METU DEPARTMENT OF SOFTWARE ENGINEERING

CENG 491 COMPUTER ENGINEERING DESIGN

SOFTWARE REQUIREMENTS SPECIFICATION

for

CONTEXT AWARE USER INTERFACE PROJECT

MOMO SOFTWARE

Burak Kerim Akkuş - 1559855

Ender Bulut - 1559996

Hüseyin Can Doğan - 1560077

2010

1. INTRODUCTION

1.1 Problem Definition

Mobil devices have become very popular in many areas such as communication in our daily lives, GPS technology in traffic and in many ways in military base recently. The demand of development of mobile device technology increases due to its usability. In many aspects, mobile device technology has progressed for users to be able to use easily. However, users have difficulties during changing environment conditions. For example, you may not use these small devices easily for different contexts, different light conditions, different colored environments or movement. An adaptable graphical user interface which can adapt itself while user is in a continuously moving state will be a very efficient answer for this problem.

1.2 Purpose

This document provides a complete description of all the functions and specifications of Context Aware User Interfaced Project under the joint task with ASELSAN. Moreover, it will serve as a guideline in the development state of the project. As a first release, it is possible that SRS would be shaped accordingly.

The intended audience for the SRS involves all users and possible software developers associated with the project.

1.3 Scope

This document includes overall description, specific requirements, behavioral model and general information of the project. It is very important is order to create our design report and progress our next steps.

The software to be produced is a context aware user interface project. Adaptable Graphical User Interface will be designed to provide users changeable and adaptable interface for various conditions. Our project will be an Android based project. Our project includes a center called “Main Server”, and nodes as a changeable number of mobile devices. When a mobile device is in Receiver mode, it can read data from Main Server. In contrast, when a mobile device is in Sender mode, it can send data to Main Service.

Our project provides two main types of communication models as one to one and one to many. That is, a mobile device in Sender mode can send data to a unique mobile device in Receiver mode or many devices all in Receiver mode.

Our project will support two data communication in terms of data types. One of them will be continuous communication containing map, environment information, feedback and GPS. The other one will be single communication containing messages and tasks.

1.4 User and Literature Survey

Although area of mobile device technology is developing quickly, there is no specific product that is related to adaptable and changeable user interface. Apple is the most powerful company about mobile devices. Especially, iPhone3 and iPhone4 models of Apple are very good at mobile applications. An iPhone has a feature that the window of the device can be rotated with the dimension which user rotates the device. User can change the contrast of the screen of the device when he/she has difficulty to see the context of the screen due to intensity of light. User can change the interface of the device manually by using a specific application. Moreover, most of the mobile devices now have sensors that help adapting the brightness of the screen, accelerometers help rotating user interface etc. However, no company has a product that can be adaptable dynamically for changing environment conditions such as movement of user, light density etc.

The potential users can be soldiers in military service, mobile phone users etc. Adaptable graphical user interface is suitable to be used in many areas.

1.5 Definitions and Abbreviations

SRS: Software Requirements Specification

PDA: Personal Digital Assistant

GUI: Graphical User Interface

3G: 3rd Generation

1.6 References

IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)

Çakıcı, Ruken "Software Requirements Specification Report"
<https://cow.ceng.metu.edu.tr/Courses/?semester=20101&course=ceng491&cedit=0>.

1.7 Overview

This document has six main sections. The first of five additional sections (Overall Description) is designed to describe the general factors that affect the product and its requirements. The third section is specific requirements which explain all the software requirements to a level of detail. This enables designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. The next section describes information domain for the software. The fifth section is behavioral model which is to present a description of the behavior of the software. The last section is about planning the team structure, estimation (basic schedule) and process model.

2. OVERALL DESCRIPTION

2.1 Product Perspective

Adaptable Graphical User Interface system typically will enable the user to provide a certain interface (in different forms) while moving (walking, running etc.) between wireless access technologies, locations. Moreover, our project will involve making the mobile device functionality available for different availability of output devices, input devices and light sensors, acceleration sensors as well as adapting the user interaction operability to the current speed, noise or operator handicaps while keeping in mind the overall applicability depending on the user preferences, his knowledge, current task etc.

Our product is a bit different from those systems in terms of the type of context adaptation. Our software will change its interface dynamically in parallel with environment conditions during in use. The light and the color problems that worsen the effectiveness of the interface will be solved directly. To be more specific, the program will understand whether you are standing, walking or running, the environment is dark or bright and whether the camera is looking at the sky, to a forest or that kind of a specific colored area. Then, it will understand user’s conditions and change its interface which it is most suitable for user. In this way, our mobile devices will have user interface to increase the quality of displayed information and to provide users more effective interaction. We plan to demonstrate our final work on an Android-based mobile device such as a PDA or a tablet.

2.2 Product Functions

When we think about the product functions, we can analyze the system as two separate parts which defines our use-case diagrams.

First part is considered after the user login to the application by writing his/her username and password. Then, the user has a capability to perform some tasks and functions.

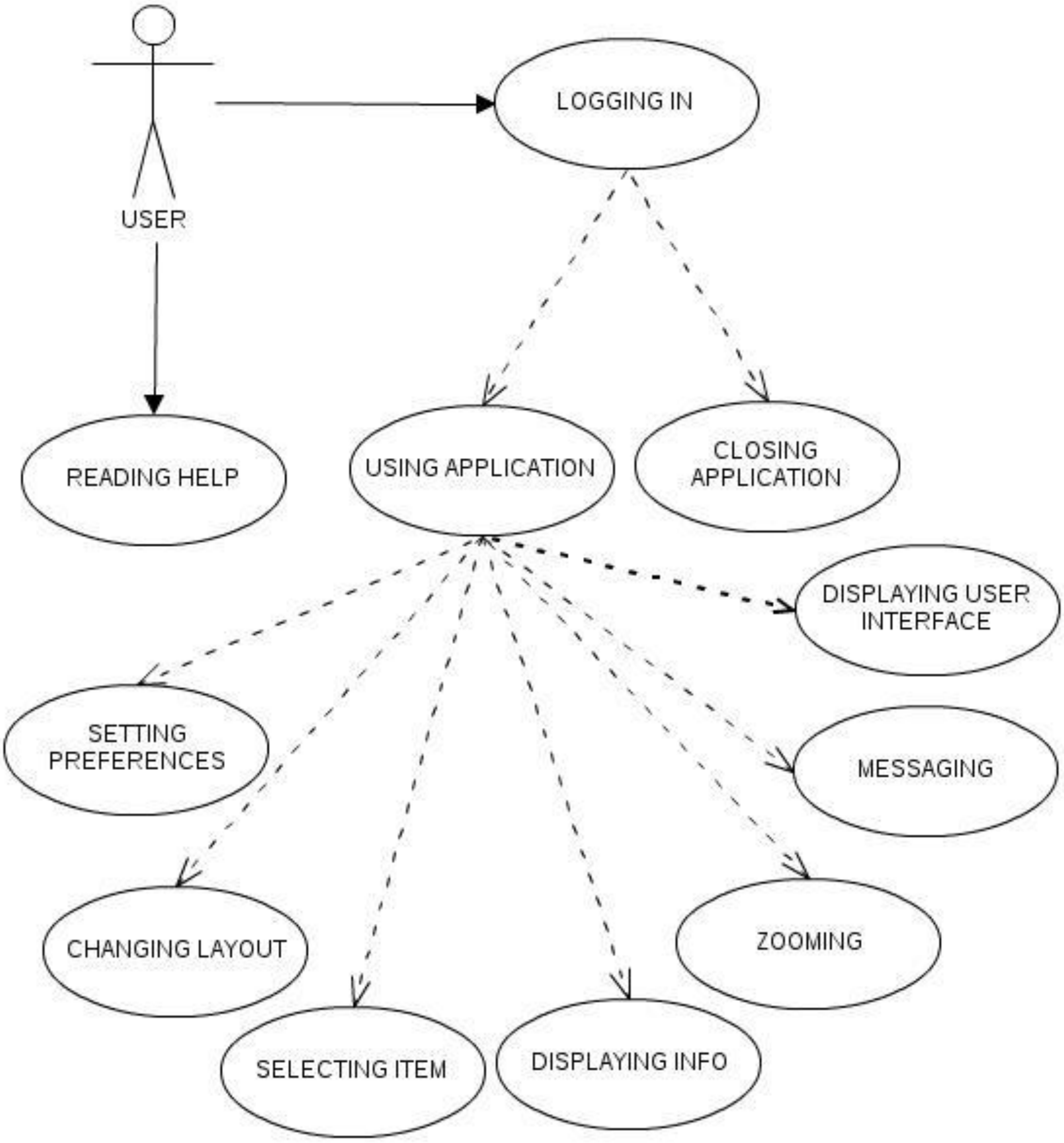


Figure 1 – Use Case Diagram 1

The use-case diagram which is at the upper part shows us our user (actor of use-case) should be able to perform some major functions by using the application.

Second part of the system that defines our use-case diagram is related with GUI controller and its major functions. In this diagram our GUI controller mechanism can be considered as an actor of diagram. Since our system has a capability to perform some major functions which is related to GUI part, all of them are placed on the following use-case diagram.

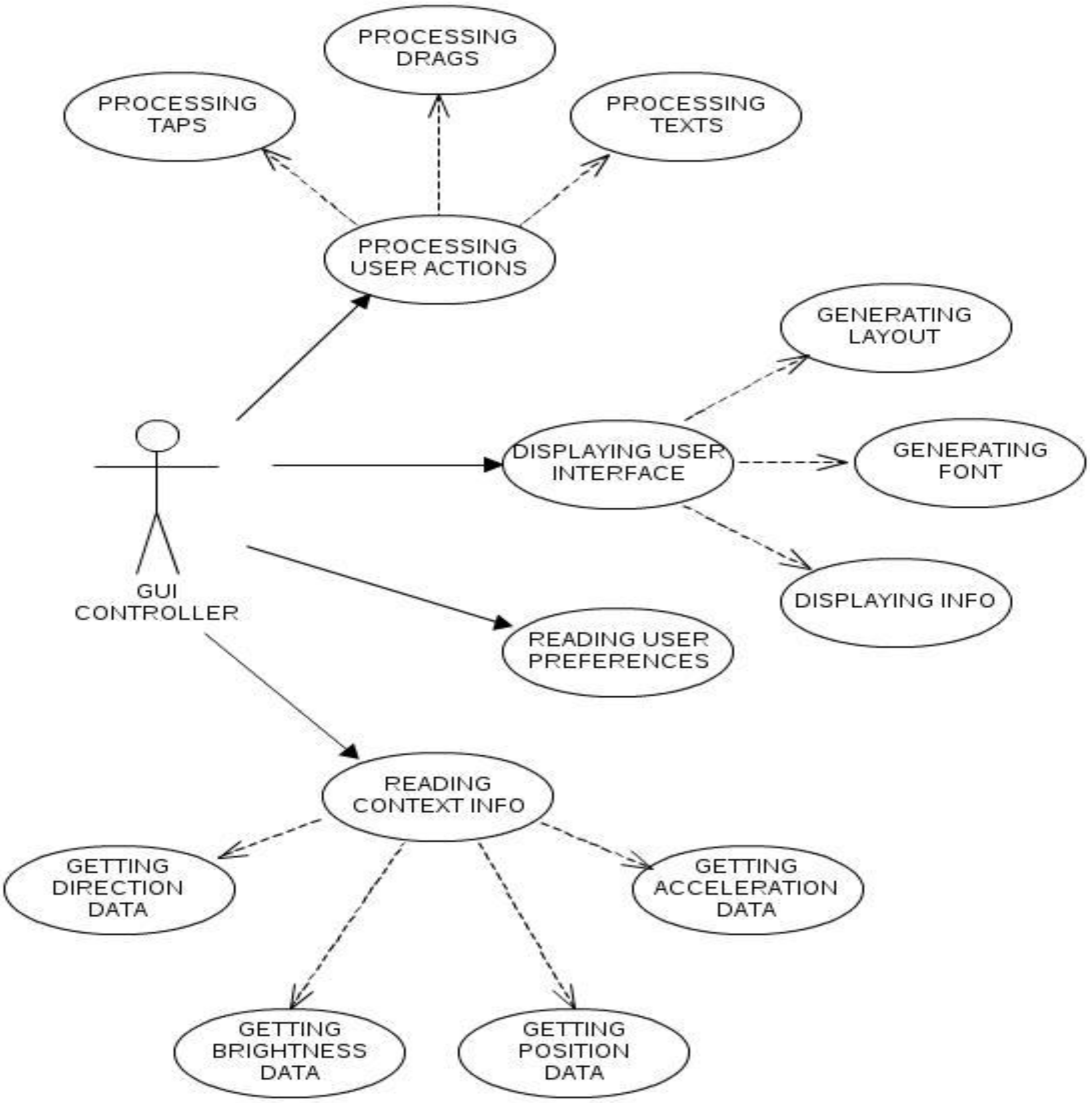


Figure 2 – Use Case Diagram 2

2.3 Constraints, Assumptions and Dependencies

Getting map info is an important part of our project. We assume that maps are stored in a known place which enables us to store and collect map info.

Safety and security are important requirements for our application. Authentication unit and the substructure of our system provide the users to use functions and features of system in a safety and security way.

Finally, we plan to develop the system by using emulator. We will use some sensor features of it and develop our codes. Although the conditions of emulator may be differ to a real device for many dimension, we will start to develop our project so that it will run on an emulator firstly.

3. SPECIFIC REQUIREMENTS

3.1 Interface Requirements

Interface concept can be considered as one of the main and important issues of our system. User is in a strong relationship with user interface, because he/she can operate the tasks and control all info by directly using it. We can analyze the main interface’s input and outputs by associating them with the use cases (especially the one related with GUI controller) that are showed in 2.2.

Processing User Actions

User interface is created for users to use and understand the system easily and cleverly. Therefore it has to understand the actions that were made by users and process them in order to answer the requested info. As explained in 2.2, processing user actions is an important requirement for our user interface. Input can be considered as the actions that were made by the users and output is reasonable result by processing these actions.

Displaying User Interface Components

Our interface contains many different objects, figures, information etc. In order to display all these parts to the user, user interface should have to generate all interface components. When the user enters to the application, map, layout, information about system and other components will be displayed on the screen.

Reading Environment Info

The user interface should be restructured according to the values of environment conditions, acceleration info and some other metrics that are related to map and direction. Therefore our interface should know about these values and changes structure of displaying window according to them. Input of this requirement is info that is related to environment conditions or map and the output is new displayed interface according to input value.

3.2 Functional Requirements

Level 0 Data Flow Diagram

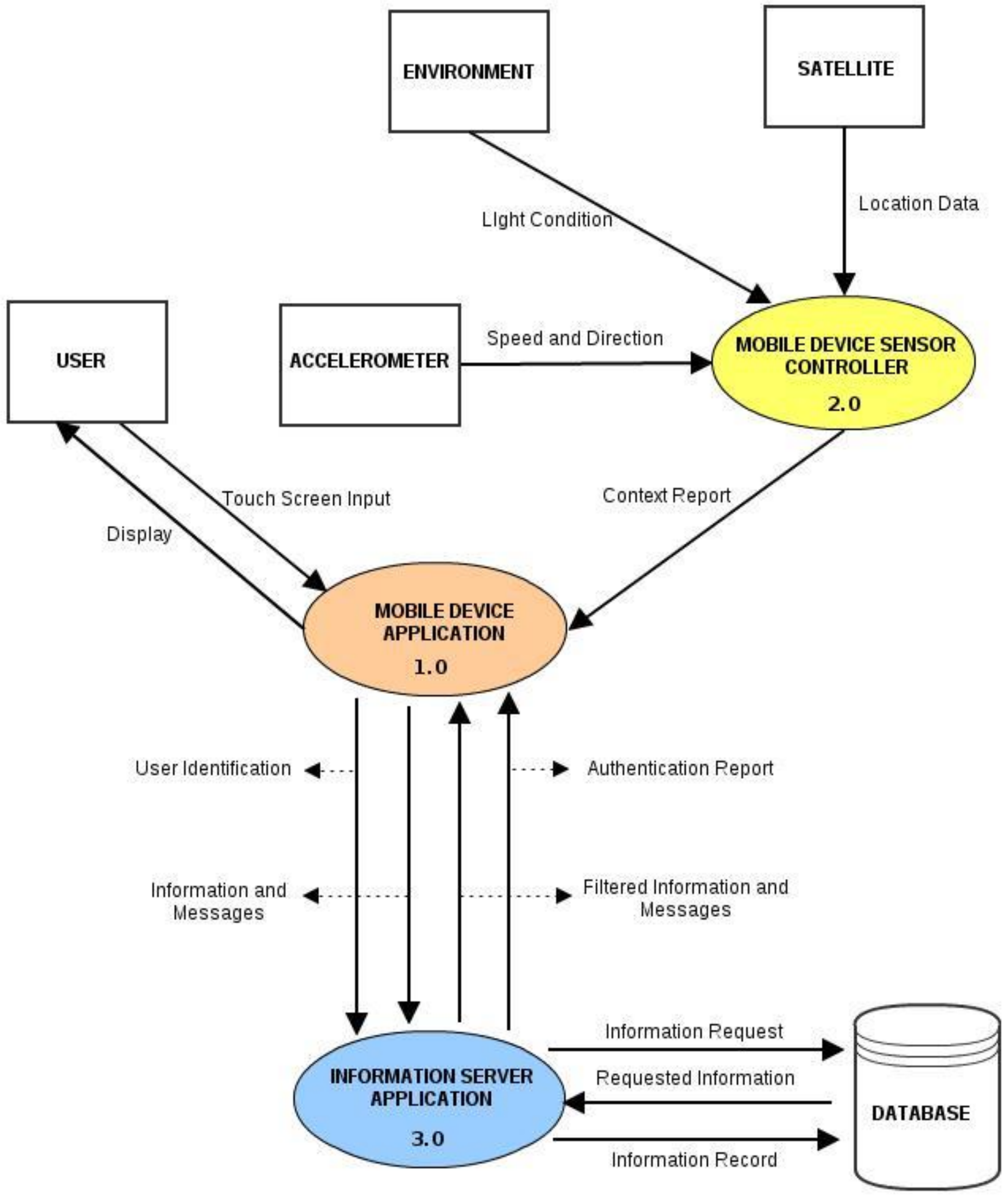


Figure 2 – Level 0 Data Flow Diagram

Level 1 Data Flow Diagram (1/3)

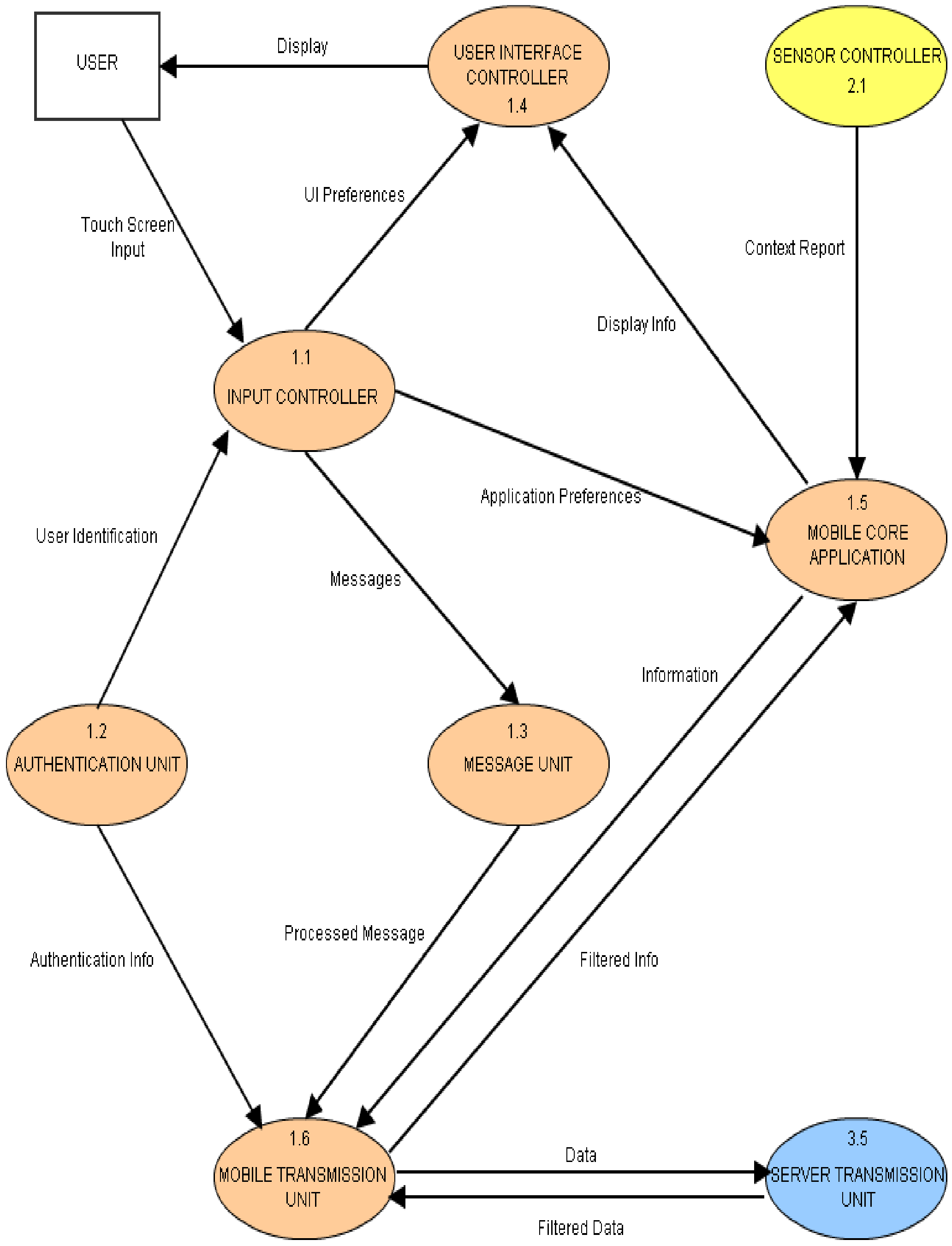


Figure 4 – Level 1 Data Flow Diagram 1/3

Name	**Touch Screen Input**
From	User
To	Input Controller 1.1
Description	All inputs from user

Name	**Display**
From	User Interface Controller 1.4
To	User
Description	Display object that can be shown with each mobile device

Name	**UI Preferences**
From	Input Controller 1.1
To	User Interface Controller 1.4
Description	The information of user interface preferences which represents user choices

Name	**Application Preferences**
From	Input Controller 1.1
To	Mobile Core Application 1.5
Description	The information of current application preferences

Name	**Messages**
From	Input Controller 1.1
To	Message Unit 1.3
Description	The data of text messages which will be processed by message unit

Name	**User Identification**
From	Authentication Unit 1.2
To	Input Controller 1.1
Description	The data containing information about user identification

Name	**Authentication Info**
From	Authentication Unit 1.2
To	Mobile Transmission Unit 1.6
Description	Shows if user is approved or not

Name	**Processed Message**
From	Message Unit 1.3
To	Mobile Transmission Unit 1.6
Description	The data describing the content of processed message

Name	**Display Info**
From	Mobile Core Application 1.5
To	User Interface Controller 1.4
Description	The data that containing all information(UI and application preferences) which is ready for display to user from the screen

Name	**Information**
From	Mobile Core Application 1.5
To	Mobile Transmission Unit 1.6
Description	Data objects about current information of the device

Name	**Filtered Info**
From	Mobile Transmission Unit 1.6
To	Mobile Core Application 1.5
Description	The data contains filtered user info by previous units from server

Level 1 Data Flow Diagram (2/3)

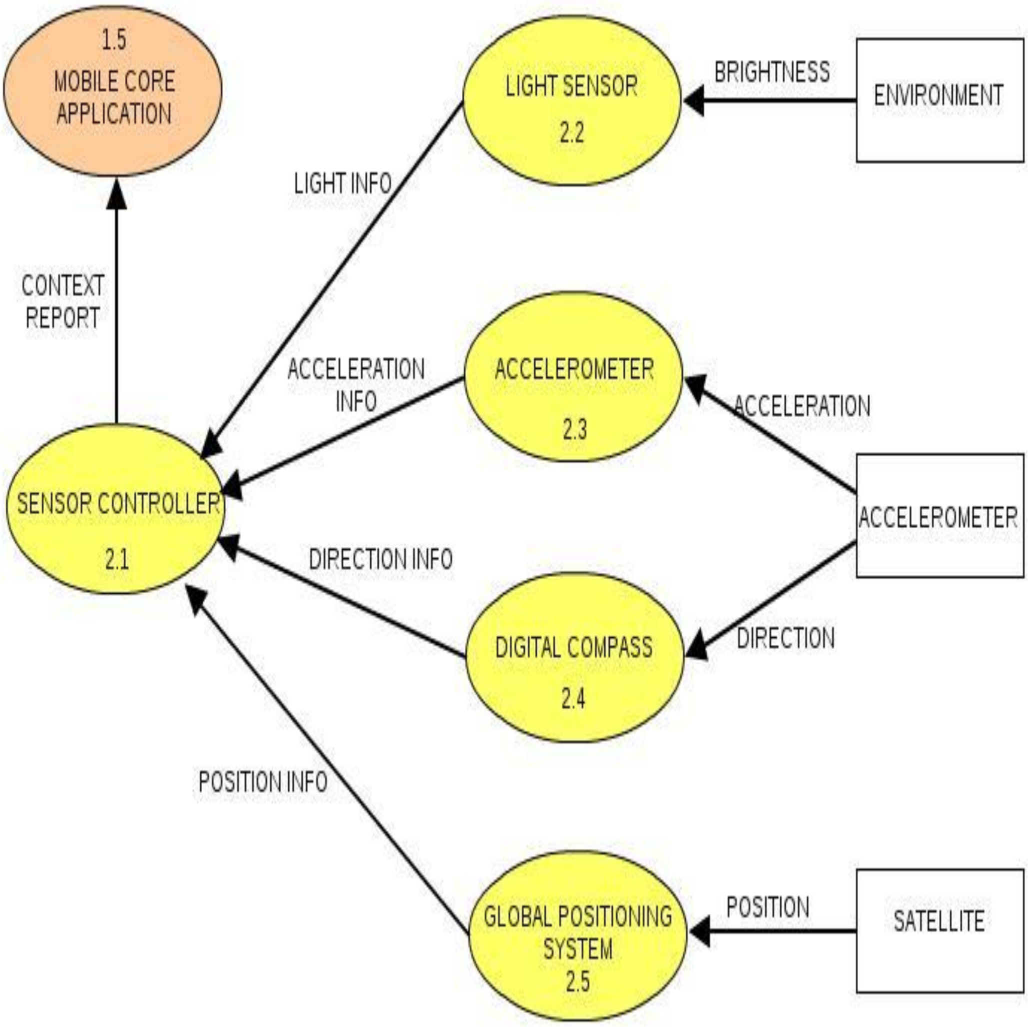


Figure 5 – Level 1 Data Flow Diagram 2/3

Name	**Brightness**
From	Environment
To	Light Sensor 2.2
Description	Describes the value of current light density

Name	**Acceleration**
From	User
To	Accelerometer 2.3
Description	Data describing the value of the current acceleration

Name	**Direction**
From	User
To	Digital Compass 2.4
Description	Data describing which direction the user goes to

Name	**Position**
From	Satellite
To	Global Positioning System 2.5
Description	Data object describing the current coordinates of the user

Name	**Context Report**
From	Sensor Controller 2.1
To	Mobile Core Application 1.5
Description	Data objects which include context information (light info, acceleration info, direction info and position info)

Name	**Light Info**
From	Light Sensor 2.2
To	Sensor Controller 2.1
Description	Includes the current light density information of device

Name	Acceleration Info
From	Accelerometer 2.3
To	Sensor Controller 2.1
Description	Data contains acceleration level information of the device

Name	Direction Info
From	Digital Compass 2.4
To	Sensor Controller 2.1
Description	Data contains the direction information of the user

Name	Position Info
From	Global Positioning System 2.5
To	Sensor Controller 2.1
Description	Data contains position information of the user

Level 1 Data Flow Diagram (3/3)

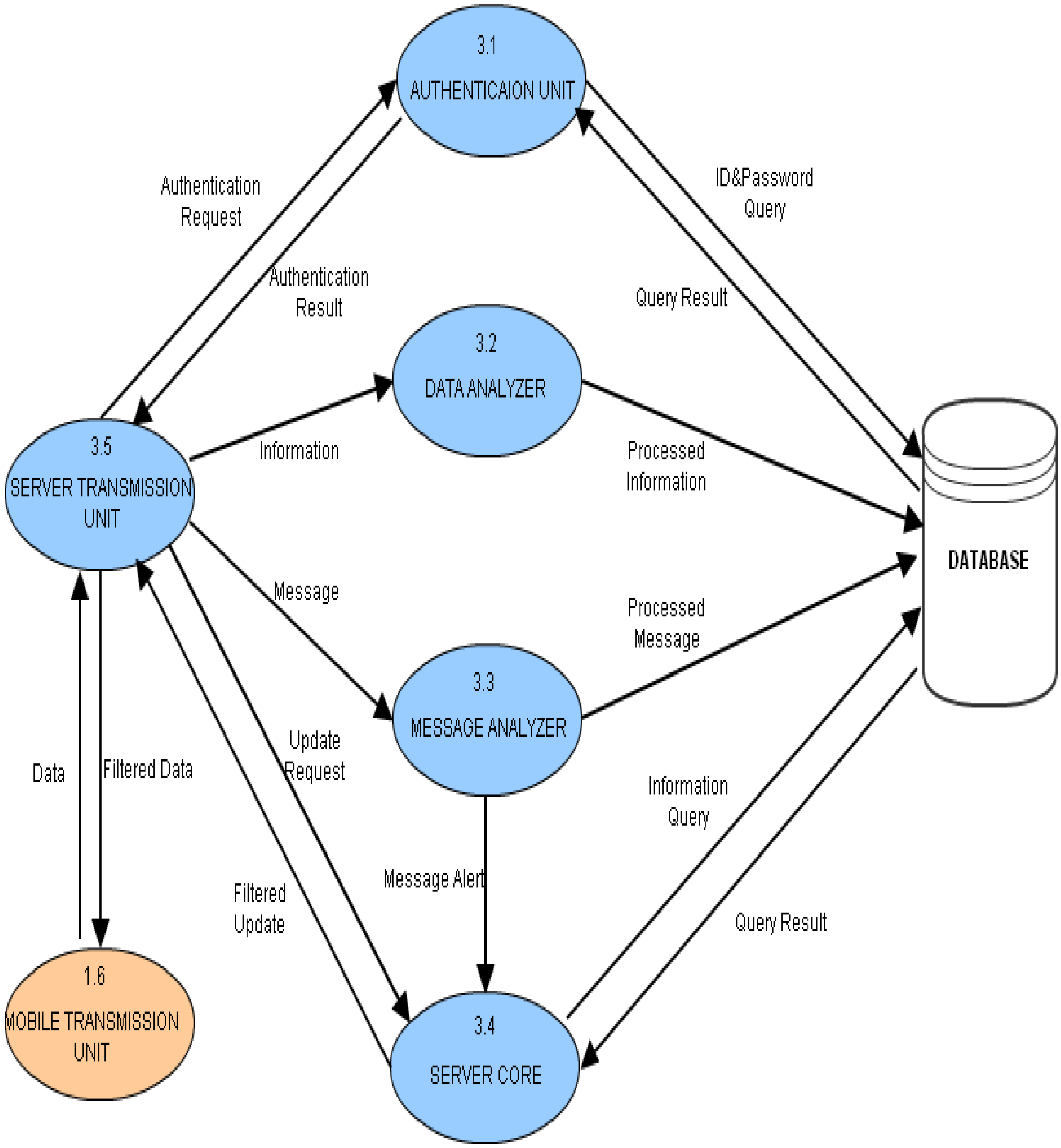


Figure 6 – Level 1 Data Flow Diagram (3/3)

Name	**ID & Password Query**
From	Authentication Unit 3.1
To	Database
Description	The data taken from user to access the database

Name	**Authentication Result**
From	Authentication Unit 3.1
To	Server Transmission Unit 3.5
Description	The result data showing if the it is approved or not

Name	**Processed Information**
From	Data Analyzer 3.2
To	Database
Description	Information which is processed to be sent to the database

Name	**Processed Message**
From	Message Analyzer 3.3
To	Database
Description	The message data which is processed to be sent to the database

Name	**Information Query**
From	Server Core 3.4
To	Database
Description	Data objects including query of information which will be sent to database

Name	**Filtered Update**
From	Server Code 3.4
To	Server Transmission Unit 3.5
Description	New filtered data which is updated with respect to database

Name	**Query Result**
From	Database
To	Authentication Unit 3.1
Description	The result data for correctness of ID & Password Query

Name	**Query Result**
From	Database
To	Server Core 3.4
Description	Data including the result of query information coming from database

Name	**Authentication Request**
From	Server Transmission Unit 3.5
To	Authentication Unit 3.1
Description	Data which contains information for authentication request of user

Name	**Information**
From	Server Transmission Unit 3.5
To	Data Analyzer 3.2
Description	Data objects about current information coming from server

Name	**Message**
From	Server Transmission Unit 3.5
To	Message Analyzer 3.3
Description	The data of text messages coming from server

Name	**Update Request**
From	Server Transmission Unit 3.5
To	Server Code 3.4
Description	New information updated about request

Name	**Filtered Data**
From	Server Transmission Unit 3.5
To	Mobile Transmission Unit 1.6
Description	Data is filtered by server

Name	**Data**
From	Mobile Transmission Unit 1.6
To	Server Transmission Unit 3.5
Description	Objects sent to the server

Name	**Message Alert**
From	Message Analyzer 3.3
To	Server Core 3.4
Description	The data containing if message come to server core or not

3.2.1 Mobile Device Requirements

Mobile Device part is one of the central part of our system. It is in a strong relationship with Sensor Controller and Main Server parts. The requirements of Mobile Device part are explained as the following one by one.

3.2.1.1 Receiving Sensor Outputs from Sensor Controller

Our Sensor Controller Part has information about external environment. In order to use this info and benefit from it, we should be able to manipulate all necessary data. Therefore our Mobile Device receives sensor outputs from Sensor Controller and after that it manipulates and uses this data for changing User Interface and making it better.

3.2.1.2 Getting User Inputs and Actions

Our Mobile Device will be used by users, so it should be able to get and process user inputs and actions. Users may want to see specific information or get some attributes and data of the system and etc. In order to answer all of these requests, the Mobile device should be able to get user inputs and act according to them.

3.2.1.3 Sending Information to Transmission System

Communication between Mobile Device and Main Server is one of the most critical parts of our application. We need a system that can forward the data from Mobile Device to Main Server so that Main Server can answer the needs and requests of the Mobile Device. Transmission System is a kind of bridge that connects these two main parts. Data connection between these two parts is handled by using wireless network connection.

3.2.1.4 Receiving Information from Transmission System

As explained before, communication between Mobile Device and Main Server is very important for our application. We need a system that can forward the data from Main Server to Mobile Device so that Mobile Device can obtain the required information from Main Server. Transmission System should be able to connect these important main parts. Data connection between these two parts is handled by using wireless network connection.

3.2.1.5 Displaying User Interface

The mobile device is also responsible for displaying the map and the related information on it. The mobile device handles the user actions via graphical user interface therefore it is a necessity for device to handle display requirements.

3.2.2 Main Server Requirements

3.2.2.1 Delivering Data to Database

Our application has database that store the relevant data of our system. It stores user information, sent messages, map information and etc. Sometimes we may want to compare a data with the one that is stored in database or we may want to store some information to it. For all of these operations, main server should have a capability to deliver data to the Database.

3.2.2.2 Reading Data from Database

As explained before, our Database system contains data that is required or processed by our application. It stores user information, sent messages, map information and etc. These data may be necessary for the application components. Therefore, Main Server should be able to read data from Database in order to manipulate or send them to the Mobile Device.

3.2.2.3 Sending Information to Transmission System

Sometimes, Mobile Device may depend on data from Main Server part, which means these two parts should be able communicate with each other. When Mobile Device wants to get this information, Main Server sends the required data to Transmission System.

3.2.2.4 Receiving Information from Transmission System

In some cases, Mobile Device may want to send data to Main Server or Database part in order to continue some tasks. When Mobile Device wants to send this information, Main Server should be able to get it from Transmission System that connects these two parts.

3.2.3 Transmission System Requirements

3.2.3.1 Forwarding Mobile Device Info to Main Server

Mobile Device and Main Server parts are in a strong relationship and their data transfers between each other are very high rates. Transmission System should forward Mobile Device info to Main Server part when some information and messages are used by Database or Main Server. Data connection between these two parts is handled by using wireless network connection.

3.2.3.2 Forwarding Main Server Info to Mobile Device

Our application data is not only sent from Mobile Device to Main Server. Sometimes the direction of the dataflow may be opposite. When Mobile Device needs to use and obtain some info that can only be accesses with Main Server. In these cases Transmission System forwards data from Main Server to Mobile Device. Data connection between these two parts is handled by using wireless network connection.

3.2.4 Sensor Controller Requirements

3.2.4.1 Processing Light Information

We have a light sensor that can receive info about brightness of the environment and give it to our Sensor Controller. It is responsible for gathering light info from light sensor and giving it to Mobile Device. Processing light information by Sensor Controller is very important in order to make user interface more clear and understandable.

3.2.4.2 Manipulating Acceleration Information

Acceleration info depends on the user rather than the environment. It changes according to motion and condition of the user. Sensor controller basically gives the acceleration info by using accelerometer and sends this info to Mobile Device part. Sending this info is very important, because our user interface should be able adapt to changing of the acceleration.

3.2.4.3 Receiving Direction Data

Receiving Direction Data is required in order to find and manipulate the direction of the user. This info is important, because the direction of the user is represented and used for user interface part.

3.2.4.4 Gathering Global Positioning Data

Gathering Global Positioning System Data is important in order to identify and show the user’s location. Global Positioning System Data is used for user interface in order to able to define the location of the user on map.

3.2.4.5 Sending Information to Mobile Device

Our Service Controller has information about light of the environment, acceleration, direction and location of the user. However, in order to use and manipulate this required info, Sensor Controller should communicate to Mobile Device. When it requests a special data from controller, Sensor Controller should be able to send it correctly.

3.2.5 Database Requirements

3.2.5.1 Getting Data from Main Server

Main Server needs to communicate the Database when it needs to authenticate the user, analyze the data and get the written messages that exist in Database. However sending data from it to our Database System is necessary to complete these operations successfully. According to this info our database starts its operation and gives the necessary results to it.

3.2.5.2 Giving Database Info to Main Server

Mobile Device and Main Server may want to access the data of database any time, since it contains important info for our application. For each of these cases, Database should give its info to Main Server by executing the queries.

3.3 Non-functional Requirements

3.3.1 Performance Requirements

Lowering User Interface Responding Time

Context aware user interface part is one of the most critical parts of our application. When a user starts to run or the brightness of the environment is changed, user interface should have to react it and change the structure of interface in a short time. Minimizing this time is very important and critical. For our application the updated user interface should have to be displayed to user at most 2-3 seconds. Because of the project's data intensity is very high; we set the limit value as 2-3 seconds rather than milliseconds degree.

Minimizing Map Refreshing Time

In our application, the map and its information are displayed to users. Map may consist of many units in it and the system should reflect the new positions of these units by refreshing the map info. By this way, user can analyze the location of himself/herself and the last position of the units. In normal cases, the program's map refreshing time should be lower than 2 seconds in order to have a dynamic map that reflects the location changes. However this value may increase if the data intensity of map is above the normal values.

Setting the Size of Database

Database part of application is responsible to store info of the system. From the users' info and locations to messages that was sent, all data is kept on this special area. Therefore, our database should be able to store all of project's information in an organized way. In other words the size of it should be set according to the size of the system data and it should work correctly.

3.3.2 Design Constraints

Usability

Usability is an important constraint of our application so that the users can use our application in an effective and correct way. Our user interface should be user-friendly. In other words when a user wants to see some specific data or send message to another user, he/she should able to handle these operations easily and effectively by using the user interface. Moreover he/she should be able to analyze the signs and units of the map easily. Shortly, our application needs to be usable and understandable for users so that they can reach our application's power and capabilities.

Reliability

Reliability is another constraint of our application. The user of the system should be able to get the results of their requests correctly in a specific time. For example when the user wants to see his/her missions or the info of their own unit, the results should be displayed on the screen in a reasonable time. Moreover the flow of information of the system should not corrupt and negatively affect the application. Data or messages are sent / received by the users without affecting the application's performance and correctness.

Security

Our application includes map info, users' and units' info, messages that was created by users, some confidential missions and etc. It is easy to see that security is a critical constraint in order to hide and protect all of this important data. For this reason, we will apply authentication mechanism. In other words, users have to enter their username and password before using our application. If these two variables are not consistent and true, the system does not allow the user for using application.

Performance

As we explained at the upper part, performance is critical concept. For this reason, database features, refreshing rate of the map and the reaction time of the context aware user-interface should be adjusted reasonable values.

4. DATA MODEL AND DESCRIPTION

4.1 Data Description

4.1.1 Data Objects

User

This table represents all necessary information about the users that will use our application. Since we have to define each user separately, firstly we add "user id" field to this table. "user_id" is a primary key of our table. Each user should have unique id number. Second field is user name that holds user names as its name implies. The important point is a user may be part of more than one unit or he/she may be given zero or more than one mission. "user_rank" field shows the rank of the user. The other field, "password", is required for authentication. The last field, "coordinates", shows the coordinate values of the user.

- user_id
- user_name
- user_rank
- password
- coordinate

Unit

A Unit table includes all information about a unit. There may be a lot of units for our application, so “unit_id” represents unique id number of a unit. The primary key of Unit table is “unit_id”. The other field “unit_name” is hold to store the name of unit. We add “unit_size” field to this table for keeping the size of the unit. Finally we have “unit_rank” field which shows the rank of the specific unit with respect to another units.

- unit_id

- unit_name

- unit_size

- unit_rank

Context

This table is considered to keep environment conditions, acceleration, direction info and position info in order to inform Main Server about which conditions user is in. The field “acceleration_val” is used to store acceleration value that is calculated by accelerometer. The third field is “direction_info”. This info is used for our user interface and calculated by digital compass and shows the direction of the device. The last field is “location_info” that keeps the location coordinates of the user.

- acceleration_val

- direction_info

- location_info

Mission

Mission is an important issue for our application and this table stores values about mission info. First of all every mission has unique id, namely “mission_id” that is a primary key. The other field that will be stored is “mission_name”, every mission should have a name. The other two concepts that we want to add are deadline and rank of the mission. Every mission should have deadline information to determine the mission must be ended. Moreover, each mission should have a rank which determines the priority of missions. For this reason, “deadline” and “mission_rank” fields are added to this table.

- mission_id

- mission_name

- deadline

- mission_rank

Map

First of all, Map has some coordinates which defines it on the user interface. Secondly, this table includes field, namely, “unit_list” which is a list of units that have to be displayed on map. After we get the initial version of the map, we add some signs to map which will be represent for each unit.

- coordinates
- unit_list

Message

Actually messages are text messages which users can send or receive and communicate with each other easily. However, the record of each message is kept in the Message table. ”message_id”, is the id number of each message and it is primary key of Message table. “writer_id”/”reader_id” is the user id of user who sends/receives the message. “sent_date”/”receive_date” is the date when the message is sent/received. Keeping message text field is another issue and it is stored in “message_cont” field. The final field is “mes_rank” that shows the importance of the message.

- message_id
- writer_id
- reader_id
- sent_date
- receive_date
- message_cont
- mes_rank

4.1.2 Relationships

UserAndUnits

We explained the fields of User and Units objects. These two tables have fields, namely user_id and unit_id, respectively. A user may be part of more than one unit or a unit may consist of many users. UserAndUnits relationship shows us that the relationship between these two tables. The ERD for this relationship is:

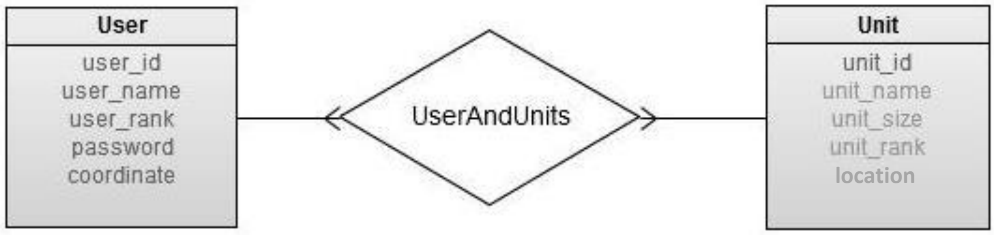


Figure 7 – UserAndUnits Relationship

The attributes of UserAndUnits table are briefly:

-user_id

-unit_id

UserAndMissions

User and Mission data objects are explained in 4.1.1 part. These two tables has many fields, however the important ones that define our relationship are user_id and mission_id. By using these two attributes, we can get info about the users that are given some mission or the missions that are given a user. The ERD of this relationship is:

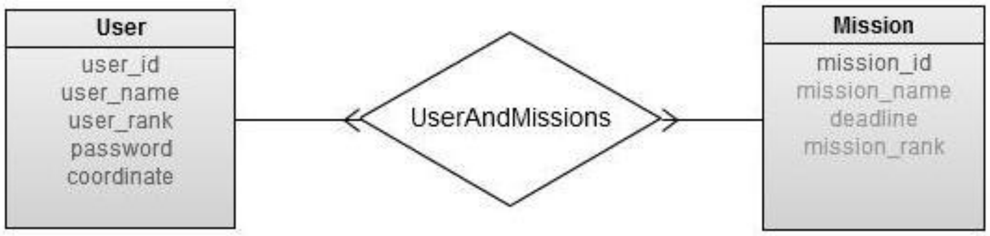


Figure 8 – UserAndMissionsRelationship

The attributes of UserAndMissions table are briefly:

-user_id

-mission_id

4.1.3 Complete Data Model

We have defined some data objects and some relationships. In order to connect them and make them meaningful we create complete data model that includes both data objects and relationship .This version of data model is created for considering the basic and necessary parts. According to the progress of our project, we may make some changes to this model. The ERD of our complete data model is displayed at the below.

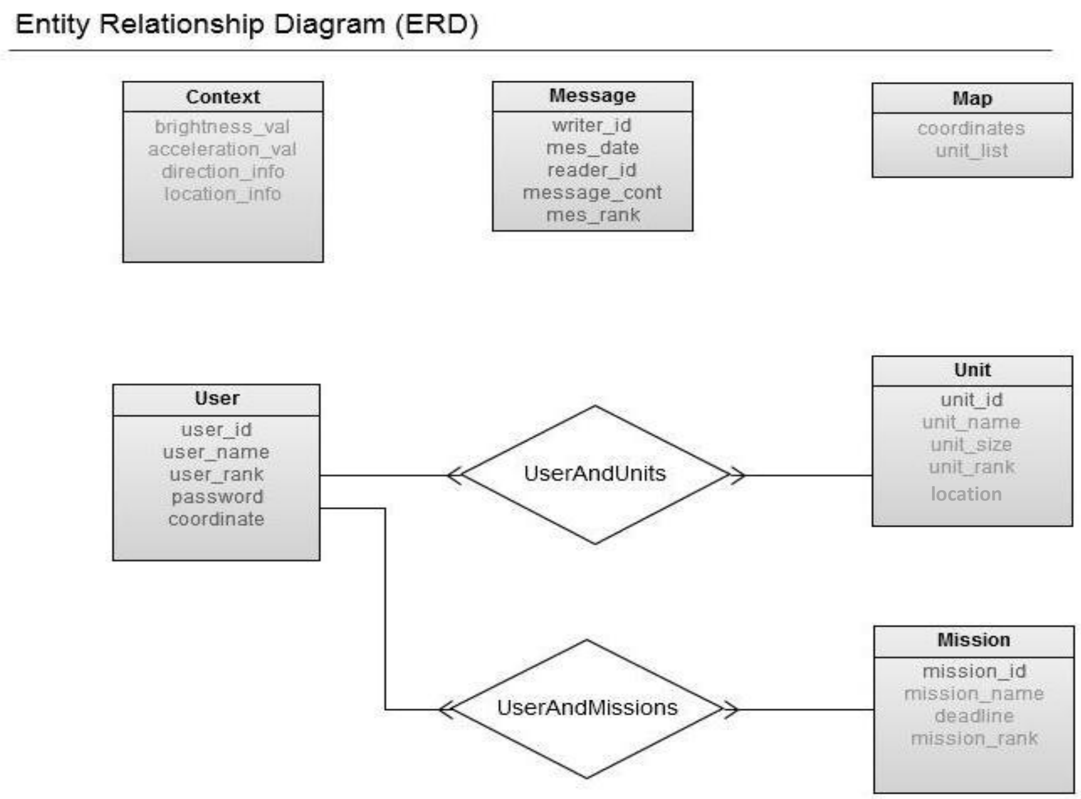


Figure 9 – Entity Relationship Diagram

4.1.4 Data Dictionary

User = user_id

+ user_name

+ user_rank

+ password

+ coordinate

user_id = *id number of user*

user_name = *name of the user*

user_rank = *rank of the user*

password = *password of the user for authentication* coordinate = *coordinate values of the user*

Unit = unit_id

+ unit_name

+ unit_size

+ unit_rank

+ location

unit_id = *id number of unit*

unit_name = *name of the unit*

unit_size = *size of the unit*

unit_rank = *rank of the unit*

location = *location of the unit*

Context = brightness_val

+ acceleration_val

+ direction_info

+ location_info

acceleration_val = *acceleration value that is calculated by accelerometer*

direction_info = *info about direction of the device*

location_info = *info about location of the device*

Mission = mission_id

+ mission_name

+ deadline

+ mission_rank

mission_id = *id number of specific mission*

mission_name = *name of the mission*

deadline = *deadline of the mission*

mission_rank = *rank of the mission*

Map = coordinates

+ unit_list

coordinates = *coordinates of the map*

unit_list = *list of all units that have to be displayed on map*

Message = writer_id

+ mes_date

+ reader_id

+ message_cont

+ mes_rank

writer_id = *user id of user who sends to the message*

mes_date = *date when the message is sent*

reader_id = * user id of user who reads to the message *

message_cont = *stored message text*

mes_rank = * importance of the message *

5. BEHAVIORAL MODEL AND DESCRIPTION

5.1 Description of Software Behavior

First of all, we explain major states of the software. Major states of software can be listed as: Login Screen, Main Window, Message Screen, Detailed Info Screen, Settings Screen and Alert Screen. Everything starts with the Login Screen. After user authentication, we can pass the Main Window state. In Main Window state, we have our main user interface application on the screen. We can pass from this state to Message Screen state for messages, Detailed Info Screen state for context awareness, Alert Screen state for emergency conditions or Settings Screen state for settings.

5.2 State Transition Diagrams

Since the application consists of two distinct parts, which are the context awareness engine and the scenario for which we will develop the user interface, we have two distinct set of states. Therefore, we have two transition diagrams that visualize the relations and changes between these states.

Firstly, there are many variables such as light, motion or position that context awareness application uses to perceive the context or circumstances. Therefore too many states arise to be shown as discrete steps. Moreover, our aim is to build a user interface engine that dynamically changes the interface at each minor change in environmental variables and to be free of just a couple predefined layouts and fonts. In other words, the states of the context awareness application are the combinations of other states that multiply the number with each new variable. Thus, it is better to use a three dimensional graph to represent the states which are approximately infinite in this case.

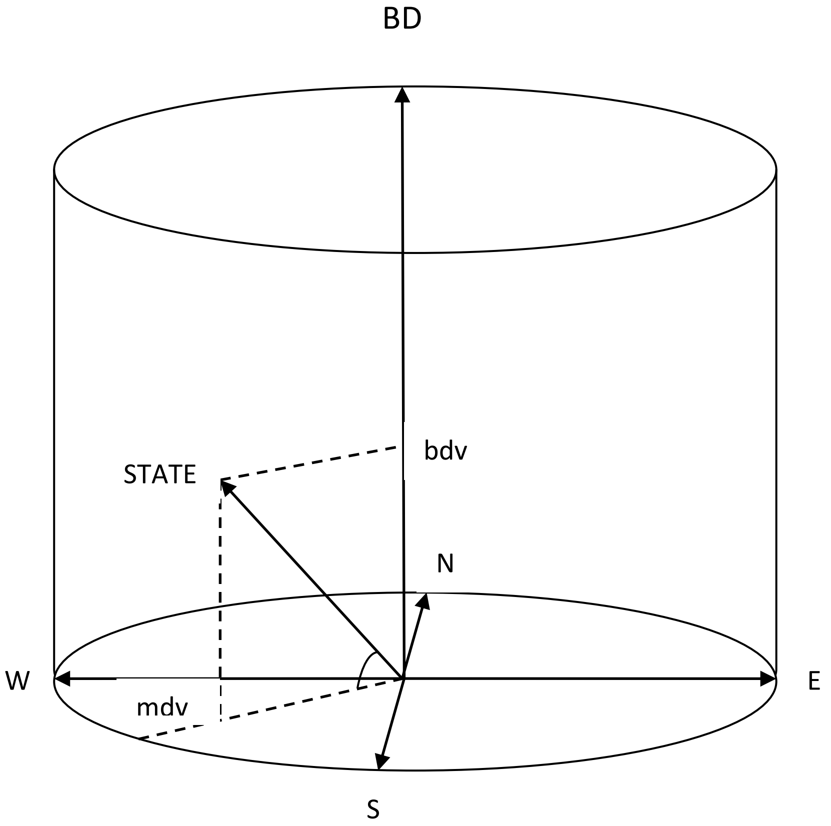


Figure 10 – State Transition Diagram 1

Here, our coordinate axis for three dimensional graph are:

BD represents brightness density and W, E, S and N represent our compass for four directions.

mdv: Movement Density and Direction Value

bdv: Brightness Density Value

In order to understand the meaning of this three dimensional graph, some examples are required. For example; the point on this graph represents the state of a context awareness engine. After some time is passed, the point position is changed. In other words, our brightness, direction and movement density values are changed; shortly we pass to the next state.

Secondly, we will build an application based on a military action scenario to represent the capabilities of the context aware user interface application. This part of the project will consist of states that are separated by specific and certain actions. Therefore, a classical state transition diagram will be the best choice for this.

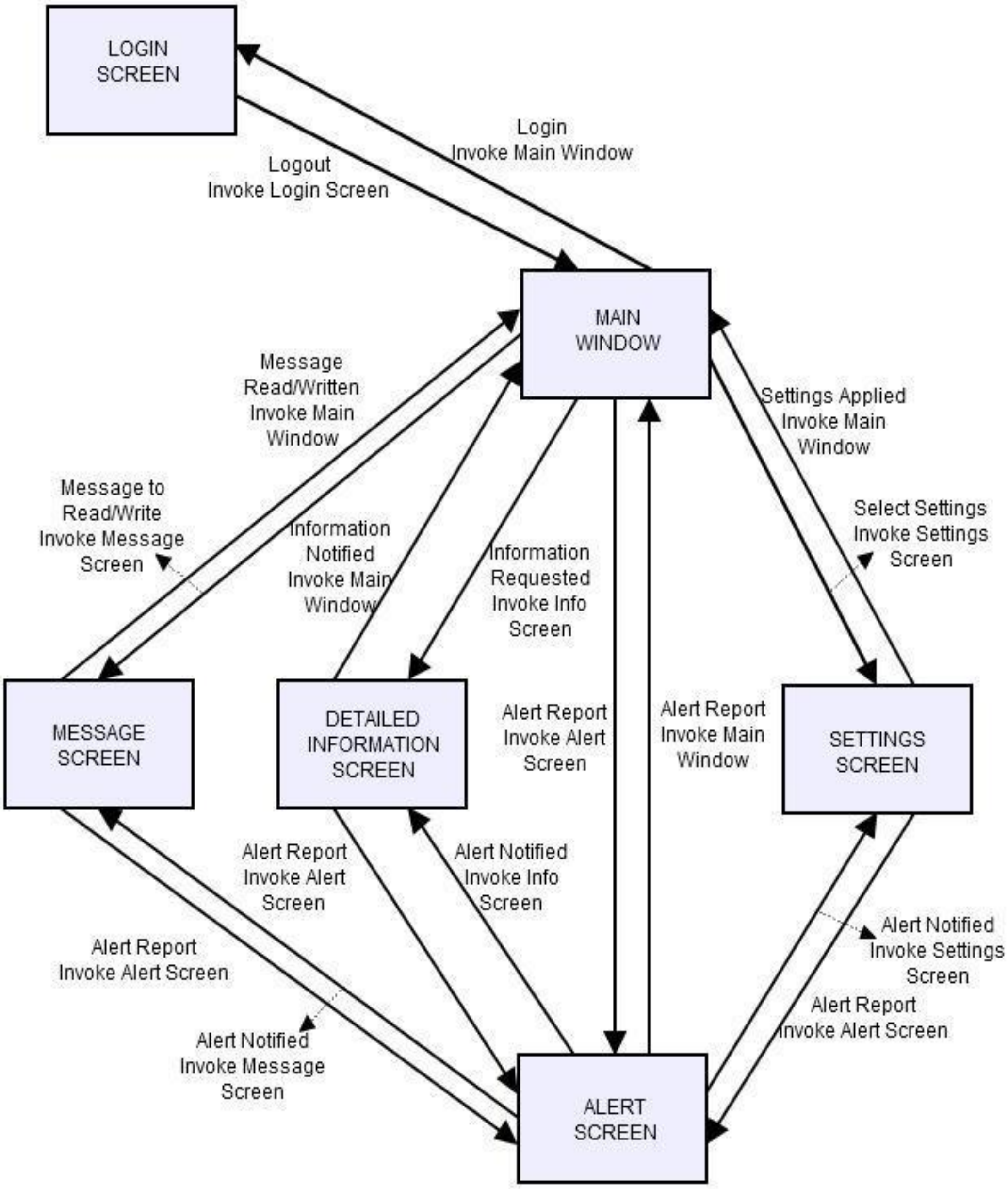

LOGIN SCREEN
Login Invoke Main Window
Logout Invoke Login Screen
MAIN WINDOW
Message Read/Written Invoke Main Window
Message to Read/Write Invoke Message Screen
Information Notified Invoke Main Window
Information Requested Invoke Info Screen
Settings Applied Invoke Main Window
Select Settings Invoke Settings Screen
MESSAGE SCREEN
DETAILED INFORMATION SCREEN
Alert Report Invoke Alert Screen
Alert Report Invoke Main Window
SETTINGS SCREEN
Alert Report Invoke Alert Screen
Alert Notified Invoke Info Screen
Alert Report Invoke Alert Screen
Alert Notified Invoke Message Screen
Alert Notified Invoke Settings Screen
Alert Report Invoke Alert Screen
ALERT SCREEN

Figure 11 – State Transition Diagram 2

6. PLANNING

6.1 Team Structure

The team consists of three members: Burak Kerim Akkuş, Ender Bulut and Hüseyin Can Doğan. Everyone has equal rights and we have a collaborative, mutual decision mechanism. Since we are always in touch and discuss the project related topics together we do not need a leader. Overall design of the project is the result of our joint decisions.

Although we have no strict divisions and everyone contributes to the whole project, each of us coordinates one of the three parts of the project, responsible for details of that part and its interaction with other two. Hüseyin is responsible for sensor coordination and user interface design, Ender is responsible for mobile application and Burak manages server and database related jobs.

6.2 Process Model

After starting to work on our application, we will do analysis and design of the project. Then we will build the software using different approaches at different stages.

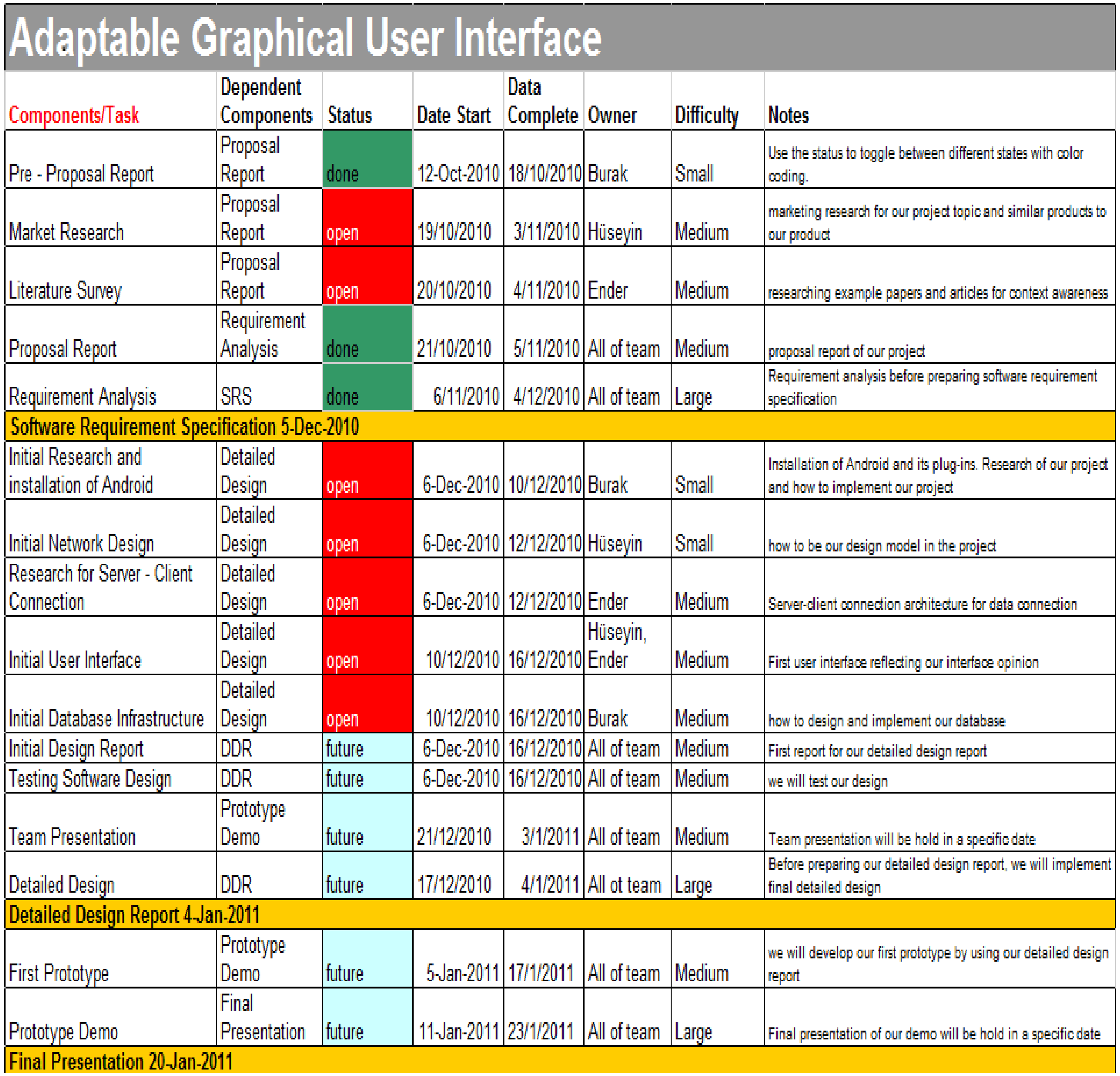
Firstly, we assume that we already build the sensor control, mobile application and the server part (i.e. jump to the rear end of the project). We will use component base approach to specify how to connect them and design the interface between them. We need to know how and in which format the data is transferred between them.

After that we will return the front end of the project and develop each part using iterative approach until we reach a quality that fulfills the product requirements. We will build a prototype with most required attributes and basic features. Then we will add new ones through the development process. At the end we estimate that we will have a product best fits requirements specified in this document.

Using this process model, we will guarantee that parts of the project are well defined, satisfy the requirements and have no trouble communicating with each other.

6.3 Estimation

We define our basic schedule by considering our conditions.



Adaptable Graphical User Interface

Components/Task	Dependent Components	Status	Date Start	Data Complete	Owner	Difficulty	Notes
Pre - Proposal Report	Proposal Report	done	12-Oct-2010	18/10/2010	Burak	Small	Use the status to toggle between different states with color coding.
Market Research	Proposal Report	open	19/10/2010	3/11/2010	Hüseyin	Medium	marketing research for our project topic and similar products to our product
Literature Survey	Proposal Report	open	20/10/2010	4/11/2010	Ender	Medium	researching example papers and articles for context awareness
Proposal Report	Requirement Analysis	done	21/10/2010	5/11/2010	All of team	Medium	proposal report of our project
Requirement Analysis	SRS	done	6/11/2010	4/12/2010	All of team	Large	Requirement analysis before preparing software requirement specification
Software Requirement Specification 5-Dec-2010							
Initial Research and installation of Android	Detailed Design	open	6-Dec-2010	10/12/2010	Burak	Small	Installation of Android and its plug-ins. Research of our project and how to implement our project
Initial Network Design	Detailed Design	open	6-Dec-2010	12/12/2010	Hüseyin	Small	how to be our design model in the project
Research for Server - Client Connection	Detailed Design	open	6-Dec-2010	12/12/2010	Ender	Medium	Server-client connection architecture for data connection
Initial User Interface	Detailed Design	open	10/12/2010	16/12/2010	Hüseyin, Ender	Medium	First user interface reflecting our interface opinion
Initial Database Infrastructure	Detailed Design	open	10/12/2010	16/12/2010	Burak	Medium	how to design and implement our database
Initial Design Report	DDR	future	6-Dec-2010	16/12/2010	All of team	Medium	First report for our detailed design report
Testing Software Design	DDR	future	6-Dec-2010	16/12/2010	All of team	Medium	we will test our design
Team Presentation	Prototype Demo	future	21/12/2010	3/1/2011	All of team	Medium	Team presentation will be hold in a specific date
Detailed Design	DDR	future	17/12/2010	4/1/2011	All ot team	Large	Before preparing our detailed design report, we will implement final detailed design
Detailed Design Report 4-Jan-2011							
First Prototype	Prototype Demo	future	5-Jan-2011	17/1/2011	All of team	Medium	we will develop our first prototype by using our detailed design report
Prototype Demo	Final Presentation	future	11-Jan-2011	23/1/2011	All of team	Large	Final presentation of our demo will be hold in a specific date
Final Presentation 20-Jan-2011							

Figure 12 – Schedule

7. CONCLUSION

In conclusion, we have analyzed overall description of the system, specific requirements, data model, behavioral model and planning about future terms of our project in this report. When we think about these titles, all of them are important firstly to create our detailed design report and secondly to continue the project with well-defined system and structure. All of the information in this report will help us while we will develop our project. Therefore, we try to create this report carefully and correctly.

Finally, we believe that this report will form the basis of our project through this term._