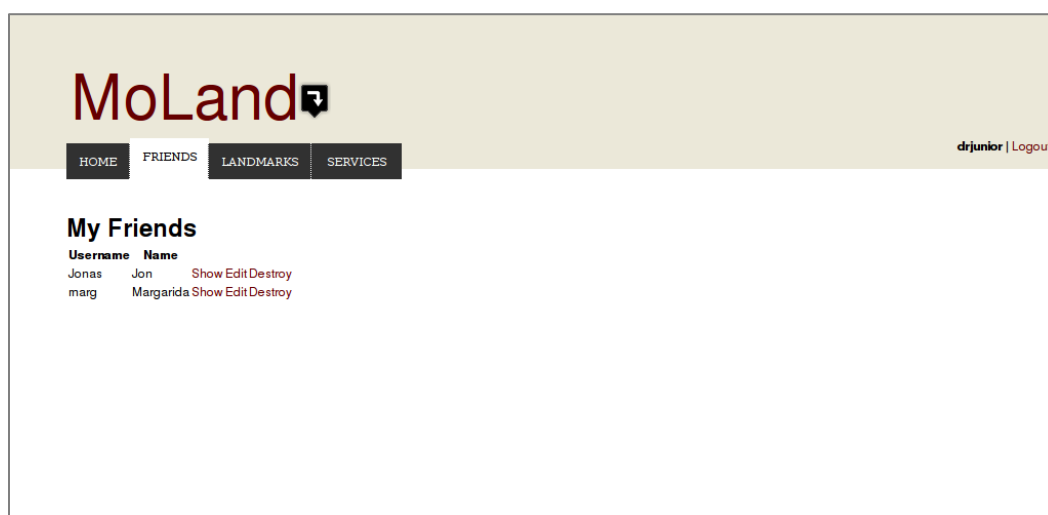


Figure 31 - MoLand friends interface

MoLand

drjunior | Logout

HOME

FRIENDS

LANDMARKS

SERVICES

My landmarks

Name	Address	Latitude Longitude	Info	
Vilar do Pinheiro		41.2703 - 8.67951		Edit
Campanhã		41.1505 - 8.58624		Edit
Combatentes		41.1653 - 8.59846		Edit
Modivas Sul		41.2853 - 8.6937		Edit
Hospital São João		41.1833 - 8.60224		Edit
Francos		41.1656 - 8.63635		Edit
Jardim do Morro		41.1376 - 8.60865		Edit
IPO		41.1813 - 8.60452		Edit
Câmara de Gaia		41.1297 - 8.60612		Edit
Trindade		41.1523 - 8.6093		Edit
Lapa		41.1571 - 8.61672		Edit
Árvore		41.3404 - 8.72556		Edit
Hospital Pedro Hispano		41.1804 - 8.66623		Edit
São Bento		41.1449 - 8.61082		Edit
Cândido dos Reis		41.2007 - 8.6497		Edit
Custió		41.2223 - 8.639		Edit
Mindelo		41.3151 - 8.71421		Edit
Pedras Rubras		41.2462 - 8.66181		Edit
Matosinhos Sul		41.1802 - 8.68855		Edit
Marquês		41.1611 - 8.60427		Edit
Mercado		41.1875 - 8.69339		Edit
Esposade		41.2161 - 8.65454		Edit
Estádio do Dragão		41.1607 - 8.58242		Edit
General Torres		41.1339 - 8.60749		Edit
Aliados		41.1486 - 8.61094		Edit
D. João II		41.1207 - 8.60616		Edit
Aeroporto		41.2371 - 8.66944		Edit
Póvoa de Varzim		41.3781 - 8.75808		Edit
24 de Agosto		41.1488 - 8.59835		Edit
Sr. de Matosinhos		41.1882 - 8.68512		Edit
Bolhão		41.1498 - 8.6059		Edit
Heroísmo		41.1467 - 8.59298		Edit
Salgueiros		41.1697 - 8.59874		Edit
S. Brás		41.3735 - 8.75412		Edit
Pias		41.2082 - 8.64712		Edit
Faria Guimarães		41.1572 - 8.60914		Edit
Casa da Música		41.1606 - 8.62828		Edit
Parque de Real		41.1791 - 8.67355		Edit
João de Deus		41.1261 - 8.60563		Edit
Câmara de Matosinhos		41.1807 - 8.68123		Edit
Polo Universitário		41.1742 - 8.60361		Edit
Araújo		41.2169 - 8.64098		Edit
Carolina Michaelis		41.1586 - 8.62225		Edit
Restaurante Marialva	Alameda Ribeiro do Fontão, 5460 Boticas, Portugal	41.6891 - 7.66728	O melhor restaurante de Boticas.	Edit

Actions

New landmark

Figure 32 - MoLand landmarks interface

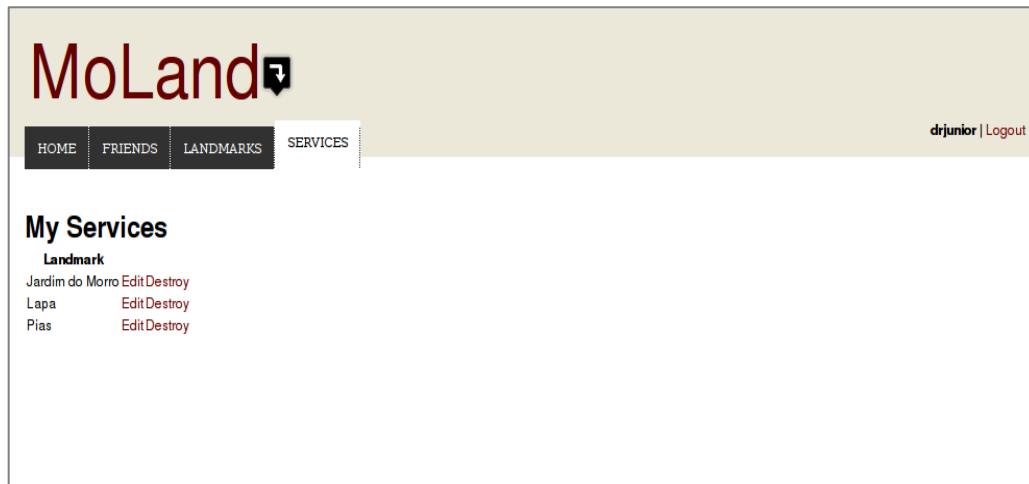


Figure 33 - MoLand services interface

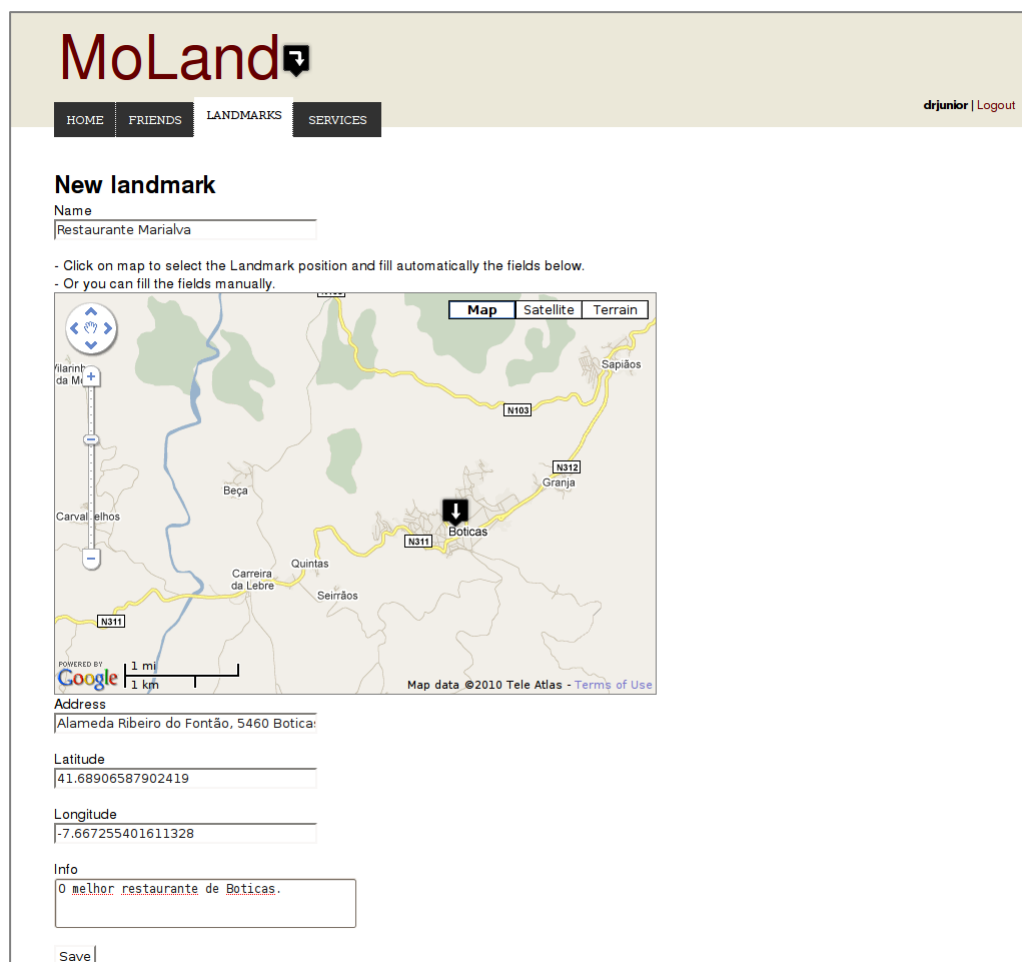


Figure 34 – Interface where user is creating a new landmark

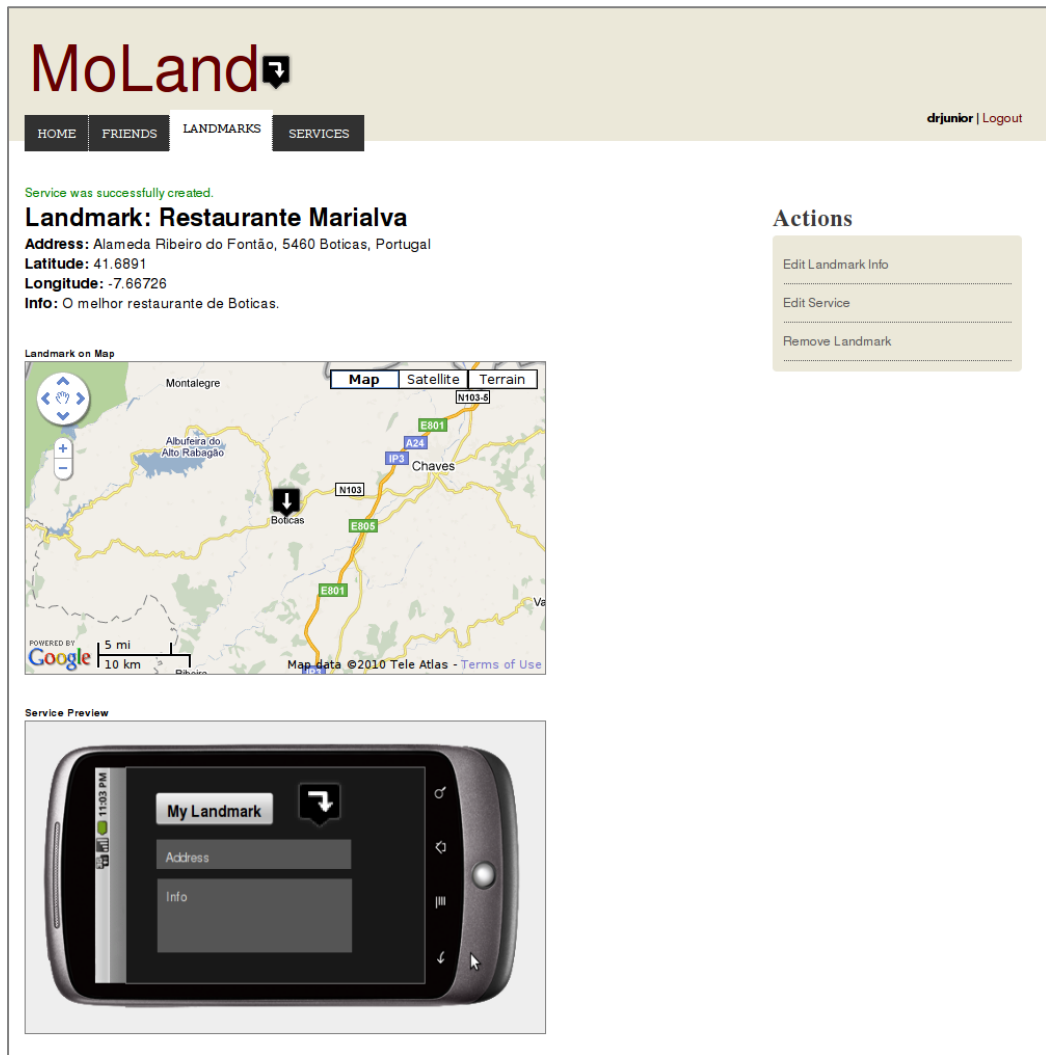


Figure 35 – Interface with landmark info saved and with a preview of the service initially associated



Figure 36 – Interface where user is creating a service for a Portuguese restaurant



Figure 37 - Mobile interface with the Landmarks around

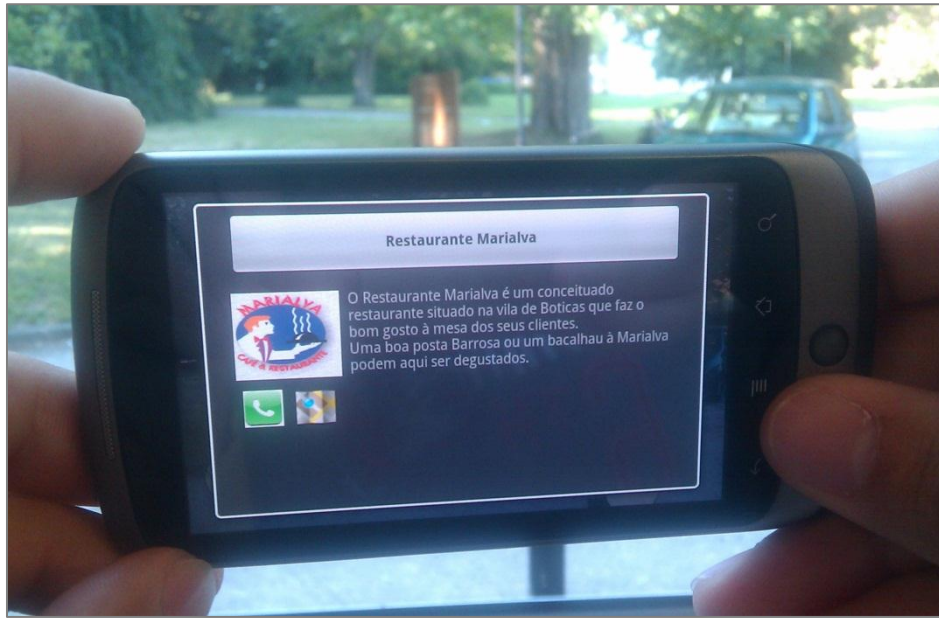


Figure 38 - Service rendered on mobile device

5.4 - Summary

In this chapter the functionalities of the MoLand prototype were evaluated through a usability test. The test showed that users are able to use the system and they are very excited with what they can do with it. However, there are functionalities on the system that need to be optimized. Among all the implemented and tested features, only the functionality of inserting landmarks needs to be optimized with a high priority. The problem was identified as: the users are not used to use the map and they expect always a mechanism where they can search the locations by providing the address. This problem will surely be addressed in future developments.

Chapter 6

Conclusions and Future Work

6.1 - Fulfilling the objectives

The result of this dissertation is a proof-of-concept prototype of the Location Based Services for Everyone concept.

The first part of the work consisted of studying the various fields related to this work. The basics of geo-location were studied as well as all the features and capabilities that may arise from its use. This study included also the basics of location based services and the concepts related to augmented reality. Then most applications already available were studied. These applications were selected because they have functionalities that can be related to the scope of this project, specifically augmented reality and location-based functionalities. This study was important to understand what are the potentialities and limitations of existing solutions. Next it was necessary to define a solution to the Location Based Services for Everyone system, and from this specification emerged MoLand. The specification will allow future developments of the system and integration with other modules.

The developed MoLand prototype takes advantage of Android capabilities to create an augmented reality application where users can see the landmarks around the mobile device. Using a web version of the system, users can do all the basic use cases related to landmarks and services. This web prototype is only a means of reaching a better integrated solution. Services are composed by components that users can add using the editor available. This editor had a major development effort. The services created in the web interface appear in the mobile application, as they were created by the user. This integration was achieved with success. The services editor on the web and the services rendering on the mobile application are one of the main innovations that can be found on this dissertation.

6.2 - Future Work

The solution specified in Chapter 3 refers to a service in the following manner: A Service represents the information that will be available in a Landmark. In the developed solution, services have static information that can be viewed on the mobile device. This information is transmitted by several graphic components that are available for users to add and edit the content. However, this prototype has also implemented the foundations for the development of future features, namely the ability to specify functions that can be associated with actions performed by the users when they are interacting with the services on the mobile application. An example: the user can add a button to the service and specify that when the button is clicked a specific web page opens. There are an infinite number of actions that can be specified and some will be added to the roadmap.

Another functionality that would be interesting is the possibility of specifying a service with more than one layout. This would allow users to specify a hierarchy in their service as well as the flow between the various layouts.

Besides typical users, it is also important to allow companies that already have web services. They will be capable to add their services to the MoLand without much effort. It would also be interesting to allow this type of companies to add services for landmarks that have a context and let the companies specify the business logic only for one landmark and apply that business logic for all the landmarks depending on the context.

Currently, it is not possible to do many things on the prototype that were planned in the beginning, such as the search functionality, the layers, the interaction with landmarks and friends, etc. These features are also present in the roadmap.

Another functionality that would be very interesting is the rendering of the services in the web interface. This gives users the security that they are really creating a service as they want, and this feature would be very useful for testing purposes when users are creating services.

One functionality that is already implemented but can be improved and be added more features is the API provided to developers interested in using the data available in MoLand and would like to add new applications. With the current API it is only possible to know what are the landmarks on the system and the services provided by them.

References

1. Ian Heywood, Sarah Cornelius, Steve Carver. *An Introduction to Geographical Information Systems*. s.l. : Longman Pub Group, 1998.
2. Monico, J F G. *Posicionamento pelo NAVSTAR-GPS: descrição, fundamentos e aplicações*. 2000. 8571393281.
3. Zhengrong Ji, Ravi Jain. Official Google Mobile Blog: Google enables Location-aware Applications for 3rd Party Developers. *googlemobile.blogspot.com*. [Online] June 2008. <http://googlemobile.blogspot.com/2008/06/google-enables-location-aware.html>.
4. Stefan Steiniger, Moritz Neun and Alistair Edwardes. Foundations of Location Based Services. *Geo Unizh*. [Online] 2009. http://www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf.
5. World, GSM. *Location Based Services*. s.l. : GSM World, 2003.
6. *Augmented reality as a design tool for mobile interfaces*. Bertelsen, Olav W. and Nielsen, Christina. New York, NY, USA : s.n., 2000. ACM. pp. 185--192.
7. *GeoHealth: a location-based service for nomadic home healthcare workers*. Christensen, Claus M. and Kjeldskov, Jesper and Rasmussen, Klaus K. Adelaide, Australia : ACM, 2007. pp. 273--281. 978-1-59593-872-5.
8. Revolver360. [Online] http://revolver360.files.wordpress.com/2009/07/augmented_reality.jpg.
9. Parr, Ben. Mashable. *mashable.com*. [Online] 2009. <http://mashable.com/2009/08/19/augmented-reality-apps/>.
10. ARToolKit. ARToolKit. *ARToolKit*. [Online] 2004. <http://www.hitl.washington.edu/artoolkit/>.
11. NyARToolkit for Android.en. *Nyatla*. [Online] 2008. <http://nyatla.jp/nyartoolkit/wiki/index.php?NyARToolkit%20for%20Android.en>.
12. LayaR. LayaR. *LayaR*. [Online] LayaR. www.layar.com.
13. Sapo. Casa Sapo. [Online] Sapo. <http://casa.sapo.pt/news/detalhe.aspx?id=9311>.
14. Wikitude. Wikitude. [Online] www.wikitude.org.
15. junaio. junaio. [Online] www.junaio.com.
16. Tagwhat. Tagwhat. [Online] www.tagwhat.com.
17. GeoVector. World Surfer. *GeoVector*. [Online] <http://www.geovector.com/applications/world-surfer/>.
18. IBM. Do you see what we see? IBM. [Online] <http://www-07.ibm.com/innovation/au/ausopen/action.html>.
19. Google. Google maps for mobile. *Google.com*. [Online] [Cited: January 10, 2010.] <http://www.google.com/mobile/maps/>.
20. Gowalla. Gowalla. [Online] www.gowalla.com.

21. **Foursquare.** *Foursquare*. [Online] www.foursquare.com.
22. **Gonzalez, Nick.** *TechCrunch*. [Online] April 28, 2008. <http://techcrunch.com/2008/04/28/yelp-lets-businesses-fight-back/>.
23. **Yelp.** *Yelp*. [Online] www.yelp.com.
24. **Buuuk.** *Buuuk*. [Online] www.buuuk.com.
25. **J., Senger C.** *Thinking Aloud Protocols: A Diagnostic Tool for teaching Legal Problem Solving*. 1993, pp. 367-382.
26. **Gould, J. D.** How to design usable systems. *Human-computer interaction: toward the year 2000*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1995, pp. 93--121.
27. **Nielsen, Jakob.** *Usability Engineering*. San Francisco : Morgan Kaufmann, 1993, 6.
28. **Google.** *Android Official Website*. [Online] January 2010. <http://code.google.com/android/>.
29. **Warren, Christina.** Yelp Comes to Google Android. *Mashable*. [Online] 7 de December de 2009. <http://mashable.com/2009/12/07/yelp-android/>.
30. **Travis, Greg.** Build your own Java Library. *IBM Developer Works*. [Online] <http://www.digilife.be/quickreferences/PT/Build%20your%20own%20Java%20library.pdf>.
31. **Stankiewicz, Adam.** Mobile Application Development using javascript. [Online] <http://blog.frontendforce.com/2010/05/mobile-application-development-using-javascript/>.
32. **Rúben Fonseca, Alberto Simões.** Alternativas ao XML : YAML e JSON. 2007. 978-972-99166-4-9.
33. *Location based applications for mobile augmented reality*. **Reitmayr, Gerhard and Schmalstieg, Dieter.** Adelaide, Australia : s.n., 2003. Australian Computer Society, Inc. pp. 65--73.
34. **MySQL.** *MySQL Cluster for Location Based Services*. 2009.
35. *Location based services using android*. **Kumar, Sandeep and Qadeer, Mohammed Abdul and Gupta, Archana.** Bangalore, India : IEEE Press, 2009. ACM. pp. 335--339. 978-1-4244-4792-3.
36. *System support for mobile augmented reality services*. **Kimura, Hiroaki and Tokunaga, Eiji and Nakajima, Tatsuo.** New York, NY, USA : s.n., 2007. ACM. pp. 1616--1623.
37. **Fonseca, José Reynaldo da.** [Online] 16 de July de 2005. http://upload.wikimedia.org/wikipedia/commons/5/5c/Confluencia_23_S_x_49_W_-_GPS.jpg.
38. **Brightkite.** *brightkite*. [Online] <http://blog.brightkite.com/wp-content/uploads/2009/08/layar-screenshot.jpg>.
39. *Recent Advances in Augmented Reality*. **al, R.T. Azuma et. [ed.]** IEEE Computer Graphics and Applic. 2001. pp. 34-47.

Appendices

Appendix A

The script used and the tasks used to evaluate the system are the following:

“Script

MoLand is a system that allows its users to create landmarks. A Landmark is simply a geographical location that can refer to any object or entity. For example, you can create a landmark for the city of Porto, you can create a landmark for the current position of a bus or even create a landmark for your workplace.

Once the landmarks are added to MoLand, it is possible through a mobile device, with an augmented reality application, to search and interact with existing landmarks. Just to simplify, what an Augmented Reality application does, it only adds information to a real scene captured by a camera.

Besides providing added Landmarks, the MoLand system also allows the user to add a service for each Landmark. This service consists of information that the user can add and configure.

To test the usability of the system MoLand, I would like to test it with you.

Throughout the test you will be asked to interact with the MoLand system through the web browser as well as the existing application on the mobile device.

During this test, I would like to ask you if you can think aloud as you perform the task, and if you have compliments or criticisms, I want that you feel free to share them.

Before we start the test, please be aware that the main purpose of the test is to evaluate the system usability and performance and not assess who is testing.

Tasks

Task 1

1.1

- Please sign up in the system.

1.2

- Please login.

Task 2

2.1

- Please, add a new Landmark to the Moland system corresponding to your place of work.

2.2

- Now, could you show me how would you proceed to change the service that is associated with the landmark that you just created.

Task 3

In this interface you can put all the information you want through the addition of content available. There are already some templates to help users, which can be added wherever you want. After saving the service, it will be displayed on your phone just like it was configured.

3.1

- Now, I will ask you to put a text box with your name in the service.

3.2

- Now, imagine that you want to put your photo in the service. How could you proceed?

3.3

- How could you proceed to put your photo smaller?

3.4

- Now, imagine that you would like to share your last latest updates of your Facebook account.

3.5

- Now that you finished editing the service, please save it.

Task 4

<Now, the user picks up the mobile phone and opens the mobile application>

4.1

- Please, find the landmark that you just created in the system.

4.2

- Imagine that you became interested in the landmark that you found. Please, see the service that landmark provides.

End

Thanks for your availability and you can now take one of the gifts available!”

Appendix B

This attachment has all the grids used by the note taker. All the subtasks are grouped by task in a single table. For each subtask where the field is filled with yes or no, depending if the user completed the subtask, the subtask has two other columns where is indicated the duration that participant took to complete the subtask and the number of errors that the participant committed doing the task. The time is indicated in minutes following this format: mm:ss.

For each task there is a table of notes where the note taker wrote all the suggestions, some notes and the errors occurred during the test session.

Task 1:

Name	Register a new user (yes/no)	Time	Errors	Understand how to login(yes/no)	Time	Errors
Participant1	YES	00:15.53	0	YES	00:03.32	0
Participant2	YES	00:17.76	0	YES	00:04.41	0
Participant3	YES	00:10.20	0	YES	00:03.98	0
Participant4	YES	00:22.86	0	YES	00:03.06	0
Participant5	YES	00:29.59	0	YES	00:06.09	0
Participant6	YES	00:25.68	0	YES	00:05.51	0
Participant7	YES	00:14.28	0	YES	00:02.80	0
Participant8	YES	00:12.21	0	YES	00:03.60	0
Participant9	YES	00:17.05	0	YES	00:08.60	0
Participant10	YES	00:14.25	0	YES	00:08.02	0
Participant11	YES	00:33.29	0	YES	00:01.05	0

Table 2 - Times and errors for Task 1

Name	Notes
Participant1	-
Participant2	-
Participant3	-
Participant4	-
Participant5	-
Participant6	-
Participant7	Suggestion: password recovery system.
Participant8	-
Participant9	-
Participant10	-
Participant11	Missing field to confirm the password.

Table 3 - Notes taken for Task 1

Task 2:

Name	Managed to create new landmark (yes/no)	Time	Errors	Managed to edit the landmark service (yes/no)	Time	Errors
Participant1	YES	02:06.07	1	YES	00:03:97	0
Participant2	YES	02:15.28	2	YES	00:02.27	0
Participant3	YES	01:07.16	2	YES	00:01.73	0
Participant4	YES	01:07.55	0	YES	00:03.78	0
Participant5	YES	01:43.53	2	YES	00:04.74	0
Participant6	YES	02:04.48	1	YES	00:01.88	0
Participant7	YES	01:44.18	0	YES	00:27.64	1
Participant8	YES	02:01.20	4	YES	00:05.23	0
Participant9	YES	01:23.08	0	YES	00:03.56	0
Participant10	YES	01:58.88	1	YES	00:08.75	0
Participant11	YES	01:20.93	0	YES	00:10.67	0

Table 4 - Times and errors for Task 2

Name	Notes
Participant1	Delay searching the map. Ask if could search by address in the field address. Did not read the tip. Error: Save without landmark name.

Participant2	Did not read the tip. Error: Tried to search by address in the field address.
Participant3	Error: Tried to search by address in the field address. Error: Did two clicks to do zoom on map.
Participant4	Reads the tip.
Participant5	Error: Tried to search by address in the field address. Error: Did two clicks to do zoom on map.
Participant6	Error: Tried to search by address in the field address. Error: Did two clicks to do zoom on map.
Participant7	-
Participant8	Error: Tried to search by address in the field address and saved with empty fields. Error: Did two clicks to do zoom on map.
Participant9	-
Participant10	Error: Saved with empty fields. Error: Did two clicks to do zoom on map.
Participant11	The landmark disappeared when participant typed the backspace key. Suggestion: Back navigation control was missing.

Table 5 - Notes taken for Task 2

Task 3:

Name	Managed to put the text box with name (yes/no)	Time	Errors	Managed to put the photo (yes/no)	Time	Errors	Managed to resize the photo (yes/no)	Time	Errors	Managed to add the Facebook template (yes/no)	Time	Errors	Managed to save the service (yes/no)	Time	Errors
Participant1	YES	00:23.17	0	YES	00:27.51	0	YES	00:06.95	0	YES	00:05.22	0	YES	00:02.60	0
Participant2	YES	00:15.17	0	YES	00:20.83	0	YES	00:10.69	0	YES	00:05.15	0	YES	00:04.64	0
Participant3	YES	00:12.96	0	YES	00:13.89	0	YES	00:09.62	0	YES	00:03.93	0	YES	00:04.77	0
Participant4	YES	00:28.84	0	YES	00:35.95	0	YES	00:10.34	0	YES	00:07.09	0	YES	00:05.61	0
Participant5	YES	00:48.92	2	YES	00:57.45	0	YES	00:09.80	0	YES	00:12.31	0	YES	00:04.72	0
Participant6	YES	00:18.17	0	YES	00:03.35	0	No	00:32.48	0	YES	00:13.54	0	YES	00:03.13	0
Participant7	YES	00:21.23	2	YES	01:01.16	0	YES	00:05.75	0	YES	00:06.86	0	YES	00:04.53	0
Participant8	YES	00:39.45	0	YES	00:22.42	0	YES	00:03.73	0	YES	00:10.99	0	YES	00:04.08	0
Participant9	YES	00:26.96	0	YES	00:46.78	0	YES	00:19.51	0	YES	00:08.63	0	YES	00:01.70	0
Participant10	YES	00:09.51	0	YES	00:21.23	0	YES	00:02.11	0	YES	00:05.42	0	YES	00:05.82	0
Participant11	YES	00:19.98	0	YES	00:26.68	0	YES	00:08.43	0	YES	00:22.53	0	YES	00:03.13	0

Table 6 - Times and errors for Task 3

Name	Notes
Participant1	He did a photo upload. Said that the resize is very nice.
Participant2	Inserted a URL on image wizard.
Participant3	Inserted a URL on image wizard. Realized that when do a resize the component stays at front.
Participant4	Inserted a URL on image wizard. Said that is a very good visual experience. Participant said that was waiting for a contextual menu.
Participant5	Error: Chosen input text instead text box.
Participant6	Error: Did not manage the resize subtask because was selecting out of the component.
Participant7	Inserted a URL on image wizard. Suggestion: The saved is a little hidden.
Participant8	Errors: Click on comment first. He did the save command to save the component value.
Participant9	-
Participant10	Suggestion: The components should appear in other side.
Participant11	-

Task 4:

Name	Managed to find the landmark (yes/no)	Time	Errors	Managed to touch on landmark to see the service (yes/no)	Time	Errors
Participant1	YES	00:38.81	0	YES	00:02.93	0
Participant2	YES	00:22.03	0	YES	00:01.50	0
Participant3	YES	00:53.58	0	YES	00:01.53	0
Participant4	YES	00:38.20	0	YES	00:01.82	0
Participant5	YES	00:35.47	0	YES	00:01.87	0
Participant6	YES	00:34.56	0	YES	00:01.64	0
Participant7	YES	00:06.31	0	YES	00:02.90	0
Participant8	YES	00:23.46	0	YES	00:05.19	0
Participant9	YES	00:33.05	0	YES	00:01.27	0
Participant10	YES	00:13.04	0	YES	00:01.66	0
Participant11	YES	00:08.26	0	YES	00:04.54	0

Table 8 - Times and errors for Task 4

Name	Notes
Participant1	-

Participant2	- Table 9 - Notes taken for Task 4
Participant3	-
Participant4	First checks the location of the landmark on map and then picks up the phone.
Participant5	Leaves the room looking for the landmark. The landmark found in the device is weird because it is in the bottom of the screen.
Participant6	-
Participant7	-
Participant8	-
Participant9	-
Participant10	-
Participant11	Asked why the landmark is appearing on the bottom of the screen.