

Inferring Social Networks from Distributed Bluetooth Scanning

Félix Manuel Rubio

DTU



Kongens Lyngby 2012
IMM-PhD-2012-118

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-PhD-2012-118

Abstract

Social behavior of an audience in a large-scale event is a valuable source of information and knowledge that can be exploit with the use of the right tool. Related to this issue, this thesis aims to develop an application, based on Android operative systems, presenting a game that has been called Roskilde Hide and Seek, which presents two different points of view. First of all, seeing as a game, the main goal is to guide the user to find a person in the experimental area, studying two concepts here, the way that the user has found that person and the observation of the ability of the game to motivate the user to move around the experimental area. Second of all, as a source of information, the system has to collect data from mobile devices to perform a social behavior research. The built application converts the mobile devices that have installed the game into distributed scanners by the use of Bluetooth and GPS sensors, which allows to obtain data from each one of the mentioned devices. The application is implemented to run in the area of Roskilde Festival, one of the biggest music festival in Europe and the results obtained after the experiment have been detailed and analyzed, assessing the concept of "gamification" and more specifically, the Roskilde Hide and seek application as a proper tool for the raised issue.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an M.Sc. in Digital Media Engineering.

The thesis supervisors are Jakob Eg Larsen and Sune Lehmann Jørgensen, Department of Informatics and Mathematical Modeling, Technical University of Denmark. The thesis is the result of work carried out in the period from February 2012 to September 2012 with a workload of 32,5 ECTS points.

Lyngby, 12-September-2012

Félix Manuel Rubio

Acknowledgements

First of all, I would like to thank my supervisors Sune Lehmann Jørgensen and Jakob Eg Larsen, for their guidance and patience, and for giving me the opportunity of being part of this amazing project.

I would like to thank the person who is more than a support in my daily life and always believes in me, Jose Luis de la Peña, and I also need to thank Susana Rojas for motivating me to achieve the best.

Lastly, I would like to thank my family, my parents Presentación Gallego-Preciados and Félix Rubio, my brother Alejandro Rubio, and my grandmother Elena and grandfather Manuel, the people who made who I am, giving me their support in each step of my life.

Contents

Abstract	i
Preface	iii
Acknowledgements	v
1 Introduction	1
1.1 Motivation	2
1.2 Thesis aim	3
1.3 Thesis Structure	3
2 Roskilde Hide and Seek	5
2.1 Game proposal	5
2.2 Outcome	6
3 Related work and experiment area	7
3.1 Gamification	8
3.2 Roskilde Festival	9
4 Analysis	11
4.1 Requirements Analysis	11
4.2 Use case diagram	13
4.3 Case Scenario	14
5 Design	17
5.1 Overview of the System	17
5.2 Sequence Diagram	18
5.3 User Interface	20
5.4 Profile Form	21

5.5	Points table of the game	23
5.6	Server and Database	23
6	Implementation	29
6.1	Overview	29
6.2	Library BTScanner	30
6.3	Android Interface	31
6.4	Android application	32
6.4.1	Informing phase	32
6.4.2	Collecting user data	33
6.4.3	Starting the game	34
6.4.4	Interactive Map	34
6.4.5	Map menu	36
6.4.6	Confirmation search procedure	37
6.5	Server side	38
7	Evaluation	41
7.1	Android application results	41
7.2	Distributed scanners data	42
7.3	Game data	44
7.3.1	Find a target	46
8	Discussion	51
8.1	What has been accomplished	51
8.2	Concept of the game	52
8.2.1	Entertainment of the game	52
8.2.2	Distance to find the target	53
8.2.3	Bluetooth and GPS in a festival	53
8.3	Gamification	53
8.4	Distributed scanner	54
8.5	Future work	54
9	Conclusion	57
A	Screenshots of the application Roskilde Hide and Seek	59
B	Class diagram	65
B.1	Hide and seek package	65
B.2	Hide and seek utility	65
C	Position graph method	69
C.1	postitionTrack.py	69
	Bibliography	73

CHAPTER 1

Introduction

The social relation has been changing during the past few years. The communication among people has evolved to a virtual level. Smartphones have become an important part of this social relation, allowing to communicate with anyone, anywhere at anytime by means of the internet and new mobile features.

Applications for smartphones is another new concept that has appeared lately. The companies who are responsible of the OS of the smartphones have built useful APIs to develop applications on them. Therefore, taking account of the possibility to collect the information with the new features and the easiness of developing apps, new smartphones bring the chance to study the behaviour of the people by monitoring them, transforming the smartphone into a **mobile sensor**.

One of the main goals of this chapter is to introduce the concepts of social network and some experiments carried out related social network. In addition to the use of the smartphone as a tool to gather information of users.

1.1 Motivation

During the past century there has been a great development on social network analysis. Researches conducted [WF94] [Fre04] have obtained answers to some social questions by providing formal definitions to aspects of economic, political or social structure. These researches have succeeded in understanding better our society and provide new services in order to make easier everyday life. As an example of a social network, it can be named the service Amazon, which through the participation into its social network of rating books can provide an excellent recommendation service.

One of the most significant researches, which has helped to understand how small are the connections in our society, is the *small world problem* conducted by *Jeffrey Traver and Stanley Milgram* [TM69] [Mil67]. An experiment that formulates the following theory, it is possible that any two people selected arbitrarily from a large population can be connected to each other.

The procedure followed in order to prove this idea was to choose arbitrary individuals, "starting persons", and a target person from United States to attempt to generate an acquaintance chain among them. Each initial person was provided with documentation about the target and instructions to continue the chain, utilizing the mail as a transfer medium. That documentation could just be sent to a first name acquaintance of the sender. Moreover, each participant was asked to return information about himself and the person who was sent the documentation, allowing to create a track the process.

The result of the experiment proved that the average of linkage to generate an acquaintance chain among starting persons to the target is 6. Based on this study, the idea of *six degrees of separation* [Kle02] was generated. From the point of view of this project, the relevance of this experiment relies on the connection achieved with participants through the documentation provided.

Observing in detail, this documentation is an important factor that may affect to the experiment. Due to the importance of this information, which can determine if letters reach the target. It is fundamental for the study to provide appropriate information to help individuals to achieve sending the letter to a next linkage in the chain. Instead of sending it to an individual who has fewer relationship with the target than the current participant.

Nowadays, relationships among people have changed due to the appearance of more ways of communication through long distances. One of the mechanics to communicate with people is the mobile phone. A device which is becoming more and more popular, being used by a high percentage of the population

in the world. The main feature of the device is the mobility which allow to talk with another person and connect to the Internet anywhere at anytime. In addition, it converges several features that allow to measure location, activity and social interaction patterns. Therefore, taking as an inspiration the "Small World" experiment, about finding a target by providing certain documentation to an individual, and the use of the devices with the new features to measure the social behavior, the bases of the project have been established.

1.2 Thesis aim

The main goal of this project is to design and implement a tool to motivate the audience of Roskilde Festival, a large-scale event where there are more than hundred thousand participants, to move around the area of the festival in order to carry out a human mobility research .

The tool for this proposal is an application developed for Android mobile phones where one person has to find its target in the festival. This application is implemented to capture information of the user and its environment by utilizing the mobile devices sensors of GPS and Bluetooth.

This idea proposes several challenges about the design of the application and the issues of carrying out the experiment in an area like Roskilde Festival, where there are limitation of resources in connectivity and power supply. This document analyses thoroughly the several options to implement a game in order to motivate the participants of the festival to move and it pays special attention to the interface of the user aiming to create a dynamic and intuitive game.

The final aim of the thesis is to explain in detail the data obtained from the application and how this data can contribute to the study of the social behavior of the participant in the Roskilde Festival.

1.3 Thesis Structure

This thesis is composed of 8 chapters with 3 appendixes. A short description of each one is introduced on the following points:

Chapter 2 describes the proposal of the game and how the game outcomes can be applied in a social research.

Chapter 3 introduces the related work based on obtain data from Bluetooth and it explain the concept of gamification.

Chapter 4 analyse the requirements of Roskilde Hide and Seek, including a UML case diagram.

Chapter 5 introduces the design of Roskilde Hide ans Seek, including a system overview, the user interface and a sequence diagram.

Chapter 6 describes the implementation of Roskilde Hide and Seek.

Chapter 7 presents the results obtained after the data captured from the Roskilde Festival.

Chapter 8 discusses the main aspects of this thesis. It analyzes the results and it suggests some improvements for the system.

Chapter 9 conclude the thesis by explained the value of the research and the objective.

CHAPTER 2

Roskilde Hide and Seek

The goal of this thesis is the implementation and design of a method to motivate the Roskilde Festival crowd to walk around the festival area, where more than hundred thousand of participants are gathered each year. Roskilde Festival is considered as a cultural event where people share common tastes and live in the same place during the 8 days of the festival. However, the experience of the festival is better if it can be shared with more people, therefore, the idea of a game arise, merging the two proposals. This game will become the tool to motivate the festival crowd and it will also generate a chance to meet people in the festival, becoming a social experience for the participants of the game who want to extend their friends in Roskilde Festival.

This chapter presents Roskilde Hide and Seek, the initial part of the chapter explains the idea behind the game and how it is going to get to people. The last part presents the scientific goal of this project.

2.1 Game proposal

Roskilde Festival is one of the biggest festivals in Europe and every year more than one hundred thousands of people are gathered to share the experience of music and festival culture. This festival has a deep social spirit with a lot of

events during the festival week, where people carry out activities in group to create a social atmosphere. The opportunity of expanding this social atmosphere by implementing a game which introduces unknown persons fits with the spirit of the festival and it also fits with the main goal of the thesis, which is to encourage people to move around the festival area.

The game experience is based in the idea of knowing as much persons as possible in order to gain points. These points will be a way to obtain more information of the searched persons and they also represent the final score, which will be compared with the rest of the users of the game.

The phases of the game have been split up in two stages:

Collected Information stage: The first part is to introduce the user to the experience of Roskilde Hide and Seek by showing the information that it will share with the rest of players. This stage will explain what this game is about and how to fill out the profile to share the information.

Gaming stage: The main game starts and the target person is showed. The user has to follow the hints of the game to find the searched person. This search will have a limitation time, the more time is spent in the search, the less points are earned. A procedure of introducing each other will be carried out when the user finds the target and the points will be assigned after confirming the search.

2.2 Outcome

From the user perspective, it is a way to know a people in the festival and to be part of the Roskilde Festival community.

From the scientific point of view, the goal of this application is to capture data of users in order to carry out social researches like , for instance, a mobility research of the festival crowd. In addition, it is also added another question into the aim of the project:

- Can a game change the behavior of a audience of a festival?

The question introduces the term "Gamification", which will be explained in the next chapter.

Related work and experiment area

Bluetooth is a short-range communication technology between mobile devices that is basically used to share information. The way this communication works is by sending a broadcasting signal to other devices and returning a list of devices identified by their Bluetooth-MAC (unique identifier for each device). The range of this signal has a limitation distance of 10 m for Bluetooth class 2, the most common class for mobile devices which uses a maximum power of 2.5 mW, therefore, assuming that the GPS coordinates of the device that emits the signal by Bluetooth are known, all devices captured can be located with an accuracy of 10 m.

This technology has obtained an increasing attention recently. Several researches have been carried out using Bluetooth as a way to capture data and some of them are related to this project as they conducted out in a mass-event [VLBdWN09]. One of these researches, which is based on Bluetooth as a track technology, was carried out in 2011 in Roskilde Festival [oIatTUoD12]. This experiment developed a study about the human mobility based on the data obtained from 33 mobile devices strategically positioned in the area of the Festival. These devices were scanning during the 8 days of the festival obtaining data from Bluetooth probing.

In reference to this project, the method of obtaining data is similar. Instead of using static scanners located in strategic positions, it will be implemented an application for a mobile phone, which can convert each device into a scanner itself and therefore, it creates a distributed scanners to capture the information. The advantages of this proposal are detailed in the following points:

- The number of devices can increase due to participants can convert their mobile phones into scanner devices.
- Devices are not static, they move around the area of Roskilde as participants do.
- Devices are not limited to the Roskilde Festival area.
- It can generate a personal study of the user who has been using the application.

Apart of these advantages exist some disadvantages:

- The battery of the device is limited.
- The collaboration of users is needed.
- The internet connection necessary for the study depends on the access of the user to internet.

3.1 Gamification

Gamification is a term closely related to the project carried out in this thesis. One of the definitions of gamification says: "*It is the use of game design elements in non-game contexts*" [DDKN11] and it also introduces the concept of influencing the players to adopt certain conduct by following the rules of the game.

In regards to the thesis, the idea is to create a game to meet people inside the festival, by making the simple action of playing a game and earning points by knowing new persons. Therefore, the proposal of the concept of gamification in the thesis, and the creation of a game to influence players, raise another aim of the thesis:

- Is It possible to modify the behavior of players through a game and is it viable to obtain the data required?

3.2 Roskilde Festival

The experiment has been carried out in the Roskilde Festival, one of the biggest annual music festivals in Europe, which has around 130,000 participants. A festival with a huge camping area near to 1 km^2 , where the participants have access to different shops, restaurants, cultural zones and information zones. The other main area of the festival is the concerts site, which covers 0.2 km^2 and contains 6 stages distributed all around it. These stages are the main attraction of the festival and have different characteristics and sizes:

- Orange stage.
- Arena stage.
- Cosmopol stage.
- Odeon stage.
- Pavilion stage.
- Gloria stage.

The festival lasts 8 days, starting Saturday afternoon and finishing Sunday midnight. The first four days, which are called warming up, are a preparation of the festival for participants and most of the stages are closed, just the Pavilion Junior stage area is opened. After those 4 days, the concert site is opened and concerts start. Inside the stages area it is possible to find all kind of services for the public. The access to the concerts site is limited to the schedule of the bands, thus during some part of every day the area is closed.

In regards to the experiment, two of services provided by the Roskilde Festival organization have to be emphasized, as they help to carry out this experiment.

The charger stands is one of the services, which are located all around the camping area and some other places where it is also possible to charge the mobile phone, allowing participants to use their mobile phones during the whole 8 days of the festival and therefore, it allows the application to run during these 8 days.

The WiFi spots introduced in 2012 allow participants to have internet access by using those 12 WiFi spots, which are distributed into the festival area. Those two services make possible to provide a valuable attention to users of the application, in case they are out of battery or without internet connection. However, the concern of the battery and connectivity are still one of the major challenges of this project.

Analysis

This chapter is focused on the analysis of the requirements to implement the Roskilde Hide and Seek game. It starts by explaining the basic conditions of the application and it goes deeper in details about the different components in charge of the functionality of the system.

4.1 Requirements Analysis

The aim of this experiment is the study of the social behavior of participants in the Roskilde Festival. Summing up, the idea of the experiment is to create an application for Android devices where users have to find another persons in the area of Roskilde Festival by following the instructions of the application. Therefore, the analysis of the requirements have been divided in the following paragraphs:

First of all, the idea of finding someone presumes that it is necessary to know where the person is. Hence, the *GPS* [Devb] is a fundamental requirement in this project. The GPS proffers information of the location of the device by sending the latitude and longitude of it. However, the use of the GPS provides good accuracy but it drains the battery. A method to avoid the draining of

the battery is the use of internet connection obtain the position of the device, because through the WiFi signals and cell tower it is possible to calculate the location [Devb], although the accuracy is not as good as the accuracy of the GPS. Therefore, the combination of both methods can provide better results for the battery and the accuracy of the location.

The next requirement is Internet connection, a fundamental part of the project because it is the way to share the information from one devices to the rest of them by communicating with a server. However, it is noticed that participants of the Roskilde Festival have issues with the connectivity in the Festival area. Thus, the application has been developed to use internet at specific moments of the game, getting to play most time of the game without connectivity.

A server is the next requirement, a way of storage that can be accessed by several devices at once through the creation of a database and a socket of communication with the application. The information stored has a great relevance in the application, creating the links between the different users. This information has to be captured in the application by offering a form that the user needs to fill out in order to participate in the game. The questions of the form have a big importance in the search of the target.

Once the user has found his target, the search has to be confirmed. The use of the bluetooth scan of the devices allows to confirm this search. According to the characteristic of the bluetooth, the devices need to be in an area of 10 meters (Bluetooth transmitter class 2) to receive the signal from other devices. Taking advantage of this feature, it can be ensured that one user has found the target.

Moreover, in order to motivate the search in the game, a reward for finding someone is included as a requirement of the game. The motivation of the user is an important feature in a game of this type and, by including points, competition with other users is created, adding to the experience of the game the desire of overcome the rest of users or even the user himself.

Summarizing the last paragraphs into the following requirements:

- Use the *GPS* to obtain the position of the user.
- Confirm the search by using the Bluetooth when the user finds the target.
- Connect with the server by using internet connection.
- The game operates major of the time without using internet.
- Server manages the storage of the sent information.

- Form of questions to capture the information from the user (emphasize the relevance of this requirement to find the target).
- Update the location of the user automatically.
- Assign points to motivate the user each time it achieves an action (for instance, filling out the formulary).

4.2 Use case diagram

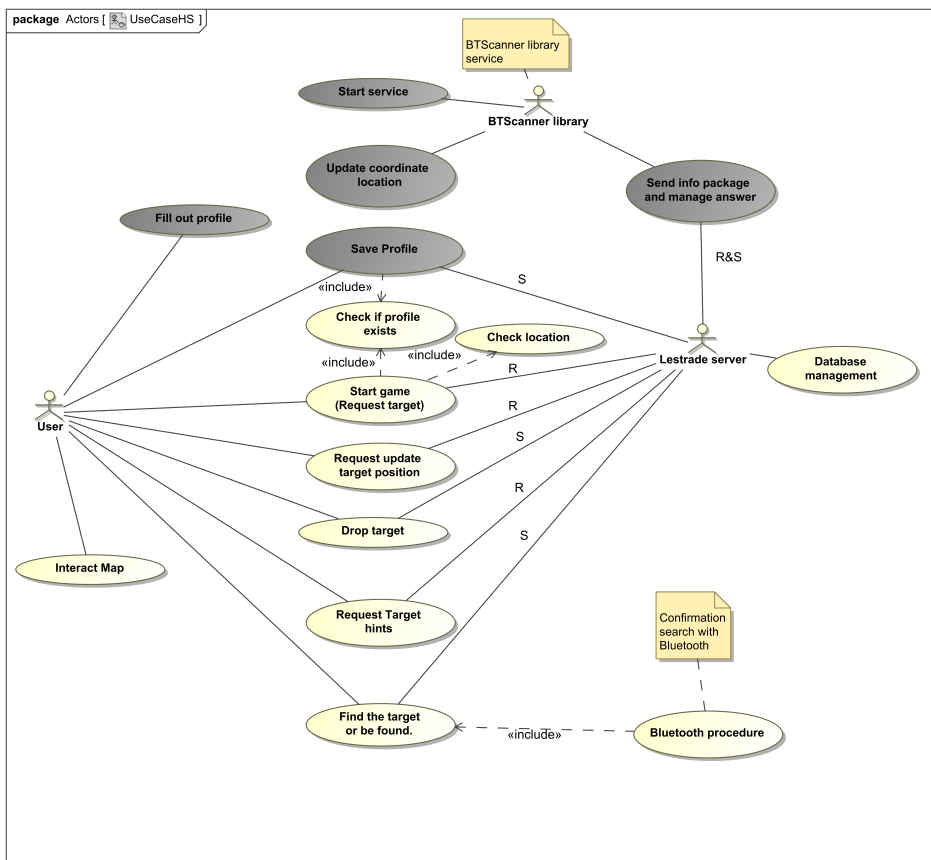


Figure 4.1: Use case diagram of Roskilde Hide and Seek application.

This section deals with the first approach to the system based on requirements exposed in the previous section. In the figure 4.1 it is shown an use case UML

diagram where it is possible to see the components of the system which interact with the three actors.

There are two separate components in the diagram that explain the interaction with the three actors. The first kind of components has a gray color and it refers to the collected information of the application. These modules get the data from the user and the device. The other kind of components are the yellow one. These components are in charge of the searching part of the game. When the game is initiated, they generate the environment to guide the user until its target. In addition, two kinds of components can be grouped depending on the communication with the server: requests for the server (R) or sending information (S).

-Send data to the server (S): These components gather information from the user or the devices and send it to the server to save that information in the database. For instance, if the user creates a profile in the application, the profile has to be stored in the server database and it will be shared to other user when it requests a target.

-Request to the server (R): Components that need information from the database of the server because certain data is required for the functionality of the application.

4.3 Case Scenario

Through the case scenario it is possible to obtain a deeper analysis and comprehension of the problem and an approximate idea of the contexts in which the application may be used. Different situations of the application, which the user can have, are explained in the following points.

Scenario 1: New user starts the application: The precondition of the scenario is the downloading and initiation of the application. The first impression of the application is important as a method to get the attention of the user and to gain its interest into the game, showing an introduction that explains what this application is about and how it is working. Furthermore, it allows the user to use some of the features of the given application.

Scenario 2: Create new profile The precondition for this scenario is the opening of the application and the user is in the area of Roskilde Festival. As the application is about finding another person in the area of Roskilde, the features of connectivity of the mobile phone has to be operative, allowing to

get the required information for the proper function of the game. The user can start creating a profile and saving it, and then starting the search of his target.

Scenario 3: Start game The precondition for this scenario is to have already created a profile and the user must be located in the Roskilde Festival area. The user performs a request to start the game, then the information of the target is sent. The content of this information gives hints to the user about where the target might be and how he looks like. As a way to handle the problem of the target location, it is utilized a map of Roskilde Festival with a mark on it, indicating where it is the target.

Scenario 4: Find the target The precondition for this scenario is to have a target assigned and to find it. In this case, the seeker and the target have to follow some steps in order to confirm the search. The clear explanation of this mechanism is fundamental for the game propose, hence a pop-up window is used as an extra help for the user.

This chapter is focused in the structure of the application Roskilde Hide and Seek by providing the design of the components of the application. It begins with an overview of the system and it goes deeper with the details of the structure of the different elements .

5.1 Overview of the System

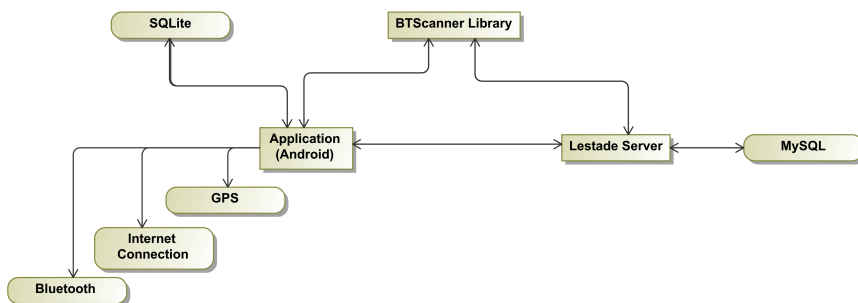


Figure 5.1: System overview of the application Roskilde Hide and Seek

The system is composed by three main modules that manage the information from the user. Each one operates in a different way and has different functionalities into the system. The interaction between them establishes the base of the application and the use of the rest of the components complement the whole system. As it can be seen in the figure 5.1, the modules are represented as a rectangle and the rest of the components as an oval. The relation between the main three components is bidirectional, meaning that the information is transferred in both direction.

The first module is the application itself, hide and seek. This module manages all the interactions with the user, therefore, it is where the user interface is developed. Due to the interaction and communication with the user are happening in this module, it can be considered as the main module of the system. In addition, the communication with the other two modules is managed by making requests or sending data.

The second module is the BTScanner library. This module provides the location of the device and the possibility of sending the information of the user when it does not have internet connection. This library is run independently of the Hide and Seek application, however the application module can request information to it or takes part in one of its services to send data to the server when there is not internet connection.

The third module is the "lestrade" server. Both modules, the first and the second one, have their information stored in this server. To connect with the server it is fundamental an internet connection.

5.2 Sequence Diagram

Figure 5.5 and 5.6 show the interaction processes of the components introduced in the Analysis chapter. Some of the components have to satisfy a precondition in order to perform the interaction, these components are represented as a rectangle composed of broken lines. The structure of the program and the sequence of interaction is explained below:

When the application starts, it is verified if the Bluetooth system is turned on in order to start the service of the BTScanner library, which captures the information from sensors of the device, turning the mobile devices into mobile scanners. This library manages the obtaining of the information from sensors by making requests each certain time, and then the application receives the information from sensors through a listener included in the library. By taking

advantage of the fact that the listener is called after certain time, the operation of updating the user information is also performed after this time. If there is internet connection, the information is sent immediately, however, if there is not internet connection, the information is sent by the BTScanner library to the server. This operation is repeated during the whole cycle of the application.

When the scanner and the application receiver are configured, the interaction with the main menu of the application, which presents 5 options, starts. The first option is the ranking of the game, where it requests to the server the 5 users with highest scores of the game. It also needs internet connection due to the interaction with the server.

Another option of the menu is the profile form, the user fills out a form with several questions and the application checks if the content introduced is correct. When all of the fields in the form are filled the profile can be saved. This operation needs internet connection and it checks again the profile to know if there is any empty field in the form. The user communicates to the application and the application sends the information to the server. A confirmation of the action is sent back.

The start button screen is another option that can be accessed in the main menu. When the user presses the start button it checks two things. First of all, if the location of the user is within the limits of the game (Roskilde Festival area). Second of all, if the profile was filled out and there are not empty fields on it. Then the application requests a target to the server, which manages the request by finding the target with the most recently updated location and it checks that the target has not been already searched by the user who requested. When the server sends back the target information, the application opens the Roskilde Festival map screen and shows the location of the target on it, therefore, the game begins.

A new part of the application is opened and several options are accessible. As the target has been assigned, the interaction in this stage of the application will be focused on getting more knowledge about the target, to drop that target or to confirm that it has been found.

The drop option allows the user to remove the assignment of the target. This action has to be notified to the server in order to allow the dropped target to be included into the list of available targets again.

To get more knowledge about the target, the application has a screen where it can be required information about it. Part of the information will be hidden and it will be necessary to pay points to have access.

Finally, the confirmation of the search is a process that uses the Bluetooth to accept the target as found. The seeker has to turn on the scanner of Bluetooth by pressing a button in the application and then the target has to turn on the discoverable mode of Bluetooth, by pressing the button for this purpose. When the seeker device identifies the Bluetooth by comparing the Bluetooth MAC of the target received with the target information, then the search will be confirmed.

5.3 User Interface

The user interface is an important aspect in a game. The position of buttons, the way that the menu is located and how the information is shown can help the user to get more interested in the game. The screen shots of the application can be observed in the Appendix A.

Starting with the first screen, the user can see the logo of the game, then, after a few seconds the screen changes into the info screen. This screen presents a list of rules and tips for the game, the purpose of this screen is to introduce the game and the different interface buttons.

-Initial screen: When the game is initialized a first screen with the icon of the game is presented. This screen is exposed for a few seconds and automatically pass to the next screen, either info screen, if it is the first time the user uses the application, or the start screen.

-Info screen: Overview of the content of the game and rules that the user has to follow to complete the tasks of the game. It is shown through a list text and with help icons for the interface of the game. This screen can be accessed after the initial screen or in the main menu of the application by using the tabs.

-Start screen: User starts to be aware about the game from this screen. A big start button in the screen and a several tabs give the impression of beginning the interaction with the application. This screen with its tabs menu at the bottom is the link to all the screens in the game.

-Be found screen and Identify Target screen: When an user finds its target then this screen is used. A few instructions to explain how can the search be confirmed are shown.

-Profile screen: To gather the information of the user, the profile screen reveals a form with several questions. Interacting with the menus of the form the corresponding answer for each field can be chosen. A detail of the interfaces used for each field can be seen in the implementation chapter. Finally, after filling out each field, the information can be saved and sent to the server.

-Ranking screen: As a motivational part of the game, the ranking screen shows the 5 users with highest scores.

-Target screen: A similar interface to the profile screen, but, in this case the information shown contains the target information. It has an interface to obtain more information by paying a certain amount of points.

-Map screen: A map of Roskilde Festival containing several options is utilized to help the user in the search of the target. This is another link screen, which communicates the stage of searching with the stage of starting the game. The map has an icon on it showing the position of the target. In addition, it has zoom in/out and drag features to make easier the movement through the map.

A connection flow of screens can be seen in the figure 5.2. The color of rectangles represents the different sections in the game. When the application is started the initial screen is pop up and it begins the first section, user offline.

Once the user create a profile and saved, then it can be started the second section, user on-line, where a target is assigned and the rest of screens game can be accessed.

In addition, as an extra help for the user it has been inserted in the game the pop up windows with some tips.

5.4 Profile Form

Due to the topic of the game is about finding someone, it is a fundamental requirement to create a profile of the user in order to share its information with the rest of participants. This information serves as hints to guide users to their

target.

The first approach to this topic was a form that would contain around 10 or 15 questions that the user has to fill out in order to increase the knowledge about it. These questions would be divided into two groups:

Basic questions: The information is not changed along festival days.

Relative questions: The information can be changed along festival days.

For instance questions proposed for this initial approach were:

- *What kind of music do you like?*

- *Where are you from?*

- *What do you do?*

- *What is the next band that you are going to see?*

This idea was ruled out due to the amount of time needed to write all answers and to avoid the use of the keyboard to fill them out. However, it is used a similar approach to this idea but making more dynamic the part of filling out the profile. In order to achieve the dynamism, it has been introduced selectable answers where the user has to choose one, or two, in some cases. This makes the profile form faster to fill out and to be updated.

The second part is the matter of asking the user the appropriate information. It is supposed that the target is probably unknown for the user, thus the questions to ask to the participants of the game are related to the physical aspect of them. Therefore, the first field of the form is a picture of the user, which might be obtained by using the digital camera of the device . Next things that could be interesting to know are the age, the height and its current location, these fields give a quick mental picture of the target and help to find the person.

Apart from the physical aspect, it is also interesting to know the clothes that it is wearing, allowing the target to be recognised by its clothes. The clothing category has been divided into 4 different fields:

- Shoes.

- Clothes over the belt.

- Clothes under the belt.
- Complement.

Summing up, the form is composed by the following fields:

- Nickname.
- Explain where are you.
- Height.
- Age.
- Shoes.
- Clothes over the belt.
- Clothes under the belt.
- Complement.

5.5 Points table of the game

The game has a system of points in order to motivate the user. In this section it is explain, through the use of a table 5.1, all of the actions of the game that will give points or will cost points to the user.

5.6 Server and Database

The storage of the information is an important aspect, this is a consequence of the application being mainly based on the data captured from the user and how to manage it. There are two database, which manage the information of the application and they are placed in different locations.

The first database is the internal one where the application saves the data while the game is in process and it contains two tables. The first one is the table

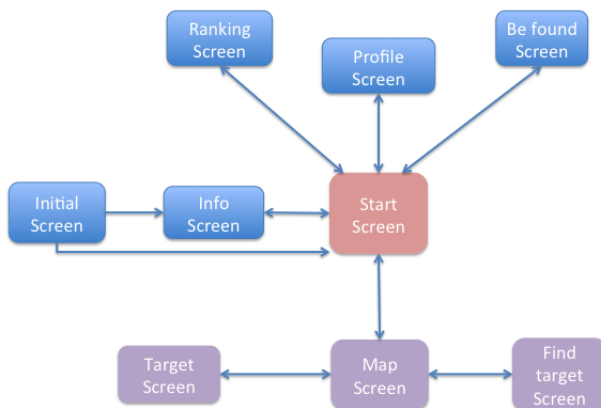


Figure 5.2: Connection between screens of the application

Table 5.1: Description of the actions to earn or lose points in the game.

Action	Points	Description
To create the profile and save it	50 points	Introduce all the profile fields correctly and save it
To edit some profile fields costs points	- 5 points	Once it is saved, if it modifies cost points
To start the game	5 points	Press the start button and wait for the confirmation to start the game
To drop target	-20 points (free after one hour)	Choose in the map menu the drop target option
To request a target hint	-5 points	To obtain more info of the target
To find a person in less than 30 minutes	40 points	
To find a person in less than 1 hour	30 points	
To find a person in less than 2 hours	20 points	

in charge of the user information, when the user fills out the profile all the information is saved into this table together with the points, latitude, longitude and the bluetooth mac.

The other table is the targets table, which saves the information of a target when it is requested by the user. This information is saved until the user finds the target or the target is dropped. Both tables have a similar content as they

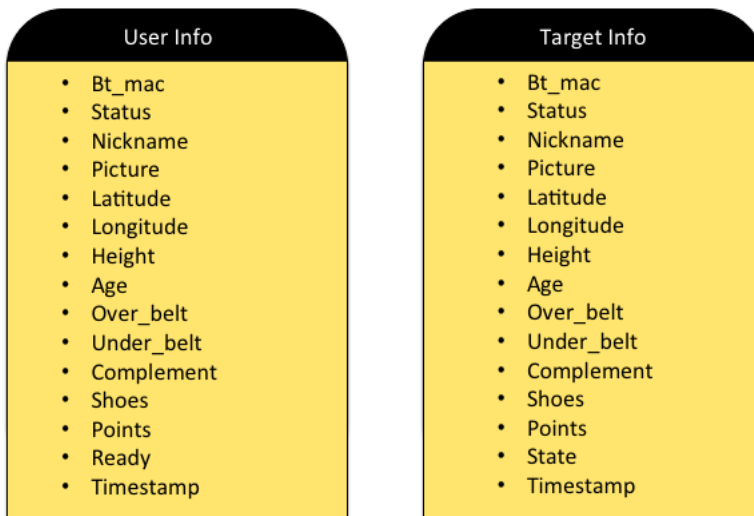


Figure 5.3: Diagram of internal database of the Roskilde Hide and Seek application

come from the same source of information. The figure 5.3 presents a diagram of the two tables, User info and Target info.

The second database to store the information is located in the "Lestrade" server. This database must be accessed through internet connection (HTTP POST requests) and the queries are managed by PHP scripts (insertion, selection and updating). The database has two tables that contain the information of the user and sensors of the device. The first one called "user" contains all the information of the user and some fields that manage the request of the target. The list of the fields is represented by figure 5.4

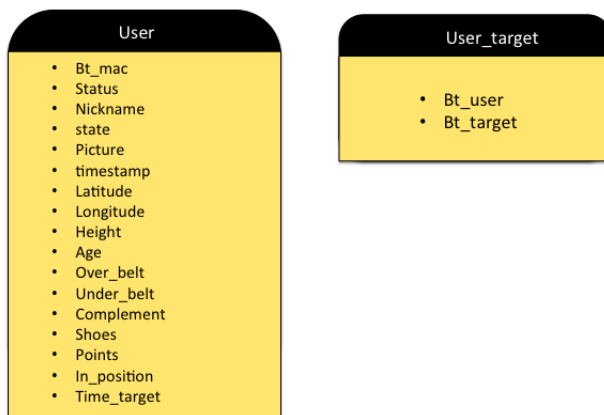


Figure 5.4: Diagram of lestrade database of the Roskilde Hide and Seek application

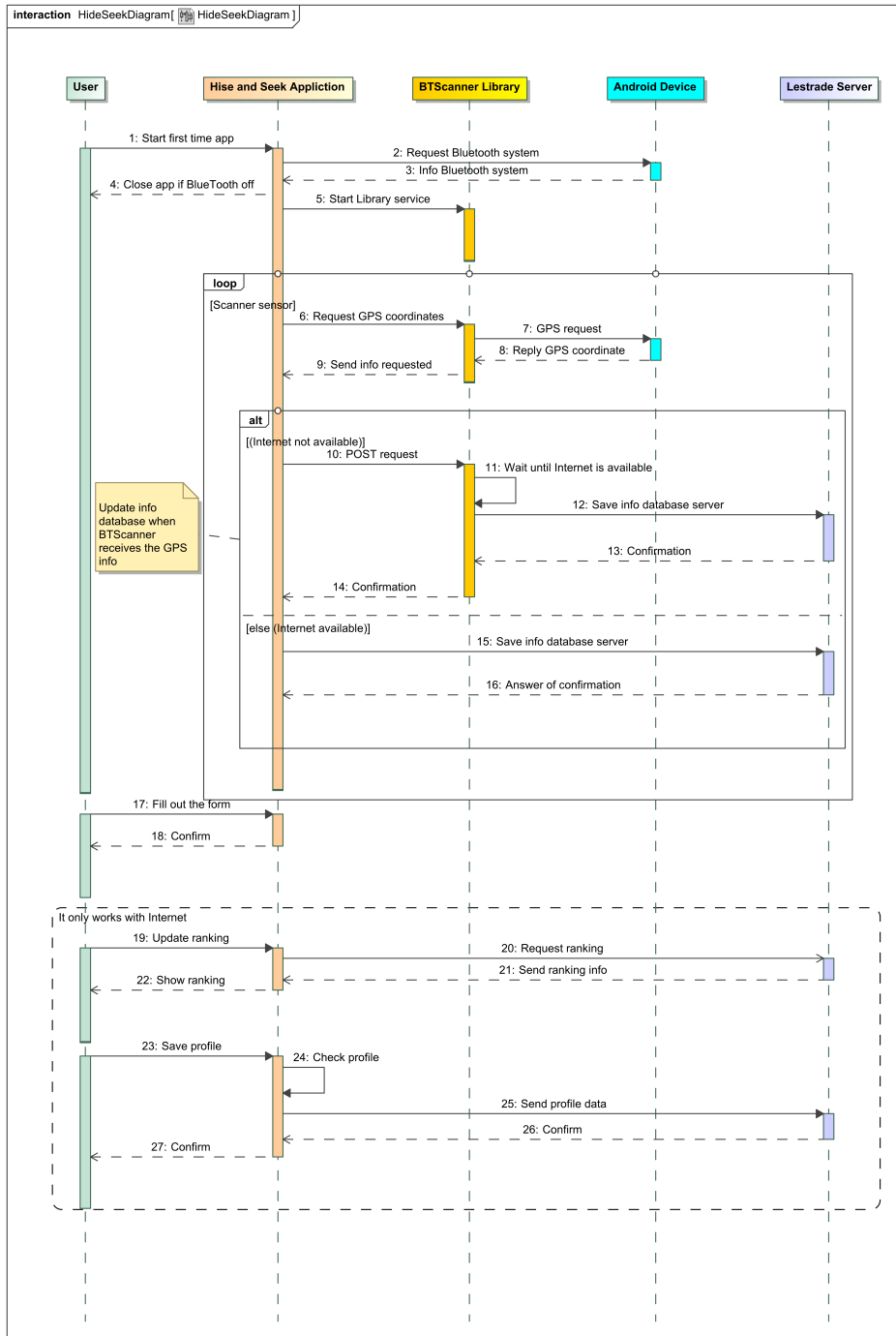


Figure 5.5: Sequence diagram showing interactions among components - 1/2.

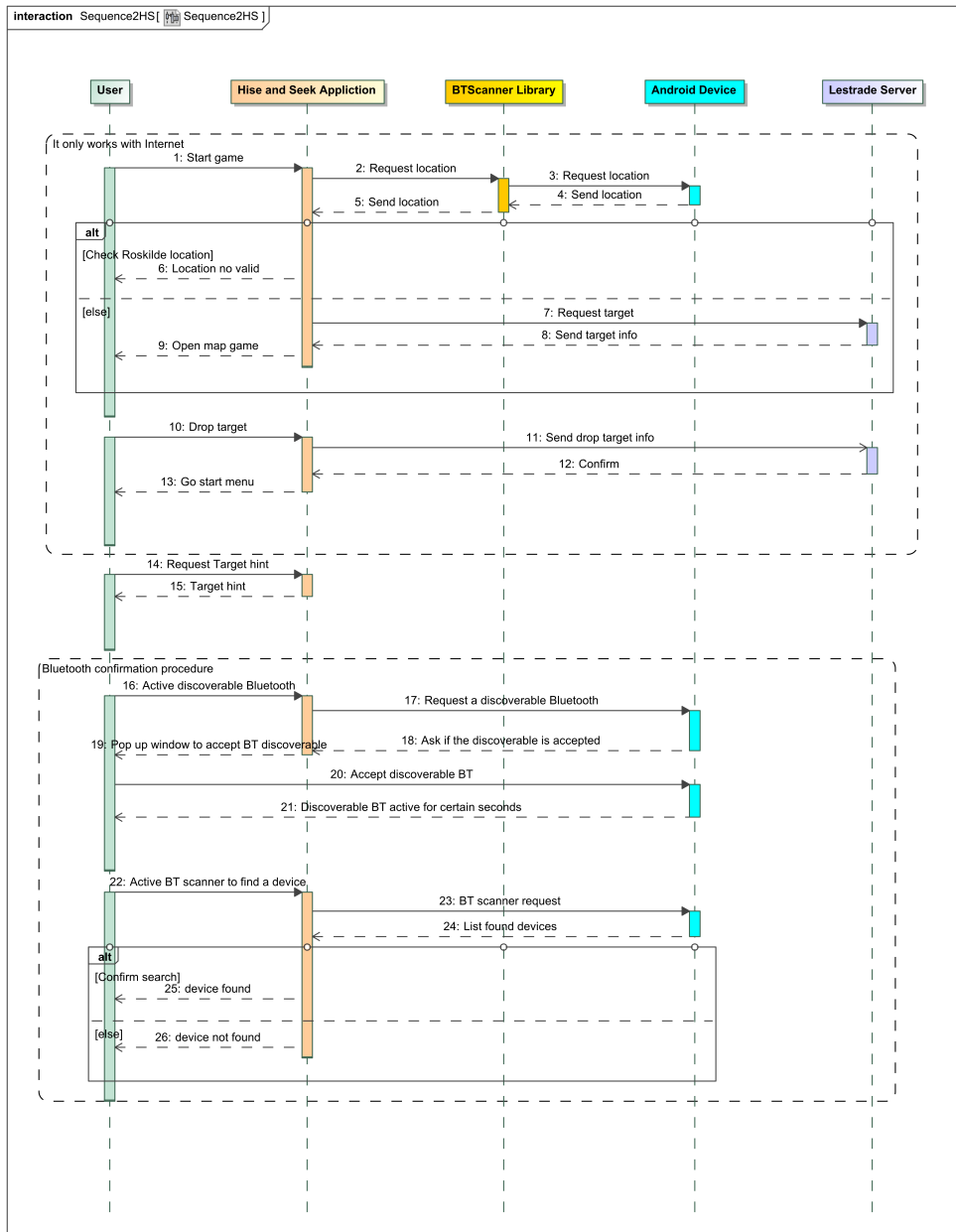


Figure 5.6: Sequence diagram showing interactions among components - 2/2.

CHAPTER 6

Implementation

The process of implementing the design explained in the previous chapter is carry out in this chapter. The design has presented that Roskilde Hide and Seek is divided in three main modules:

- BTScanner supporting library.
- Server lestrade and database.
- Android application game.

The following sections present the final solution of the application Roskilde Festival.

6.1 Overview

The platform utilised for carrying out the project is Android. This platform provides a wide source of resources for developing a mobile application by means of using Java programming language. One of the most suitable IDE for Android

is *Eclipse*, well known for its easiness to include new functionality, offering a great integration with the "Android development tools" (ADT) [Dev12a].

Apart from ADT of Android, it is also included into Eclipse the subversion to save the progress of the work, as well as, "PHP development tools" [Pos12] to edit PHP script and finally, "Remote system explorer" [Obe] to connect with the server through SSH and be able to modify PHP scripts in the server part.

Android was released in 2005 by Google and since then, the versions of Android have been improved until the current version Jelly Bean 4.1. However, most of users of Android have older versions and in order to cover as much devices as possible it has been developed the version 2.2, API 8, as a minimum version to run the application. [Dev12b]. Another factor to take into account, in the development of an Android application, is the variety of devices who use Android as an operating system. As a consequence of this, those devices have different features. An important feature to consider is the screen size and density, controversial characteristic due to the resolution of the elements in the interface depends on it.

The last module is the "Lestrade" server which manages all the data obtaining from the application. This server allows to use PHP scripts and contains a MySQL database.

6.2 Library BTScanner

The library *BTScanner* [FD12] has been development for the student Marcos Fuentes and Lukasz Dynowski, two DTU students who are collaborated with this project. This library offers several services, which run behind the main application, Roskilde Hide and Seek in this case. The services implemented on this library are:

- GPS service: it calls the GPS scan after certain time and spread the information to the listeners implemented in the foreground application. it has been made a modification to include the update of the position in the game Hide and Seek Roskilde. The library has a option to create a listener of this service and it will send the information back to the listener when the location manager of Android provides the location data.
- Bluetooth service: Provides the Bluetooth MAC and generates a scan after a variable time that is calculated based on the number of devices captured.

- **Battery service:** Obtains the information of the battery.
- **Server service:** Administers the sending of the information gathered in the application by generating a package and sending it when the internet connection is available.

This library provides an useful tool to gather information of the user through the mobile device, converting the device into a sensor which sends information to the "Lestrade" server periodically. The application Roskilde Hide and Seek is based on this library to obtain all the information of the device from its sensors (Bluetooth, GPS, WiFi, Battery sensor).

6.3 Android Interface

This section introduces the "User Interface" (UI) of Android and explains in detail some concepts of the implementation of the application.

Layout is the method adopted to set up the interface in Android. Through the definition of a layout the developer is able to control the components of the screen and the position of those elements. There are two ways to define Layout:

- **Declare in XML:** Android provides XML files to configure the views.
- **Instantiate layout elements at runtime:** Instead of declaring the layout in the XML, it can be inserted programmatically.

In addition, Android brings the option to create layout referencing to the type of resolution (Small, Medium and High). Both options, either how to define the layout or the resolution, have been utilized in this project and it will be explained in a future section.

Regarding the types of layout, this list wants to be an introduction of different layouts used in the project and it provides a short explanation of them:

- **Linear Layout:** As its name indicates, this layout arranges their elements in a single column or in a single row.
- **Relative Layout:** This layout allows the position of an element can be described in relation to other.

- **Table Layout:** Concerning to the position of elements by rows and columns.

Another element into the interface of the user, that is used in the implementation, is the view [Deva] . It represents the basic block for the interface and it is used to manage the interaction through text fields, Button, ImageView.

Layout and views contain several modifiable options in order to customize each part of the user interface. However, when the customization is not enough in some cases, it exists the option to extend the functionality of them in the runtime part programmatically.

6.4 Android application

The aim of this section is to explain in detail the development of the game Roskilde Hide and Seek. The game has been divided in several phases by following an order of execution of the application screens and the regular flow of the game in order to get a better understanding of the implementation. The Appendix B contains a class diagram which illustrates the connection of the classes explained in this section.

6.4.1 Informing phase

This phase of the application is in charge of informing the user about the game and it also starts the different services to gather the data of the device.

The first class to comment for this phase is the *RoskildeHideAndSeekActivity*. It is the initial class of the application and it checks if the Bluetooth is activated due to the *BTScanner* library needs to have the Bluetooth in order to work. This library schedules the calls of the different sensors of the mobile phone and manages the upload of the information obtained from the server. Once the services of *BTScanner* library are launched the mobile phone turns to a mobile scanner.

The second class is the *DescriptionActivity* which has two type of screens to explain the game, the one with rules and instructions show in text format and one with pictures of icons with a short description. Another method to help the user with the application is through the "Dialog" and Alert windows. These are pop up windows launched when the user performs certain action in the application. They can display either a error or a help message.

6.4.2 Collecting user data

Once the information is explained, it is necessary to collect the data from the user. *ProfileListActivity* is the main class in charge of this functionality and capture all the information required from the user by a form.

There are three kinds of fields in the form to obtain the user information:

1. Field to obtain a picture of the user: A *imageView* layout with a touchable option activated, allowing to open a pop up window by touching the *ImageView*. Then a "custom dialog" is created giving two options to select the picture ("Take from camera" or "Select from gallery"). Finally the picture is located in the space of the "*ImageView*". However, the picture obtained from the gallery or the camera are rectangular and the "*ImageView*" in the layout is a square, therefore, it is applied a crop action to remove part of the top and the bottom of the picture by using the "*Bitmap*" library of Android SDK which allows to manage the size of a picture.
2. Text box field: Several options of the form has a open answer, which means that it has to use the keyboard. Those fields are managed by a "custom dialog" that generated a pop up window with a text box. These "Dialog" windows contain a verification control about the character of the text, through regular expressions which only some characters are allowed, and a maximum number of characters. Once the user introduces a correct answer, it saves in the local variable of the field.
3. Selectable answer: The last type of field is the selectable answer implemented with the spinner option for a "custom dialog". It is included a list of elements by taking them from a file. Thus, the user has to choose one of those elements by touching it. Finally, the element chosen is saved in the local variable for the respective field.

By pressing the save button, the application checks if all the fields has been filled out correctly. An important issue is to control the uniqueness of the nickname because the application uses it as a reference of the game. After all the verifications, the user information is sent to the server and saved in the database. In addition, this process implements a GPS request to the *BTSscanner* to obtain the most updated location of the user. In the case that the device has the GPS sensor disabled, it is given the chance to turn it on by opening the GPS setting window. Although, the server has saved the user information, there is also an internal database for the user which is managed by the *UserInfo*. This class has several methods to operate with all the fields of the profile and

manage the save of all of them. The main reason is to have access to it without using internet and moreover, this database has faster accessed than the server database.

6.4.3 Starting the game

This phase represents the real game where a target is assigned and the user has to find it. The initial screen of this phase is the start screen managed by the class *StartScreen*. This screen contains a button with the word start on it. Once the button is pressed, several verification have carried out. First of all, it has to be checked if the profile is created, the user needs a profile before request a target.

Second of all, the user has to be into the limit of Roskilde Festival area by which the location coordinates of the user are checked.

The reason that the user has to accomplish both requirements is to confirm that it is an active participant of the game and therefore, it can request a target.

Once the requirements are confirmed, the application requests a target to the server. In order to obtain a target, a sentence for the database is developed. This sentence selects the most updated user by observing to the field "timestamp". it is also verified if the target is not already assigned and the user has not already tried to find it. The structure of the database is explained in the chapter of Design 5.6

```
SELECT * FROM user_table WHERE bt_mac != <user_btmac> AND state = <not assigned> AND in_position= <location_Roksilde_Festival_area> ORDER BY timestamp DESC
```

When the application receives the target information, it is ready to start the real game: find a person in Roskilde Festival.

6.4.4 Interactive Map

A map of Roskilde Festival area is shown as a first screen in the part of finding the target.

This screen is managed by two classes. The first class *MapViewActivity* administers the activity of the application and manages the view by calculating the size of the device screen. In addition, this class instances the custom "ImageView" which is controlled by the class *TouchImageView*.

TouchImageView implements its own thread to manage the changes applied to the image. This thread generates a loop where the image is refreshed. The

method *onDraw*, which contains the canvas of the "ImageView", draws all the elements in their respect position of the screen calculating where they have to be in pixels of the screen. This method is called by the thread and it changes the elements by means of the user interaction with the "ImageView" or based on the time of playing the game, it has been developed an element which represents the timer of the game.

Moreover, the "ImageView" has a method to control the touch event, hence, it can implement a button over the "ImageView" to open menus.

In addition, the "ImageView" contains the action of scaling and zooming the map by dragging the image. The calculation of the scale is carried out by an Android library "Scale Gesture Detector" [DevC] which controls the multi-touch event and returns the scale factor. Based on the scale factor, the Roskilde Festival map is modified by calculating the new screen points of the image in pixels of the screen.

Another point in the implementation of this method that has to be emphasized is the issue with the different sizes of screen of devices, this issue is managed by obtaining the size of the screen through the method "onMeasure" and based on the measure of its width and height, it can be calculated the ratio of screen and therefore, it can be positioned in the correct location all the elements.

To sum up, the custom "ImageView" can perform the following actions:

- Zoom in/out into different levels.
- Draggable picture around the picture.
- Touchable buttons on the picture.
- Timer of the game.

One of the main points to use the map is to show where is the target. As it was explained in previous paragraph, the custom "ImageView" can control what to draw over the Roskilde Festival map, so it can be marked the position of the target by adding an icon where the target is situated. First of all, to implement this idea it must be obtained an image of the Roskilde Festival map knowing exactly the number of pixels of the image and the GPS coordinates of the top right corner and the bottom left corner. Once it is obtained, it has to be converted the pixel of the image into the pixel of screen, calculating the conversion of the image pixel into the screen pixel in order to translate the GPS coordinates of the Image in pixels of the screen.

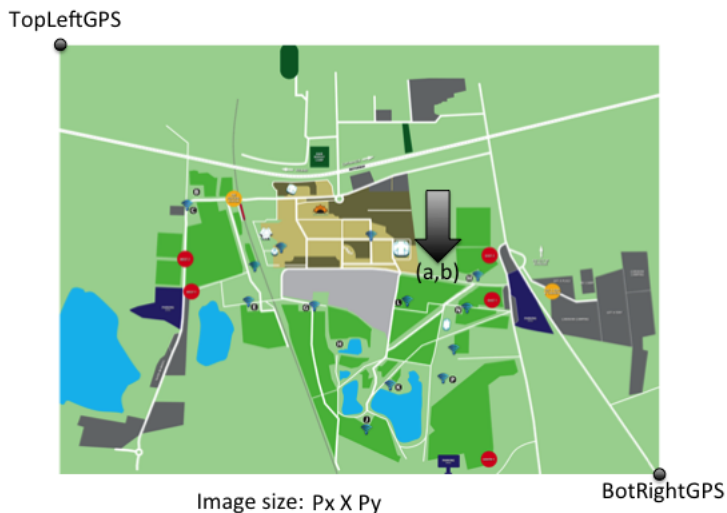


Figure 6.1: Roskilde Festival map sketch.

The figure 6.1 and the following equation shows the calculation to convert GPS coordinate into screen pixels :

$$\begin{aligned}
 a &= \left| \frac{\text{targetGPS.latitude} - \text{TopLeftGPS.latitude}}{\frac{|\text{TopLeftGPS.latitude} - \text{BotRightGPS.latitude}|}{P_y}} \right| \\
 b &= \left| \frac{\text{targetGPS.longitude} - \text{TopLeftGPS.longitude}}{\frac{|\text{TopLeftGPS.longitude} - \text{BotRightGPS.longitude}|}{P_x}} \right|
 \end{aligned} \tag{6.1}$$

6.4.5 Map menu

The interactive Roskilde Festival map of the application has integrated a couple of areas in the screen where it can generate an action by touching them. One of those areas control the update of the target location by connecting with the

server and obtaining the most updating coordinates of the target. When it returns the new location, the map is refreshed with the calculation of the new target position.

Another interesting area can be identified by a wrench icon. When this area is touched a pop up window appears and shows the menu of the map. This menu includes 5 options:

- **Resume:** Close the menu and go back to the map.
- **Target:** Open the target information screen which provides more info about the target. The information is showed with similar design as the user profile. However, there are some fields of information which are hide in order to make the user tries to find the target without any help. To avoid that the user gets stuck in the search and lost the motivation on the game, the application allows to show more information of the target. To do this, it must be paid 5 points of the overall punctuation.
- **Target Found:** Open the target identification screen where it has to be followed a procedure to confirm that it has been found the target.
- **Main Menu:** Back to the main menu of the application. It can be returned to the map and continue with the search.
- **Drop Target:** Communicate to the game that the target assigned wants to be rejected. The application requests to the server to change the state of the target as a "not assigned" and it also includes the target in the list of already assigned in order to not reassign the same target to the user.

6.4.6 Confirmation search procedure

The confirmation of the search is based of the Bluetooth technology due to limitation of the signal to 10 m and an unique MAC for each device which ensure that the user is unique and is close enough to know that the target has been found. Once the user finds the target, it must follow a procedure to confirm the search and earn the points. This procedure involves the two subject of the game, the seeker and the target.

First of all, the target has to activate its Bluetooth into a discoverable mode in order to let the seeker finds its device. The class of the target in charge of activate the discoverable mode of the Bluetooth is *CodeActivity*. This class implements a button to access to the Bluetooth adapter and allows to activate the discoverable mode during 120 seconds.

Once the target has its Bluetooth activated, the seeker can initiate the scanner of its device and confirm the search. The class *TargetInfoActivity* implements a method where the Bluetooth performs a scan and compares the target Bluetooth MAC, obtained in the phase of requesting the target, with the Bluetooth MAC captured in the scan. The BluetoothAdapter from Android creates a broadcasting signal which obtains all the devices in discoverable mode and it allows to get information of them, like Bluetooth MAC or the name of the device. If one of the Bluetooth MAC captures matches with the target MAC, the confirmation of the search in the part of the seeker is completed. The points earn for the search is calculated by taking a look to the time spent in finding the target.

In addition, the target needs to confirm that it has been found. The same procedure is carried out but in this case the target does not know the Bluetooth MAC of the seeker. Hence, the seeker, when it has confirmed the target, changes the ID name of the Bluetooth to the target Bluetooth MAC and activate the Bluetooth discoverable mode. The target performs the same procedure as the seeker before and implements a scan to find the ID name of Bluetooth with its same Bluetooth MAC. If the device is found, the search is confirmed and it is assigned the corresponding points.

6.5 Server side

"Lestrade" server contains the MySQL database where the information of the application is stored. The communication with the database is managed by PHP scripts which received the information from the application through POST request (HTTP). This type of request allows to send simple data as a text file but it also allows to send pictures. Both type of data are required for this application.

In order to send this request from the application, it has been used a library of Apache [asf12] which allows to use the http protocol in an easy way, and therefore send the information using http POST protocol programatically in Java. The class in the application in charge of this is *ServerRequest*. Basically, the class has some methods that generate the POST request with the information introduced through method parameters.

In the server part, all the information is received by the *index.php* and it distributes to method which performs queries to the database by obtaining a variable from the post called "method".

The figure 6.2 provides an example of one row of the database which was filled

out for a user.

```
<bt_mac>XX:XX:XX:XX:XX:XX</bt_mac>~  
<nickname>Tman</nickname>~  
<status>At my tent</status>~  
<state>0</state>~  
<timestamp>2012-07-05 16:03:54</timestamp>~  
<latitude>55.62390967</latitude>~  
<longitude>12.07713583</longitude>~  
<height>186</height>~  
<age>33</age>~  
<shoes>Sandals+Brown</shoes>~  
<overbelt>Shirt+Green</overbelt>~  
<underbelt>Shorts+Black</underbelt>~  
<complement>Hat</complement>~  
<picture>XXXXXXXXXX|.png</picture>~  
<in_position>1</in_position>~  
<points>45</points>~  
<time_target>2012-07-06 09:56:04</time_target>
```

Figure 6.2: Sample of user information in the database.

Evaluation

This chapter presents the results obtained since the release of the application and, particularly, the results obtained at Roskilde Festival. Those results have been divided into three different parts. The first part comprises the analysis of the application, from the point of view of data obtained from Google Play. Two other parts are focused in the data captured from the sensors of the device. The first one is more related to the Bluetooth and GPS data oriented to the social research and the second one is more concentrated in the explanation of the results obtained from the game experience. An overview of the results will be discussed in the chapter 8.

7.1 Android application results

The application was released the 17th of June [Rub12], two weeks before starting Roskilde Festival. It had got 117 downloads at the beginning of the festival and at the end of the festival it had reached the amount of 155. The number of active devices using the application was 107. The figure 7.1 shows a graph illustrating days when users have installed the application. The significance of the information showed by the figure is the amount of installations obtained the previous days to the festival. This data provides an overview of when an application for a festival like Roskilde Festival has to be released.

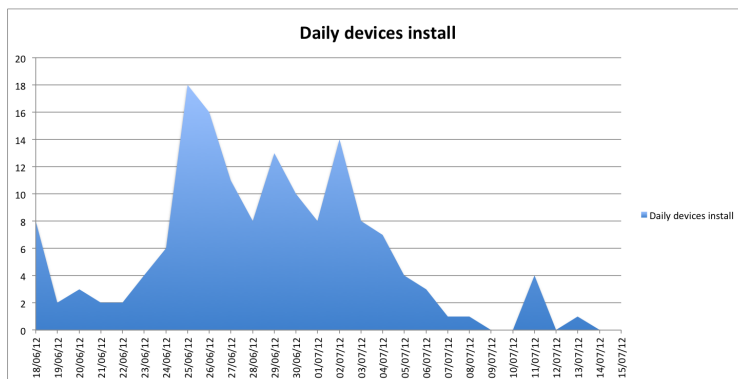


Figure 7.1: Devices that have installed the Roskilde Hide and Seek application

From the point of view of the implementation of an application, Google Play provides data about the most used Android version and the model of the devices. About the Android version, it has been obtained a total of **46.6%** devices with Android 2.3.3 - 2.3.7 and **42.2%** devices with Android 4.0.3 - 4.0.4, by knowing this information it is possible to get a better planning for the characteristics of an application. About the model of devices, the most used device for this application is the **Samsung Galaxy S2** with a **38%**. The rest of models have a low percentage of use compared to Samsung Galaxy, being **HTC** the most common brand, with models like **HTC Desire S**, **HTC Incredible S** and **HTC Desire** with **6.6%** each one. These devices have some common features, like the screen, which has a resolution of 800x480. The knowledge of this information provides a chance to work in a better design of the user interface.

7.2 Distributed scanners data

This section explores the results of the application as a distributed scanner by showing the information captured from the mobile devices. The information presented has been obtained since the release of the application (17th June) until the end of the festival (6th July). However, the last two days of the festival have presented a low rate of information and they have not been included in the figures.

The number of active users who have been uploading information from their devices has been 64. This users have provided information of GPS coordinates, Bluetooth, Battery level and WiFi information, that have been obtained through

the *BTSscanner* library. By focusing in the Bluetooth data obtained the number of observations has been 10300, including a total of 1350 unique devices discovered.

In the case of the festival area, the number of active user is 25. The figure ?? presents the number of devices captured during the festival period. This chart provides an interesting insight about when the application has been more used by observing the number of devices captured per day. The highest numbers of devices discovered are produced in the warming up days and they are reduced when the concerts start.

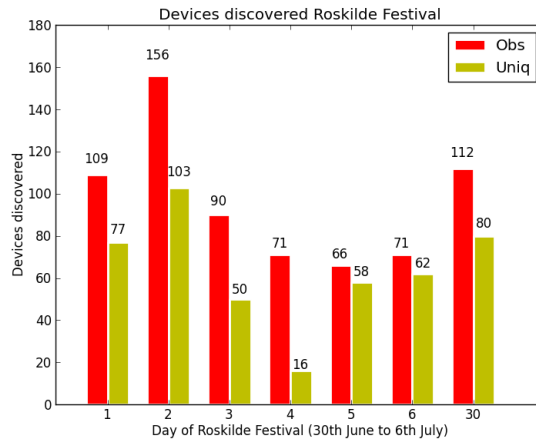


Figure 7.2: The number of captured devices in the festival.

In addition, the application has also recorded positions of participants by obtaining the GPS coordinates from their devices. The figure 7.3 illustrates the positions of 25 users of the Roskilde Hide and Seek application. The information showed was captured during the period since the 30th of June until the 6th of July. This information suggested the main points where the users have been located during the festival. It is possible to identify the camping area where they were established and it is also interesting to see how the movement through the concerts area starts in the days 5th and 6th by showing points concentrated around stages. The process to obtain the figure 7.3 has been explained in the appendix C.

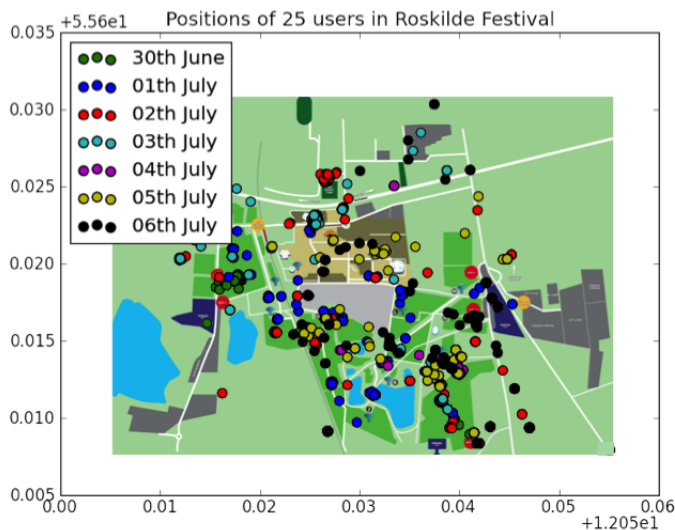


Figure 7.3: Track the users of the application in the festival area.

7.3 Game data

This section is focused in the information obtained from the game experience and results of players performance in the game. First of all, it has to be explained that there is a difference between users of the application and players of the application. A user is a person who installs the application and runs it, but it does not create a profile for the game, therefore, it sends information of its mobile sensors but it does not share personal information. On the other hand, a player is a person who installs, opens and creates a profile, sharing the information required in the game.

The GPS and Bluetooth data of the contribution of 35 players of the game is shown in the table 7.1. The first column of the table represents the number of scans conducted for the application in each device of the user. The second and third column represent the number of devices discovered with the Bluetooth, being the second column the total of all devices discovered for each device of the user and the third column the number of unique devices. The mark "X" means that those players did not contribute with data from Bluetooth or GPS.

The number of players in the game is 35 and the number of participants in Roskilde Festival is 130.000 which means that the application has been used

Table 7.1: Data obtained from 35 players in the Roskilde Festival 2012.

Users	BT xObs	BT Uniq	GPS scans	Users	BT Obs	BT Uniq	GPS scan
User 1	X	X	X	User 18	X	X	X
User 2	X	X	7	User 19	X	X	1
User 3	X	X	7	User 20	X	X	X
User 4	64	60	66	User 21	X	X	X
User 5	X	X	7	User 22	X	X	X
User 6	X	X	88	User 23	X	X	4
User 7	X	X	227	User 24	X	X	282
User 8	28	28	26	User 25	X	X	7
User 9	X	X	X	User 26	X	X	24
User 10	X	X	X	User 27	X	X	X
User 11	X	X	3	User 28	X	X	7
User 12	2	2	27	User 29	X	X	52
User 13	X	X	55	User 30	38	9	60
User 14	148	89	193	User 31	98	82	27
User 15	X	X	7	User 32	X	X	X
User 16	57	39	58	User 33	72	49	36
User 17	X	X	3	User 34	X	X	X
				User 35	X	X	X

for less than 0.01% of the participants in the festival. The total number of Bluetooth observations are 505 and these included a total of 328 unique devices discovered for the players of the game, which is the 0.4% of the participants of Roskilde Festival. Furthermore, there are 10 out of the 35 players which have not contributed with any data of their GPS and Bluetooth sensors.

The figure 7.4 presents the graph of the players who have played the game per day and it reaffirms what it was commented on the previous sections, that the application has been more used in the warming up days that in the concerts days. Apart from the information obtained from the mobile devices sensors, it has been collected personal information of users. The players of the game had to filled out a form about themselves when they created their profile. This provides another source of knowledge from the Roskilde Festival participants. As an example, the table 7.3 presents the average age and the average height of players of the application, which can be applied for a study about the crowd of the festival. Clothes, shoes and complements of the players can be also obtained from the data collected in the profile.

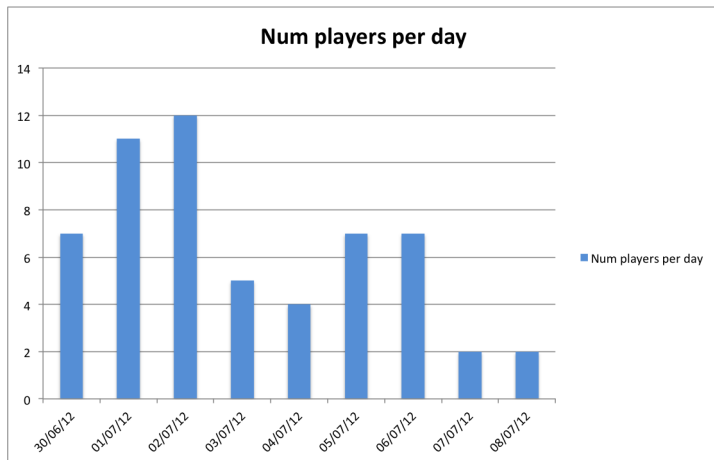


Figure 7.4: Number of players per day in the Roskilde Festival.

Table 7.2: Sample of users personal information collected by the application.

	Average	Max	min
Age	24	40	18
Heigh (cm)	180	205	162

7.3.1 Find a target

The main objective of the game is the act of meeting the person assigned. Players registered during the festival were 35, from which 16 of them have tried to find their target by making the action of pressing the button to start the game. However, these users have not been able to find any of their targets. The main causes for this are exposed in the following points:

- 1. Users have not updated their profiles.
- 2. GPS sensor is deactivated in most of the users (7.1).
- 3. Internet connectivity of Roskilde Festival presents a delay to update the data.
- 4. Low involvement into the game from the users.

These causes have reflected on the gameplay of the application and an example of two players is shown, for a better understanding of them. The two users selected for this example are the user 4 and the user 6, due to they have provided enough information from their devices to track their movements .

The first explanation is related to the update of the profile. The table 7.3 presents the information introduced by the two users of this example. The first register of these users in the application is shown in the column of the table "Last update time". As it can be seen both have registered the profile the 5th of July, however, they have played the 6th of July and they have not updated their profile. What this example reflects is that users did not update the profile as it was required in the game, which causes a lower probability of finding a person due to the target information is out of date. This situation has been contemplated with the rest of the users, where most of them did not upload the profile in the whole time of the game.

A point to emphasize before explaining the example of the track of two users is the requesting of the target. As it was explained in the implementation chapter 6.4.3, the way the application selects a target is by checking the most current location of all users in the application. If this location has not been updated recently, it would be provided a target which has an old location.

The figure 7.5 and the figure 7.6 present the track of users 6 and 4 during two days of the festival. Based on the user 4, who has tried to find the user 6. The request of the target has been performed the day *06/07/2012* at 08:00 and it was assigned the user with the most updated location among all the users, user 6, in this case. However, the user 6 has sent the last update of its location at 17:30 of the day *05/07/2012*, which means that the location is not updated and the user 6 would probably have a different location. This issue can explain two of the causes exposed previously. In addition, these two figures contain an interesting observation of the involvement of users in the game. The user 4 is searching the user 6 and the game provided to the user 4 the last known location of the user 6, which is the location of the day 5th of July that is represented with red points in the figure. If the user 6 would have started to find the user 6, then it would have moved to the area of the red points of the user 6 figure. However, the user 4 stayed in the south-east area instead of moving to the user 6 location as it is represented by the blue points in the figure of the user 4. This example of the user make a reference to the involvement of the user in the game.

The main issues presented in this section have occurred with most users of the application. In the chapter 8 is discussed the probably reason of the causes explained.

Table 7.3: Personal information of users 4 and 6.

	Age	Height	Over the belt	Under the belt	Shoes	Complement	Last update time
User 4	22	184	White sneakers	Green t-shirt	Blue shorts	None	2012-07-05 22:42:45
User 6	33	186	Brown sandals	Green shirt	Black shorts	Hat	2012-07-05 16:03:54

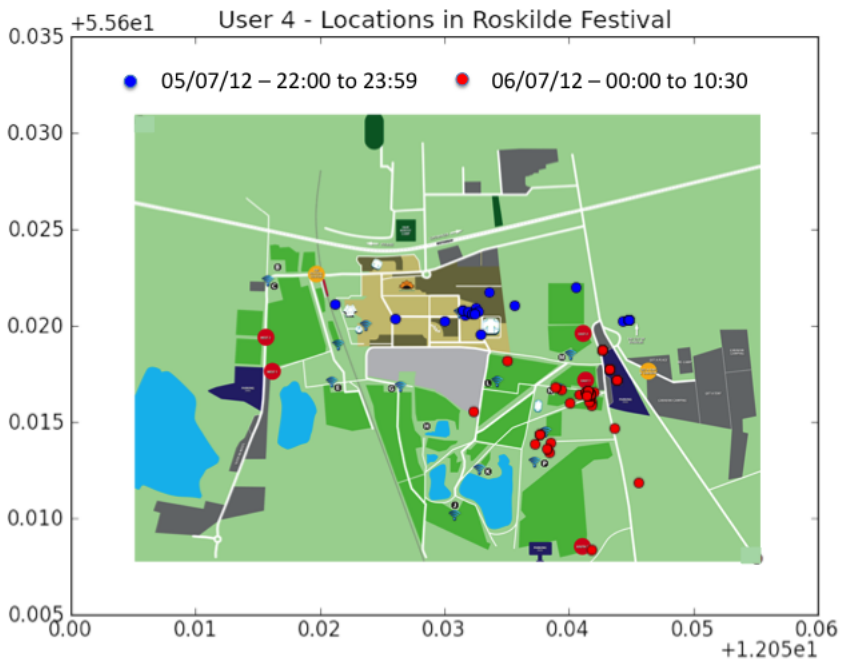


Figure 7.5: Spatio-temporal locations of the user 4 in Roskilde Festival

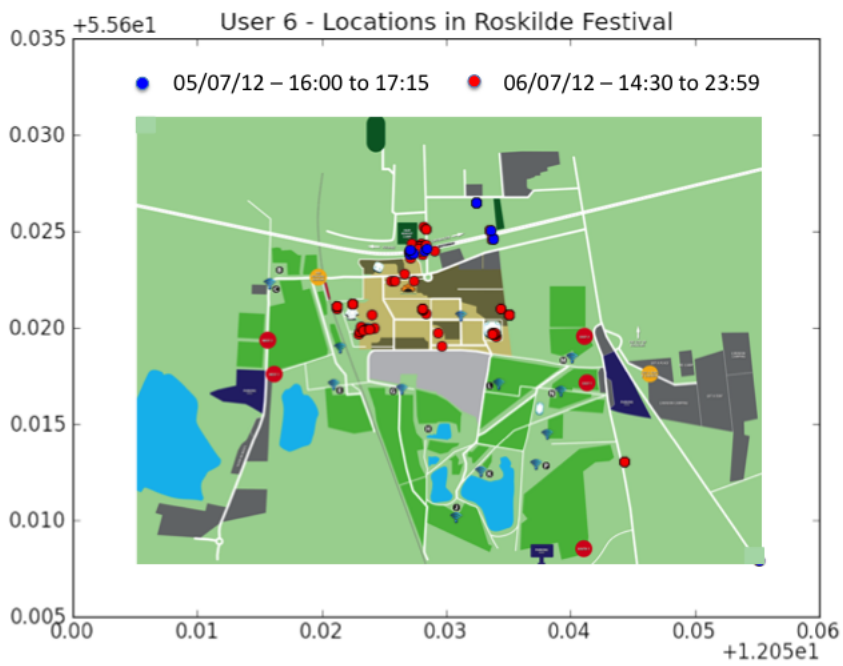


Figure 7.6: Spatio-temporal locations of the user 6 in Roskilde Festival

Discussion

The aim of this thesis is the design and implementation of a tool with two functionalities, to find a person in Roskilde Festival and to influence the Roskilde Festival crowd, by using a social game called Roskilde Hide and Seek. The way the game has been implemented has been explained in the early part of the thesis and the results obtained by following the method developed have been analysed in the previous chapter 7. The interpretations of the results and a deeper explanation of all the advantages of the game and the disadvantages are presented in this chapter and it also includes suggestions for the application and proposals for future work.

8.1 What has been accomplished

The main challenge of this thesis was the development of a game which accomplish the functionalities above explained, getting as result the game Roskilde Hide and Seek. This application suggests a way to generate a social game based on the information acquired from the user and the GPS sensor of a mobile device. The complete functionality and all the requirements requested have been obtained by developing an application based on Android operative systems. Furthermore, the game has included the library *BTScanner* to convert the device into a mobile scanner, this library has accomplished to obtain the data from the

user during the festival period and out of that time, while the game was running in the foreground.

However, when the game has been utilized in Roskilde Festival area, it could not provide the expected results due to players of the game could not find any target. The analysis of this matter was introduced in the section 7.3.1 and it is extended in the following sections.

8.2 Concept of the game

Roskilde Hide and seek is based on a social game which means that it is necessary the continuum collaboration of players in order to obtain a good game experience. The results presented in the previous chapter have reflected that the game has been able to obtain data of the user anywhere of the Roskilde Festival area, at anytime. Therefore, if it is supposed that the connectivity has worked and the data of the device has been obtained based on the results acquired, then the main matter in the application has been related to the participation in the game. The involvement of players could be determined by several causes, which are analysed in the following sections:

8.2.1 Entertainment of the game

The game arises a proposal to the user of finding a person in the Roskilde Festival, which has obtained great acceptance due to there have been 155 downloads. However, once the game started the collaboration of the users have decreased by acquiring 35 players of the game among the persons who downloaded it. The most probable reason, which is the focus of this section, is that the game does not get the enough attention from the players. When the game was designed, the punctuation system was included in order to allow the user to compete with the rest of the players, getting the attention of the user to make it play everyday. Nevertheless, this concept did not get the result expected.

The improvement of the system of rewards in the application can encourage the users to play the game. The following points offer some suggestions of rewards:

- To obtain a cold drink after finding the target.
- To create a badgets system that provides certain sense of achievement each time the user makes an action in the game (for instance, "the second

update of the profile")

- To generate different levels, where the difficulty of finding a target changes depending on the number of targets found.

8.2.2 Distance to find the target

Roskilde Festival covers an enormous area, which is divided in smaller zones to camp and the area of the concert. This game was designed to use the entire area of the festival without paying attention to the different zones. The consequence is that the user, when is trying to find a target, has to cover a large distance to find it. As it was observed on the example provided in the section 7.3.1, the user avoided to start the search and the distance could have been a possible cause.

A solution for this issue may be to change the assignation system of the target, by considering the location of the target and the user. It can be defined a range of search distance, for instance the camping areas, and therefore, the user would not be moved more than certain distance to find the target.

8.2.3 Bluetooth and GPS in a festival

The application designed is based on the information captured from the sensors of the device. These sensors are managed by the library *BTSscanner*, which runs the scans of the sensors either if the game is in the foreground or in the background of the Android operative system. The consume of battery of these sensors is high and for this reason, it was designed to make scans after certain time. However, the mobile devices are an important item in a festival for its functionality as a phone, for this reason, the user can consider to avoid an application which drains the battery of its device. For this case, probably one of the main reasons for the user to reject to play with the application is because of the continued use of sensors like Bluetooth and GPS.

8.3 Gamification

This thesis is based on a mobility research of the participants of Roskilde Festival and from this idea, it was designed a game to influence the users to move around of Roskilde Festival area. Therefore, the game inducts the user to move in order

to make actions to gain points but the reality is that the user is guided in order to obtain more data for the mobility research.

After observing the results 7.3.1, although, it cannot be affirmed that the user has been influenced to move due to the information captured does not provided any evidence that any user has tried to move to the target location.

8.4 Distributed scanner

Another aim of this thesis is the social research that can be carried out from captured data from the sensors of the device. The data obtained has been introduced in the chapter 7 by showing the possibility of measuring the movement of users and the number of devices discovered by the Bluetooth. However, if data obtained is compared with the experiment carried out with fixed scanners [oIatTUoD12], it can be observed that the amount of information gathered is rather low. The main reason is that the fixed scanners were controlled but the distributed scanners cannot be controlled due to the user controls them. As it was observed in the table of the players of the game, most of them had not been contributing with data from the scanners. That is the most important cause for which the distributed scanners can be considered inefficient for a social research, the low involvement of users who control them. One way to get the information necessary for a social research is by using a large number of distributed scanners, which means that it is required a large amount of users willing to participate in the research. This can be achieved by creating an application, like the one created, that provides an entertainment while the information is gathered, however, Roskilde Hide and seek has not acquired the enough users.

8.5 Future work

Roskilde Hide and Seek has proposed a game to find a person in Roskilde Festival. This game has been developed following a specific method to find a person, explained throughout this thesis, which in some aspects has to be improved. The first step in the further work should be to implement all the parts introduced in previous sections, which will get a better performance for the application in Roskilde Festival and, therefore, more data for the social research. One of the aspects that can be emphasized is the profile menu in the application, the information gathered in this section of the application can be modified or extended to make a deeper research about the type of user who attends to Roskilde Festival.

For instance, based on the information obtained from the user in the application, it can be performed a research about the type of clothes more chosen by the players, which may give a clue about what kind of clothes the merchandising stands need. Moreover, this information can personalise the study of the mobility in the festival by classifying the crowd into groups depending on the age, height or music taste, for example.

Roskilde Hide and Seek has provided a way to obtain information from the user, this method could also be extended through the connection with social networks like Facebook. For instance, if the user allows the application to have access to its Facebook graph, the relationships of friends in Facebook can be achieved, thus, by having the graph of all users with the application installed, connections among them may be established by searching for common friends, generating a services to recommend common friends between users and obtaining more information for the social research.

Conclusion

The proposal that has been arisen in this thesis is based on the wide spread of mobile devices in the society and the opportunity of carrying out a social research in a crowded environment like Roskilde Festival. The main goal of the thesis is to design and implement a game for Android operative system, which consists on the search of a person in the festival area. The intention of developing a game has two reasons:

- To influence the crowd of Roskilde Festival to walk around the festival area.
- To convert the device into a distributed scanner to capture data from users.

The biggest challenge in this project has been to design all the aspect of the game and to implement a functional application for Roskilde Festival. The research carried out has generated the application Roskilde Hide and Seek as a response, therefore, the objective of creating a game has been accomplished. Although the application was designed to find a person, the data captured from Roskilde Festival can not confirm that users have been influenced to move and, therefore, the objective cannot be consider as an accomplishment. Causes for this have been discussed in chapters 7 and 8. However, the application has

served as a platform to obtain data from the user. The data obtained after the festival has proved that it is possible to track the movement of an user and to discover Bluetooth sensors around the devices running the application, which demonstrates the possibility of achieving distributed scanners by using mobile devices.

APPENDIX A

Screenshots of the application Roskilde Hide and Seek

This appendix presents the screen captures of the application Roskilde Hide and Seek. Each of the figures are included with a short description of the role they play in the game.



Figure A.1: Initial screen.

The initial screen of the application **A.1** controlled by a *ImageView* layout. It is displayed the logo of the application and it is checked if the Bluetooth sensor is activated.

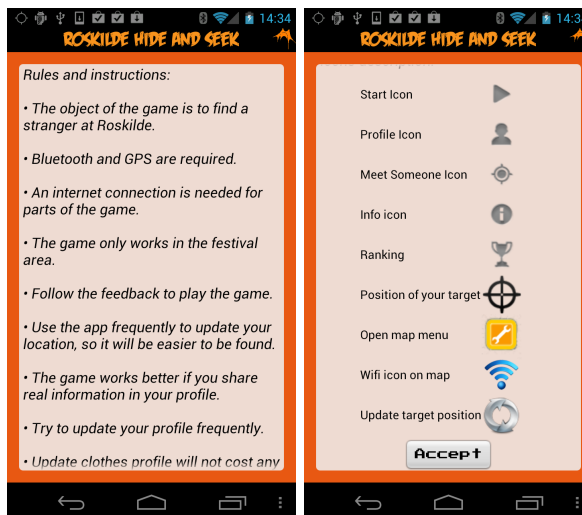


Figure A.2: Description screens.

The description screen **A.2** is in charge of the explanation of the game for the user. There are two screen, the first one is a list of string manage by a

RelativeLayout and the second one is a list of icons of the application manages by a *TableLayout*.



Figure A.3: Start menu of the application.

The start menu is shown in the figure **A.3**. It is the screen where the game can be started and it also the first screen shows to the user, where starts the interaction with the application.

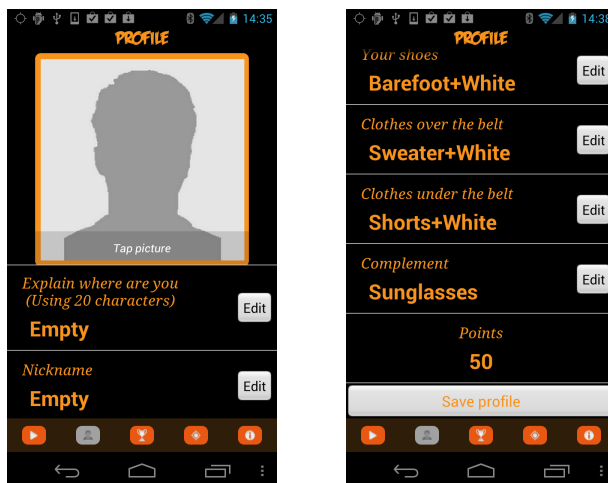


Figure A.4: Profile screens.

The profile screen, showed in the above figure, is in charge of gathering the information of the user and it manages by means of the combination of *RelativeLayout* and pop-up windows *Dialog*. In this screen the user is asked to fill out a form and to save the information.



Figure A.5: Map screen with the target marked.

The map screen is accessed once the profile has been created. The figure A.5 shows how is the map of Roskilde with the mark of the target and it is demonstrated the features of zooming in/out and dragging. This features are managed by a *ImageView* customized, which control the elements showed in the screen.



Figure A.6: Map menu and option of dropping the target.

The map contains a menu to interact with the target. The figure A.6 presents the options of the menu inside of a *AlertDialog*. It is also shown the info message of the application, in this case applied to the drop selection in the map menu.

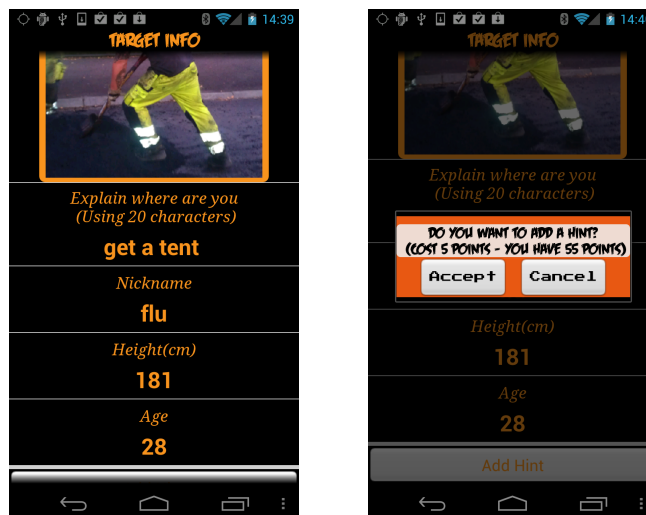


Figure A.7: Target information and request of a hint.

One of the options of the menu is the target information showed in the figure **A.7**. The menu is similar to the profile menu, however there are fields which are hide. In order to get access to the hide information, the user has to pay a certain amount of points, as it is shown on the right picture of the figure.

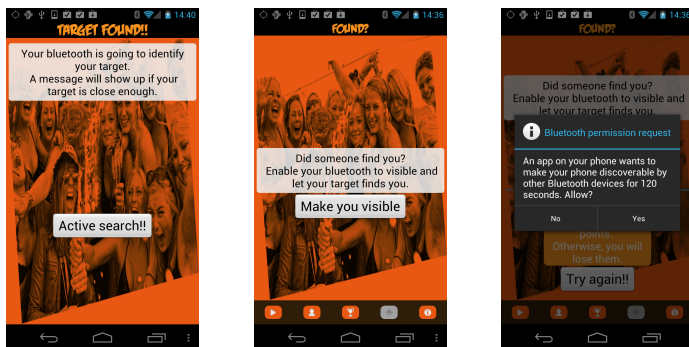


Figure A.8: Screens to confirm the search.

Finally, when a user finds its target, the three pictures **A.8** are shown. Depending on the role of the use, if it is a seeker (right picture) or a target (middle and left picture), will use one type of the screen or the other. These screen manages the sensor Bluetooth in order to confirm the search, as it can be seen on the right picture the request to the user for activating the discoverable mode.

APPENDIX B

Class diagram

The next sections present an overview of the system through a class diagram based on two packages, the main package where the activities of the system are saved and the utility package where support classes are saved.

B.1 Hide and seek package

The figure B.1 contains a class diagram where the connections of classes of the system with the ActivityPlus can be seen. The ActivityPlus is a class which extend the basic Activity provided from Android, but it also includes more functionalities. The activities of this diagram collect all the functionality of the application.

B.2 Hide and seek utility

The figure B.2 presents the classes with support some functionalities of the activities, which control the flow of execution of the system.

Position graph method

The figure showed in the chapter ?? has been obtained through a python script called "positionTrack.py". This script needs a csv file with three columns (latitude, longitude, date) as an input parameter. Once the csv is parsed, the data can be modified to include the two points of reference that get to fit the plot into the Roskilde map. These two points are the coordinates points of the top-left corner and the bottom-right corner of the picture. When all the data is saved in the appropriate format, then the methods from the "matplotlib" library are called and the plot generated is saved with a transparent background. Finally, the plot is fit into the Roskilde map picture by using an image edition software, given as a result the figure 7.3. The rest of figures presented in the chapter have been obtained by performing the same process.

C.1 postitionTrack.py

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.lines as lines
import matplotlib.transforms as mtransforms
import matplotlib.text as mtext
from datetime import datetime
top_left_longitude = 12.0555495768
```

```

top_left_latitude = 55.6306074533
bot_right_longitude = 12.1050896234
bot_right_latitude = 55.6079238251

reader = open("bt_scans.csv","rb")
bt = {}
i = 0
for item in reader:
    #print 'Row: ' + str(index + 1)
    if i == 0:
        i = 1
        continue
    row = item.split(';')
    try:
        bt[row[2].strip('\"')] ['sender'].append(row[0])
        bt[row[2].strip('\"')] ['mac'].append(row[1])
    except:
        bt[row[2].strip('\"')] = {'sender': [row[0]], 'mac': [row[1]]}

reader.close()

reader = open("locations_scans.csv","rb")
items = []
for item in reader:
    #print 'Row: ' + str(index + 1)
    items.append(item)
reader.close()
lt = {}
users = []
i = 0
for item in items[0].split('\r'):
    if i == 0:
        i = 1
        continue
    row = item.split(';')
    lat = list(row[1].replace(".", ""))
    log = list(row[2].replace(".", ""))
    lat.insert(2, '.')
    log.insert(2, '.')
    lat = ".".join(lat)
    log = ".".join(log)
    if (float(lat) < bot_right_latitude) | (float(lat) > top_left_latitude)
    | (float(log) < top_left_longitude) | (float(log) > bot_right_longitude):
        print 'not use'
    else:
        users.append(row[0])
        try:
            lt[row[5].strip('\"')] ['sender'].append(row[0])
            lt[row[5].strip('\"')] ['lat'].append(float(lat))
            lt[row[5].strip('\"')] ['long'].append(float(log))
        except:
            lt[row[5].strip('\"')] = {'sender': [row[0]], 'lat': [float(lat)], 'long': [float(log)]}

reader.close()

print len(set(users))
#draw a chart of devices discovered
def bar_chart_bt():
    obs = {}
    for t in bt:
        aux = datetime.strptime(t, '%Y-%m-%d %H:%M:%S')
        try:
            obs[aux.day].extend(bt[t] ['mac'])
        except:
            obs[aux.day] = bt[t] ['mac']

```

```

obslist = []
unqlist = []
for t in obs:
    print t
    obslist.append(len(obs[t]))
    unqlist.append(len(set(obs[t])))

N = len(obs.keys())
ind = np.arange(N) # the x locations for the groups
width = 0.35 # the width of the bars
fig = plt.figure()
ax = fig.add_subplot(111)
rects1 = ax.bar(ind, obslist, width, color='r', ec="w", lw=2)

rects2 = ax.bar(ind+width, unqlist, width, color='y', ec="w", lw=2)

## Try

ax.set_xlim(ind[0]-0.5, ind[-1]+1.5)
ax.set_ylim(0, 180)

# add some
ax.set_ylabel('Devices discovered')
ax.set_xlabel('Day of Roskilde Festival (30th June to 6th July)')
ax.set_title('Devices discovered Roskilde Festival')
ax.set_xticks(ind+width)
ax.set_xticklabels(obs.keys())

ax.legend( (rects1[0],rects2[0]), ('Obs','Uniq') )

def autolabel(rects):
    # attach some text labels
    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x()+rect.get_width()/2., 1.05*height, '%d'%int(height),
                ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)
plt.show()

## Volume BT in location
def location_chart(day):
    result={}
    for i in lt:
        aux = datetime.strptime(i,'%d/%m/%y %H:%M')

        #resultLat.extend(lt[i]['lat'])
        #resultLog.extend(lt[i]['long'])
        try:
            result[aux.day]['lat'].extend(lt[i]['lat'])
            result[aux.day]['long'].extend(lt[i]['long'])
        except:
            result[aux.day]={'lat':lt[i]['lat'],'long':lt[i]['long']}

    for key in result:
        result[key]['lat'].insert(0,top_left_latitude)
        result[key]['long'].insert(0,top_left_longitude)
        result[key]['lat'].insert(len(result[key]['lat']),bot_right_latitude)
        result[key]['long'].insert(len(result[key]['long']),bot_right_longitude)

matplotlib.rcParams['axes.unicode_minus'] = False
fig = plt.figure()
ax = fig.add_subplot(111)
plt.subplots_adjust(bottom = 0.1)
ax.set_title('Positions of 25 users in Roskilde Festival')

```

```
plt.scatter(result[30]['long'],result[30]['lat'],marker='o',c='g',s=50,label='30th June')
plt.scatter(result[1]['long'],result[1]['lat'],marker='o',c='b',s=50,label='01th July')
plt.scatter(result[2]['long'],result[2]['lat'],marker='o',c='r',s=50,label='02th July')
plt.scatter(result[3]['long'],result[3]['lat'],marker='o',c='c',s=50,label='03th July')
plt.scatter(result[4]['long'],result[4]['lat'],marker='o',c='m',s=50,label='04th July')
plt.scatter(result[5]['long'],result[5]['lat'],marker='o',c='y',s=50,label='05th July')
plt.scatter(result[6]['long'],result[6]['lat'],marker='o',c='k',s=50,label='06th July')
#plt.scatter(result[1],result[0],marker = 'o', c = color,s=50)
plt.legend(loc="upper left")
plt.show()
fig.savefig('location_scans_legend.png', transparent=True)

location_chart([30,1,2,3,4,5,6])
```

Bibliography

- [asf12] The apache software foundation. Apache httpcomponents. <http://hc.apache.org/>, 2012.
- [DDKN11] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek '11, pages 9–15, New York, NY, USA, 2011. ACM.
- [Deva] Android Developers. Android views. <http://developer.android.com/reference/android/view/View.html>.
- [Devb] Android Developers. Location strategies. <http://developer.android.com/guide/topics/location/strategies.html>.
- [Devc] Android Developers. Scale gesture detector. <http://developer.android.com/reference/android/view/ScaleGestureDetector.html>.
- [Dev12a] Android Developers. Android development tools. <http://developer.android.com/tools/sdk/eclipse-adt.html>, August 2012.
- [Dev12b] Android Developers. Platform versions. <http://developer.android.com/about/dashboards/index.html>, August 2012.
- [FD12] Marcos Fuente and Lukasz Dynowski. Exploring spatio-temporal data from distributed Bluetooth scanning. *Department of Informatics and Mathematical Modelling*, 2012.

- [Fou] The Eclipse Foundation. Eclipse. <http://www.eclipse.org/org/>.
- [Fre04] Linton C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.
- [Kle02] Judith Kleinfeld. Could It Be A Big World After All? The "Six Degrees of Separation" Myth. *Society*, 2002.
- [Mil67] S. Milgram. The small world problem. *Psychology Today*, 61:60–67, 1967.
- [Obe] Martin Oberhuber. Remote sytem explorer. <http://tmober.blogspot.dk/2006/11/remote-system-explorer-10-is-released.html>.
- [oIatTUoD12] Department of Informatics and Mathematical Modelling at the Technical University of Denmark. Crowds, bluetooth and rocknroll: Understanding music festival paticipant behavior. 2012.
- [Pos12] Jacek Pospychala. Php development tools. <http://www.eclipse.org/projects/project.php?id=tools.pdt>, June 2012.
- [Rub12] Felix Rubio. Roskilde hise and seek. https://play.google.com/store/apps/details?id=dk.dtu.imm.hideandseek&feature=search_result#?t=W251bGwsMSwxLDEsImRrLmR0dS5pbW0uaGlkZWZuZHNlZWsiXQ.., 2012.
- [TM69] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.
- [VLBdWN09] Delafontaine M Van Londersele B and Van de Weghe N. Bluetooth tracking - a spy in your pocket. *GIM International:23-25*, abs/1111.4570, 2009.
- [WF94] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Number 8 in Structural analysis in the social sciences. Cambridge University Press, 1 edition, 1994.