2. Introduction

The aim of this report is to describe the requirements of the “Health@Hand” project of group “BUTA”. In this report, the field research regarding the project is explained, detailed use cases and requirements are described and the road map of the project is presented.

2.1. Background Information

Nowadays, mobile phones are the most widely used communication tools. Almost everyone has mobile phone or even phones. Therefore, using mobile phones just for communication is a kind of narrow usage. Together with the rapidly developing mobile industry, mobile device producers offer lots of beneficial alternatives to their customers. Therefore, today’s mobile phones are much more than just a communication device. They are now used as entertainment centers (music players, TVs, game players, web surfers, etc.), communications hubs (email, text, MMS, the web), general purpose computers (calculators, organizers, spreadsheets, etc.). Even they can analyze blood to detect HIV, Malaria and more.

Mobile phones become a part of our life. Most people cannot go somewhere without taking their phones with them. They almost live together with their phones. Therefore we could take advantage of this situation to ease our life more. Especially it can help people living in rural areas where it is harder to meet basic human needs such as health care.

2.2. Project Purpose

Health@Hand is a mobile application which enables users to

- See their health log
- Get an appointment from doctors
• See doctor or hospital places in Google Map view
• Electronic Prescription (e-prescription)

2.3. Scope (Project Summary)

Health@Hand will be Java mobile application and to develop it, Turkcell API, Google Map API, Java mobile SDK will be used.

Health@Hand will enable users to get health care services easier. Health@Hand takes the doctor or hospital information from its database and enable user to see doctor or hospital schedule. Moreover, people can see the places of doctor or hospital from the Google Map view. Therefore using Health@Hand application, people can find the closest available doctor by saving their time. They no more need to go to hospital to see whether the doctor is available or not. Furthermore, Health@Hand database keeps the users health log also. Hence users or doctors can know patient’s earlier health information. Besides on these feature, Health@Hand enables electronic prescription (e-prescription). Hence, users no need to go to doctor and wait in the patient queue to take their prescription and then go to pharmacy. Doctor sends prescription by examining the patient health logs and user can take the prescription via Health@Hand. Moreover, user can confirm to buy medicine electronically.

2.4. Team Info

Our team is composed of four Computer Engineering students. The members of the team are Hüseyin Dirican, Ugur Cebeci, Yusuf Dönmez and Hilal Baydarov which aim to create an impressive application that is used by lots of people.

2.4.1. Structure

Our team has “Democratic Decentralized (DD) "structure. We do not have a permanent team leader. Every decision is made by team consensus.

Ground rules of our team are:
• Every Friday a meeting must be held and every one must be on time.
• If somebody cannot come to the meeting, he must have an excuse.
• The given responsibilities in ex-meeting date should be completed until the next meeting date.
• Every team member should check his mail daily.
• Emergency meetings are done according to team consensus.

2.4.2. Member Roles
• Hilal Baydarov -> devil’s advocate
• Ugur Cebeci -> recorder, timekeeper
• Hüseyin Dirican -> devil’s advocate, facilitator
• Yusuf Dönmez -> gatekeeper, timekeeper

3. Research

3.1. Marketing Research

In this section, currently available products in the mobile health care market that are close to our project are explained, and further discussion about their negative and positive points.

3.1.1. medline

Medline is a health-care centre which has online services. It serves almost all medical branches at home or hospital.

Medline has telephone line and also web site to be reach by people. They can send doctor and any needed instrument that can be transportable to home. People can be done mini-checkups in their homes. Medline has heath packets to customers which include emergency health services, telephone medical advice, location determination, pharmaceutical service, visiting the doctor in house etc.

Medline targets the Turkish people and it is only accessible via telephone line. It doesn’t even provide an interface for mobile devices. And it is available just in Ankara and Istanbul not rural areas.
“Health@Hand” provides seeing all contracted health centers in costumers’ town on Google Map wherever they are. They can make an appointment without waiting busy phone lines and people who can be bored phone calls in all-day.

3.1.2. istanbuleah.gov.tr

Istanbul Educational and Research Hospital official site has online appointment opportunity that rare in Turkey. Hospital is in list of most known in Turkey and Istanbul with huge number of doctors and patient capacity. It has a success and greeting since their online service.

However it also doesn’t provide an interface for mobile device users like medline.com.tr. Costumers can reach only make an appointment at this hospital.

“Health@Hand” provides mobile user interface and much hospital option according to user's location.

3.1.3. sislietfal.gov.tr

Another application similar to “Heath@Hand” is used at Şişli Educational and Research Hospital official web site. This site differs from istanbuleah.gov.tr by providing patients laboratory results via online. The web site also provides online appointments for their hospital only.

Our project gives opportunity to users to see their laboratory results, prescriptions, making online appointment. The most important difference of “Health@Hand” is again its mobile device interface.

3.2. Technology Research

3.2.1. Development Tools

3.2.1.1 Android

Android is a software platform and operating system for mobile devices, based on the Linux kernel, and developed by Google and later the Open Handset Alliance. It allows the developers to write managed code in the Java language, controlling the phone via Google-developed Java libraries.
Android is an open source project and new technology, but for entrepreneurs in Turkey like BUTA is not suitable because of lack of enough Android users.

3.2.1.2 Symbian

“Symbian is an operating system (OS) designed for mobile devices and smart phones, with associated libraries, user interface, frameworks and reference implementations of common tools, developed by Symbian Ltd.”

Despite the fact that symbian is the most popular operating system among mobile industry, it’s stated nearly at all web sites about mobile development that Sybian Application Development is difficult. Also, “Health@Hand” depends on Google Maps and since there is no native Google Maps API, Symbian OS is not the best choice.

3.2.1.3 Windows Mobile

“The Windows Mobile platform is an open platform that support needs beyond mobile messaging. It’s based on Microsoft .NET, giving developer freedom to innovate.”

Since Windows mobile APIs and development tools are not open source and since they are not free to use for commercial purposes, it can limit the “Health@Hand” project’s future and this makes Windows Mobile a bad choice. Additionally since BUTA Inc members do not have experience on Windows development environments, selection of more familiar environments is more suitable.

3.2.1.4 iPhone

“iPhone OS or OS X iPhone is the operating system developed by Apple Inc. for the iPhone and iPod touch.”

iPhone has an APP Store where developer can sell their products. This can be a good opportunity for companies like BUTA Inc., which have never sold mobile products.

Although iPhone OS offers a good API and development environment for developers, the biggest drawback of iPhone OS is that it lacks both Java and Flash
support. This can be a problem for future steps of our project. Hence, development of first mobile client side implementation of “Health@Hand” will not be on iPhone OS.

3.2.2. Software Researches

3.2.2.1 Php

PHP is very common in today’s web market. Most of the web applications today are developed with PHP and its various frameworks.

Advantages:

- PHP is very popular along web developer
- Easy to code
- Easy to find code samples on web, because PHP is open source

Disadvantages:

- PHP seems to be popular among hackers and hobbyists, which is a bad reputation
- PHP has a confusing and inconsistent class/object system

3.2.2.2 Java

Java is another option for server-side development.

Advantages:

- It is platform independent
- Leads ‘enterprise’ tools and in general has better support for web services and other connectivity mechanisms.
- Maturated and advanced in last 3 years
- Easy to detect errors while coding
- Open source

Disadvantages:

- Edit / compile / loop timeframe in Java tends to be slower
- Java is complex
- Big apps eat huge amounts of memory

### 3.2.2.3 Database Management

The main part of our project is Database Management. The time table of each doctor and hospital are stored in our database. We will use MySQL in web service and JDBC in our mobile application.

- JDBC
- MySQL

**JDBC**

**Advantages:**

- Open Source
- Simple

It is preferred to use JDBC when using Java technologies.

**MySQL**

**Advantages:**

- Is not always free but very cheap
- Has pure performance due to its default table format MyISAM
- Binding with too many programming languages

Listing all pros and cons of these 2 database servers, we decided to use MySQL due to its compatibility with Java and much more advantages over other servers.

### 3.2.2.4 Netbeans IDE

Using Netbeans we can develop some languages. Java is one of these languages with these IDE it possible to run application as in the real phone. Connection speed and other features can be adjusted as real phone. Using Netbeans it is possible to use Turkcell’s API. Netbeans support the JME and it is known that programming in Java is convenient in Netbeans.
3.2.2.5 Google Maps

Google Maps (for a time named Google Local) is a basic web mapping service application and technology provided by Google, free (for non-commercial use), that powers many map-based services, including the Google Maps website, Google Ride Finder, Google Transit, and maps embedded on third-party websites via the Google Maps API. It offers street maps, a route planner for traveling by foot, car, or public transport and an urban business locator for numerous countries around the world. According to one of its creators (Lars Rasmussen), Google Maps is "a way of organizing the world's information geographically".

4. Requirements

4.1. Functional Requirements

4.1.1 Data Flow and Interfaces

The data flow within the system and the external interfaces are illustrated in the data flow diagrams below. As it can be seen from the diagrams, the application is mainly interacts with user as a mobile application. The system also gets location information from mobile devices. This data can be gathered by GPS or other location services provided by mobile phone operators. The google API will be used to get this information from mobile device. Other system services like, Internet communication and map services are also provided. The system also communicates with other social network applications with their web services. This communication will be handled by the server.
4.1.1.1 Level 0

Diagram:

- **Google Map**
  - User Location info

- **MOSHEALTH**
  - User Location info
  - Places On Map
  - User Command And Data

- **Hospitals**
  - Appointment Data
  - User Request

- **Turkcell**
  - User GSM Info

- **User**
4.1.1.2. DFD Level 1 MosHealth System

MosHealth System

Appointment Data

Process Appointment Data

Hospitals
Members

Get User Places Info

Process Command
Collect Places

User Location Info

Places On Map

User Request

User Command And Data
4.1.1.3 DFD Level 2 Process Command

- Process Command
- Hospitals
- Members User Id
- Parse Command
  - Update User History
  - Update User Profile
- User Hospital Request
- User Info
- User Selected Hospital
- User Place Info
- User Command And Data
4.1.1.4 level-3 Get Patient Place Info

Get User Places Info

Hospitals

Find Close Place

Place List

User Location

Prepare Output

Place Info

Place Info

Get Place Info

Hospitals
4.1.2 Use Cases

System

- Create Account
- Edit Patient Profile
- Edit Hospital Profile
- Get Patient History
- Get Close Hospital
- View Profile
- Mark And Push Hospital
- Rate And Comment Hospitals & Doctors
- Make Appointment

Guest

Patient

Admin
4.1.2.1 Create an account

Description: User must have an account to use the specific features of the application
Assumption: This use case is both for place doctors and hospitals and patientss.
Steps:
1. Patient enters application’s web interface and accesses the registration form.
2. Patient fills the form and registers to the system with a unique user name.
3. User synchronizes their account with other hospital network accounts
4. User sets the privacy configuration as he desires.

4.1.2.2 Edit Patient Profile

Description: Users are able to make changes on their user profiles
Assumption: User has logged in into mos-healt system.
Steps:
1. User accesses his profile by selecting the "Edit my profile" menu.
2. User updates his personal information and settings created in registration.
3. System saves changes when user clicks the "save" button

4.1.2.3 Edit Hospital Profile

Description : Hospital manager are able to make updates and publish information about events in their place
Assumption: Hospital or doctor has registered into system and his account has been approved by the administrator.
Steps:
1. Place owner accesses his place's profile by selecting the "Edit profile" menu.
2. Owner modifies the general information about the place.
3. He enters upcoming events which are going to be held in this place.
4. System publishes the changes after transaction is completed.
4.1.2.4 Get patient's history

Description: Hospital or doctor wants to learn health history of the people
• Assumption: A number of events can be stated in database for one patient.
• Steps:
  1. Place owner opens his own place’s profile.
  2. System gets the list of patients in the place.
  3. System compares and combines the health events of the patient.
  4. System displays the dominant illness of the patient.

4.1.2.5 Get Place Suggestion

Description: Patients are able to discover new health places according to their health problems
• Assumption: System has enough information about user preferences to make appropriate suggestions to the user.
• Steps:
  1. User initializes the suggestion operation by selecting it on the main menu.
  2. User may select other friends who will come with him, to get a common suggestion.
  3. System determines the list of places that are close to user.
  4. System calculates the compatibility ratings of these places according to patients’ previous experiences.
  5. System displays these places ordered by the ratings and marks them on a map widget.

4.1.2.6 View Profiles

• Description: Users are able to view the closest hospital and doctor’s office or the closest settings of pharmacy
  Assumption: User has a mobile phone where location service is available or he/she access internet
• Steps:
1. User log in to Google maps
2. System shows the closest hospital
3. User select one of them
4. System displays the corresponding results related to user’s request.
5. User accesses the related request by selecting the appropriate result.

4.1.2.7 Mark hospital place

- Description: Users are able to mark the place that they are close to the hospital or pharmacy
- Assumption: User has a mobile phone where location service is available or he/she access internet
- Steps:
  1. User selects "mark location" menu in the main screen.
  2. System gets the location of the hospital and displays places near him.
  3. System shows the place where he currently is.
  4. System displays the trace that how he/she can go there
  5. System gives the time (nearly) when he/she can reach there
  6. System gives the list pharmacy closest to him/her

4.1.2.8 Rate and comment hospitals & doctors

- Description: Users are able to add comment, which are not currently in the database.
- Assumptions:
  - User knows the name of the doctor or the location of the hospital.
  - The comment has not been written yet.
- Steps:
  1. User initializes the "add comment" operation by selecting the appropriate item.
2. System displays the city map, the location of hospital and the doctor profile
3. User marks the location of the place on the map.
4. User enters name and description about the place
5. User adds the comment about the doctor.
6. User assigns tags to the place or the hospital.
7. User gives a rating to the place in a particular scale (e.g. over ten)
8. System updates the overall rating of the place and adds new comment to the database.

4.1.2.9 Browse time table of hospital
• Description: Users can rearrange his/her schedule according to the hospital's schedule
• Assumptions:
  • The hospital has already been registered in the database.
  • The hospital has already been entered its schedule to our database.
• Steps:
  1. User log in to Google Maps
  2. User select the hospital which was registered in our database
  3. User click the “show schedule” button
  4. User select the appropriate hospital or doctor
  5. User take a book from selected one

4.1.2.10 Display the appropriate pharmacy
Description: Users can browse the pharmacies which are appropriate to them
Assumptions:
The pharmacies had registered in the database
Steps
  1. Users log in Google Maps
  2. User select hospital
  3. User click the “show pharmacies” button
  4. User selects the appropriate pharmacy
  5. User selects the appropriate drug according to its price
  6. User ordered this drug
  7. User log out
4.2. Non-Functional Requirements

4.2.1 Runtime Requirements

4.2.1.1. **Usability:** The system should have easy user interfaces. It should have minimum number of interface steps for any facility that it supports to user at client side.

4.2.1.2. **Documentation:** The system should include a client side tutorial to ease user’s experience. Also a more detailed online documentation about the system’s usage should be prepared.

4.2.1.3. **Availability/Reliability:** The system should be available to its users almost every time. When it is not possible to reach the system’s server side, the client side application should log events of the user for further push to server.

4.2.1.4. **Scalability:** The system should able to be used when it have 100 – 100,000 clients. So, an automated logic behind the system is mandatory. (i.e. An approach where admins control data legibility will most probably fail.) Also the facilities supported to users should not depend on much user crowd. Almost all facilities should be meaningful when the system has less or more users.

4.2.1.5. **Security:** The system’s client and server side should communicate with each other through a secure way, like https, since client side will push personal info about user.

4.2.1.6. **Quality of Service Requirements:** System should be able to process at least 10000 transactions between server-client per minute.
4.2.2 Development Requirements

4.2.2.1. Localizability: The system doesn’t have to localizable necessarily. But an approach which enables localizability in the future could be beneficial.

4.2.2.2. Modifiability/Extensibility: The system should have enough modularity so that it could be extended by new features/facilities for users in the future.

4.2.2.3. Portability: The system’s server-client communication should be platform-independent.

4.3. System Requirements

4.3.1. Client Side Requirements

When we look at all the possible technologies, libraries, platforms that we will use in our project; we have seen that the most convenient programming language is Java. Because of its platform independent property it is the most convenient programming language. Google Maps API is written in Java and most of the Turkcell project is written in Java. It is another reason why we will use Java in our project. All these reasons led us to use Java language as our applications default developing language.

4.3.2. Server Side Requirements

We will have a simple web service in order to find the place of the. This information will save and update their stock information from web. For this purpose we needed to search for current web developing languages and frameworks. At first glance we have the following options:

- PHP
- Java
5. Project Schedule

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analysis</td>
<td>Mon 06.10.09</td>
<td>Tue 11.10.09</td>
</tr>
<tr>
<td>2</td>
<td>Topic selection</td>
<td>Mon 06.10.09</td>
<td>Mon 12.10.09</td>
</tr>
<tr>
<td>3</td>
<td>Project proposal</td>
<td>Mon 12.10.09</td>
<td>Mon 15.10.09</td>
</tr>
<tr>
<td>4</td>
<td>Field Research</td>
<td>Mon 19.10.09</td>
<td>Mon 25.10.09</td>
</tr>
<tr>
<td>5</td>
<td>Marketing Ideas</td>
<td>Mon 19.10.09</td>
<td>Wed 21.10.09</td>
</tr>
<tr>
<td>6</td>
<td>Technology</td>
<td>Wed 21.10.09</td>
<td>Mon 26.10.09</td>
</tr>
<tr>
<td>7</td>
<td>Requirement Analysis</td>
<td>Mon 26.10.09</td>
<td>Wed 08.11.09</td>
</tr>
<tr>
<td>8</td>
<td>Illustrations</td>
<td>Wed 08.11.09</td>
<td>Tue 17.11.09</td>
</tr>
<tr>
<td>9</td>
<td>Initial Design</td>
<td>Tue 17.11.09</td>
<td>Fri 11.12.09</td>
</tr>
<tr>
<td>10</td>
<td>Components</td>
<td>Tue 17.12.09</td>
<td>Fri 27.12.09</td>
</tr>
<tr>
<td>11</td>
<td>Interfaces</td>
<td>Mon 23.11.09</td>
<td>Mon 30.11.09</td>
</tr>
<tr>
<td>12</td>
<td>Date Specifications</td>
<td>Fri 27.11.09</td>
<td>Fri 04.12.09</td>
</tr>
<tr>
<td>13</td>
<td>Illustrations Initial Design</td>
<td>Fri 04.12.09</td>
<td>Fri 11.12.09</td>
</tr>
<tr>
<td>14</td>
<td>Detailed Design</td>
<td>Fri 11.12.09</td>
<td>Mon 11.01.09</td>
</tr>
<tr>
<td>15</td>
<td>User Interface Drafts</td>
<td>Fri 11.12.09</td>
<td>Mon 14.12.09</td>
</tr>
<tr>
<td>16</td>
<td>Communications and Interface</td>
<td>Mon 14.12.09</td>
<td>Fri 18.12.09</td>
</tr>
<tr>
<td>17</td>
<td>Database Design</td>
<td>Fri 18.12.09</td>
<td>Wed 23.12.09</td>
</tr>
<tr>
<td>18</td>
<td>Class Hierarchy</td>
<td>Mon 21.12.09</td>
<td>Fri 25.12.09</td>
</tr>
<tr>
<td>19</td>
<td>Illustrations Detailed Design</td>
<td>Fri 25.12.09</td>
<td>Mon 01.01.10</td>
</tr>
<tr>
<td>20</td>
<td>Implementation</td>
<td>Fri 01.01.10</td>
<td>Mon 31.01.10</td>
</tr>
<tr>
<td>21</td>
<td>Google Map API</td>
<td>Fri 10.10.10</td>
<td>Fri 22.10.10</td>
</tr>
<tr>
<td>22</td>
<td>Java UI prototype</td>
<td>Fri 22.10.10</td>
<td>Fri 22.10.10</td>
</tr>
<tr>
<td>23</td>
<td>Verse Profiles</td>
<td>Thu 25.10.10</td>
<td>Mon 30.10.10</td>
</tr>
<tr>
<td>24</td>
<td>Illustrations (Prototype)</td>
<td>Fri 25.10.10</td>
<td>Fri 25.10.10</td>
</tr>
<tr>
<td>25</td>
<td>Authentication Auth</td>
<td>Fri 05.10.10</td>
<td>Fri 12.10.10</td>
</tr>
<tr>
<td>26</td>
<td>TurboSort Service</td>
<td>Fri 12.03.10</td>
<td>Fri 19.03.10</td>
</tr>
<tr>
<td>27</td>
<td>Verse Markup / Code</td>
<td>Mon 15.03.10</td>
<td>Sun 21.02.10</td>
</tr>
<tr>
<td>28</td>
<td>Illustrations/Release 1</td>
<td>Sun 21.02.10</td>
<td>Fri 05.03.10</td>
</tr>
<tr>
<td>29</td>
<td>User Profiles</td>
<td>Fri 26.02.10</td>
<td>Fri 26.03.10</td>
</tr>
<tr>
<td>30</td>
<td>Networking and Friends</td>
<td>Mon 06.03.10</td>
<td>Mon 15.03.10</td>
</tr>
<tr>
<td>31</td>
<td>Messaging System</td>
<td>Mon 17.03.10</td>
<td>Mon 22.03.10</td>
</tr>
<tr>
<td>32</td>
<td>Illustrations/Release 2</td>
<td>Mon 22.03.10</td>
<td>Fri 05.04.10</td>
</tr>
<tr>
<td>33</td>
<td>Suggestion System</td>
<td>Tue 15.04.10</td>
<td>Sun 18.04.10</td>
</tr>
<tr>
<td>34</td>
<td>Social Network Info</td>
<td>Mon 19.04.10</td>
<td>Fri 21.04.10</td>
</tr>
<tr>
<td>35</td>
<td>Toast and Debug</td>
<td>Fri 21.04.10</td>
<td>Fri 31.04.10</td>
</tr>
<tr>
<td>36</td>
<td>Developer Documentation</td>
<td>Fri 31.04.10</td>
<td>Mon 01.05.10</td>
</tr>
<tr>
<td>37</td>
<td>End User Workflow</td>
<td>Mon 21.05.10</td>
<td>Mon 31.05.10</td>
</tr>
<tr>
<td>38</td>
<td>Illustrations (Final)</td>
<td>Mon 31.05.10</td>
<td>Mon 31.05.10</td>
</tr>
</tbody>
</table>

6. Conclusion

During the preparation of this requirement analysis report, our team has gained invaluable insights and experiences about the project. Firstly, some points that were unambiguous in our minds at the beginning of the term are now clearer to all team members. Though preparing this requirement report was hard and tiring but
we, BUTA, think that it will be very beneficial to us in the coming months. This report aimed to show our project’s requirement details in terms of several aspects. We give a brief summary of “HEALTH@HAND” in this report. Then, a marketing and technology research is carried out and results are established. And at the main part, the requirement details of the project are described. As a last work, project’s schedule is presented. This report tried to focus on the aspects which thought to be important. So, there is no part that reflects irrelevant or useless information. This report was very useful for clarifying the project’s scope. Also, it’ll be beneficial for further planning.

The project which we work on (“HEALTH@HAND”) is an intersection point of software and hardware. Without making a good research about the subject and requirements, we would surely mess up. We aimed to avoid this with a complete and satisfying requirement analysis report.

Software research has taken too much time, but finally we have found what we need. Also, we have finished our initial design related with the hardware. We have arranged our schedule and risks such a way that nothing unexpected will happen.

There is a considerable need this project in health world and software world. We think that our project will satisfy the need and lead new brilliant ideas in this field.

To sum up, we have chosen this project because we wanted to combine our technical knowledge about hardware, software and other engineering concepts. Our goal is to conclude our opinions and researches with this report.