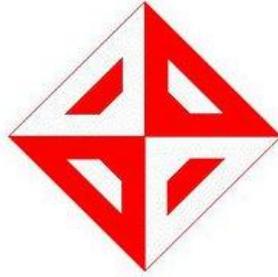




MIDDLE EAST
TECHNICAL UNIVERSITY



COMPUTER ENGINEERING
DEPARTMENT

CENG 491
SOFTWARE REQUIREMENTS SPECIFICATION
REPORT



Korsan Yazılım

Contents	Page
1. Introduction	5
1.1 Problem Definition	5
1.2 Purpose	5
1.3 Scope	5
1.4 User and Literature Survey	6
1.5 Definitions and Abbreviations	6
1.6 References	6
1.7 Overview	7
2. Overall Description	7
2.1 Product Perspective	7
2.2 Product Functions	9
2.2.1 Toolbar Operations	9
2.2.2 Create Filter Operations	9
2.2.3 Assignment Operations	9
2.2.4 Message Operations	10
2.3 Constraints, Assumptions and Dependencies	10
3. Specific Requirements	10
3.1 Interface Requirements	10
3.1.1 User Interfaces	10
3.1.1.1 Map Window	10
3.1.1.2 Toolbar Window	11
3.1.1.3 Messages Window	11
3.1.1.4 Assignments Window	12
3.1.1.5 Create Filter Window	12
3.1.2 Hardware Interfaces	12
3.1.3 Software Interfaces	12
3.1.4 Communication Interfaces	13
3.2 Functional Requirements	13

3.2.1 Download Map	13
3.2.2 Zoom in/out	13
3.2.3 Move left/right/up/down	14
3.2.4 Open Messages	14
3.2.5 List Assignments	15
3.2.6 Create Filter	15
3.2.7 Show Inbox	16
3.2.8 Show Sentbox	16
3.2.9 New Message	17
3.2.10 Delete Message	18
3.2.11 Order Assignments	18
3.2.12 Describe Properties	19
3.2.11 Redesign GUI	19
3.3 Non-Functional Requirements	20
3.3.1 Performance Requirements	20
3.3.2 Design Constraints	20
3.3.2.1 Hardware Requirements	20
3.3.2.2 Software Requirements	20
3.3.2.3 Software System Attributes	21
4. Data Description	21
4.1 Data Objects	22
4.1.1 Map Data	22
4.1.2 Toolbar Data	22
4.1.3 Message Data	23
4.1.4 Assignment Data	23
4.1.5 New Message Data	25
4.1.6 Create Filter Data	25
4.2 Relationships	26
4.2.1 Map-Server Relation	26

4.2.2 Map-Toolbar Relation	26
4.2.3 Map-Assignment Relation	27
4.2.4 Toolbar-Message Relation	27
4.3 Complete Data Models and ER Diagrams	27
5. Behavioral Model and Description	32
5.1 Description of the Software Behavior	32
5.2 State Transition Diagrams	32
6. Planning	33
6.1 Team Structure	33
6.2 Estimation	33
6.3 Process Model	34
7. Conclusion	34

1. Introduction

This document contains software requirements of “Context Aware User Interface Project” of group “Korsan Yazılım” and indicates descriptions of functions and specifications of this project which will be performed by senior METU Computer Engineering students.

1.1 Problem Definition

In the present day, mobile devices are commonly used by most of the people in any kind of environment and situation. Despite the fact that uses of these devices are so widespread, unfortunately, screens of mobile devices are not legible when we consider some difficult environmental cases. Therefore; we decide that we will develop a system which identifies the environment and state changes and then adapts the graphical user interface to provide convenience to users accordingly in these extreme situations. This system will use images taken by camera of mobile devices as inputs and will process them in order to determine how user interface will change itself for usability.

We decided to customize this problem to military problem because people who generally have to deal with these bad environmental situations we speak of are generally soldiers and we implement this project by sponsorship of Aselsan. We form our problem around a soldier who is a team leader who needs to get and accomplish missions.

1.2 Purpose

Aim of this report is specification of requirements and preparation of some framework for project before getting into deep design issues. Audience for this report is Aselsan (our sponsor for this project) and teaching members of course of Computer Engineering Design.

1.3 Scope

“Yaver”, name of our product, will be a Java-based application for a mobile device, which helps accomplishing missions of a military team leader. This specific soldier will be able to get missions from command center and send messages to center through this application. Mission related information which is got from center will be accessible on the user interface. This interface will include a map and a toolbar with some operations which can be implemented on application, such as filtering map in order to show only specified buttons. There will be three sections accessible from toolbar besides some other operations. Mission handling section is for getting and listing missions. Message handling section is for receive and send messages. Lastly, create filter section is for changing settings of filters by user.

In addition to these properties, graphical user interface of this application will be **context-aware**. In other words, we are going to propose a solution for user interface adaptation for different contexts, different light conditions and different colored environments

to increase visibility of displayed information and interaction of the user interface while user is in a continuously moving state. We will develop a system which works in mobile devices to determine environmental conditions and to adapt user interface for these cases. Camera is the sensor which provides us to recognize environmental changes. This is a real-time system, since the system should determine environmental conditions immediately and change the user interface according to them.

1.4 User and Literature Survey

In Turkey, adaptable user interface is a new technology. Since mobile devices and technology of those are newly developing, this project may become pioneer and encourages innovative side of mobile-based companies.

In the world, there exist many companies work on mobile technologies, but only small percentage of them considers adaptable user interfaces. SUN, which is global supporter of Java Language, provides a library to facilitate developers to implement this type of interfaces. By using this library, after handling of events, redesigning the GUI is much easier when compared to AWT (Abstract Window Toolkit).

Synaptics, which is well-known touchpad and touch-screen producer, works for supplement Dynamic GUI's to its products. Its product called ClearPad provides Dynamic and adaptable GUI to users for Character Recognition.

When we think specifically, target audience of our application is units who involves in military operations due to fact that we will prepare an application for people who will try to achieve military missions. However, this idea of context-aware system can be implemented in many other areas.

1.5 Definitions and Abbreviations

SRS: Software Requirements Specification

DB: Database

GUI: Graphical User Interface

1.6 References

- [1] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- [2] Programming a Dynamic User Interface by Thom Blum, Doug Keislar, Jim Wheaton, and Erling Wold, March 1998
- [3] Official web-site of ClearPad which is <http://www.synaptics.com/solutions/products/clearpad>

1.7 Overview

The rest of SRS ,in appearance order, contains pages generally describe the general factors that affect the product, all software requirements, information domain for software, behavior of software, its planning and finally conclusion.

2. Overall Description

2.1 Product Perspective

Mobile devices are commonly used in the present day by most of the people. However, graphical user interfaces coming within them are 'static', that is they are not adaptive to various environment conditions. At this point, dynamic graphical user interfaces in the mobile devices would be very useful for those who work in any environmental condition and also in critical missions, such as soldiers. Korsan Yazılım fills this gap with an innovative product.

Our product, namely Yaver, is designed especially for team leaders in military units. Its outstanding feature is that it identifies the environment, state changes and then adapts the graphical user interface to increase viewability of the displayed information.

Yaver is capable of recognizing color and light density of the environment and changing the color and light of the user interface according to these recognized densities. Moreover, it can sense the motion of the user as three states: stationary, walking, running. According to this activity recognition, size of the elements of the user interface will change. User will be able select which information will be shown in the user interface in these three states of motion.

Yaver's dynamic user interface has five main parts:

1. Map window
2. Message window
3. Toolbar window
4. Assignments window
5. Create Filter window

Changes in the environment conditions will be reflected into these windows and its elements.

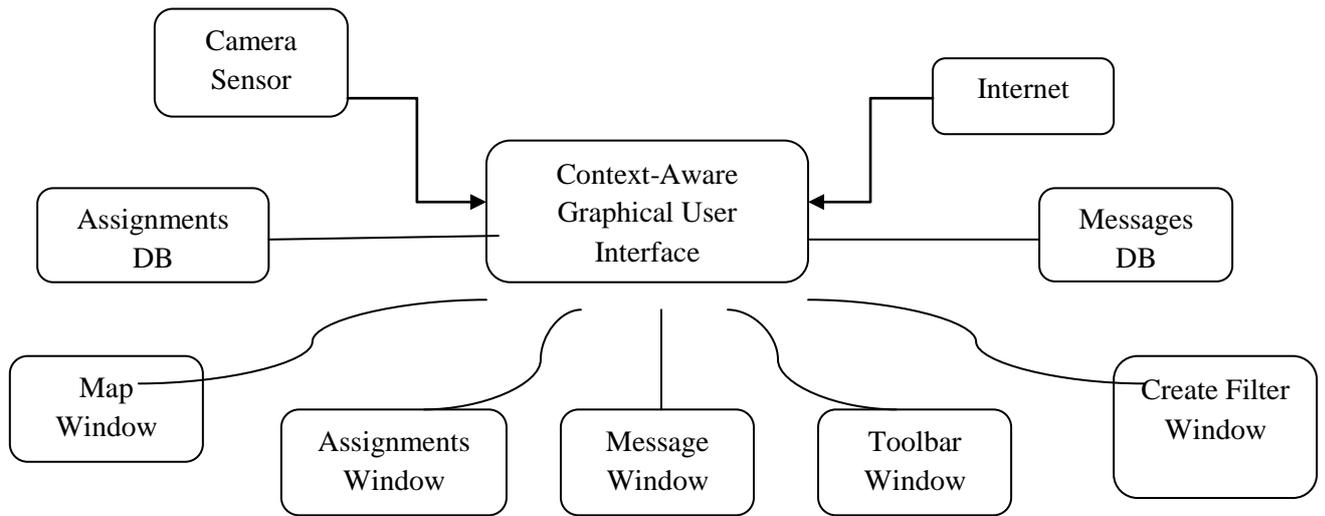


Figure1: General Structure of the Yaver

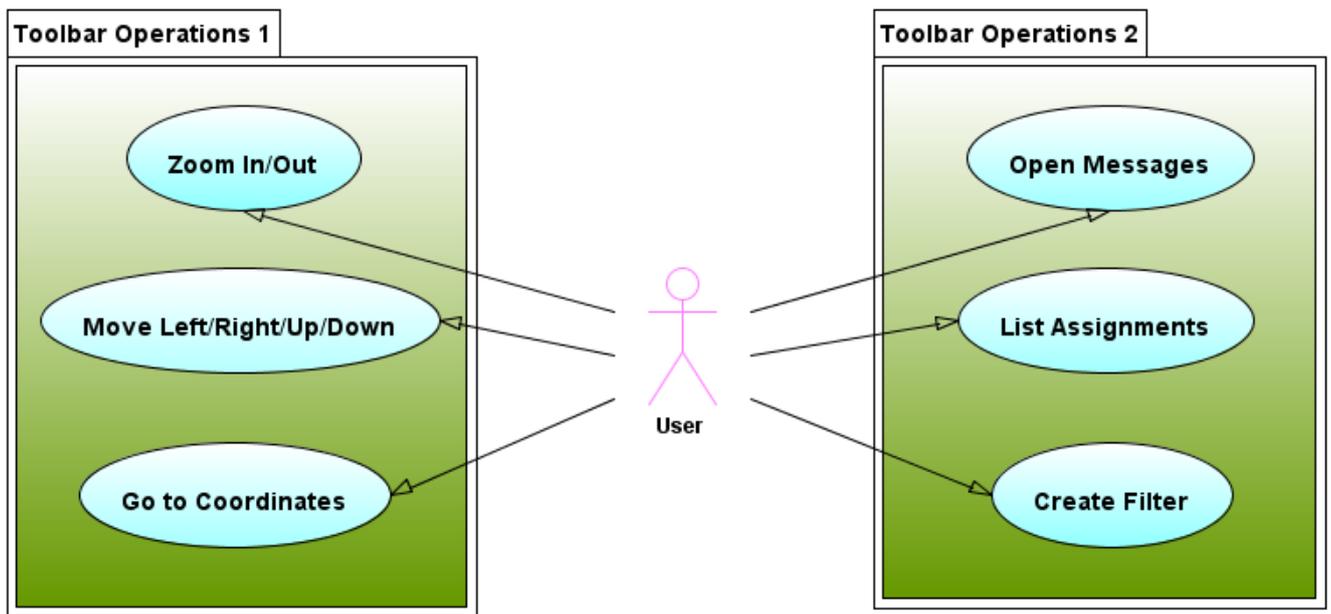


Figure2: Use Case Diagram I

2.2 Product Functions

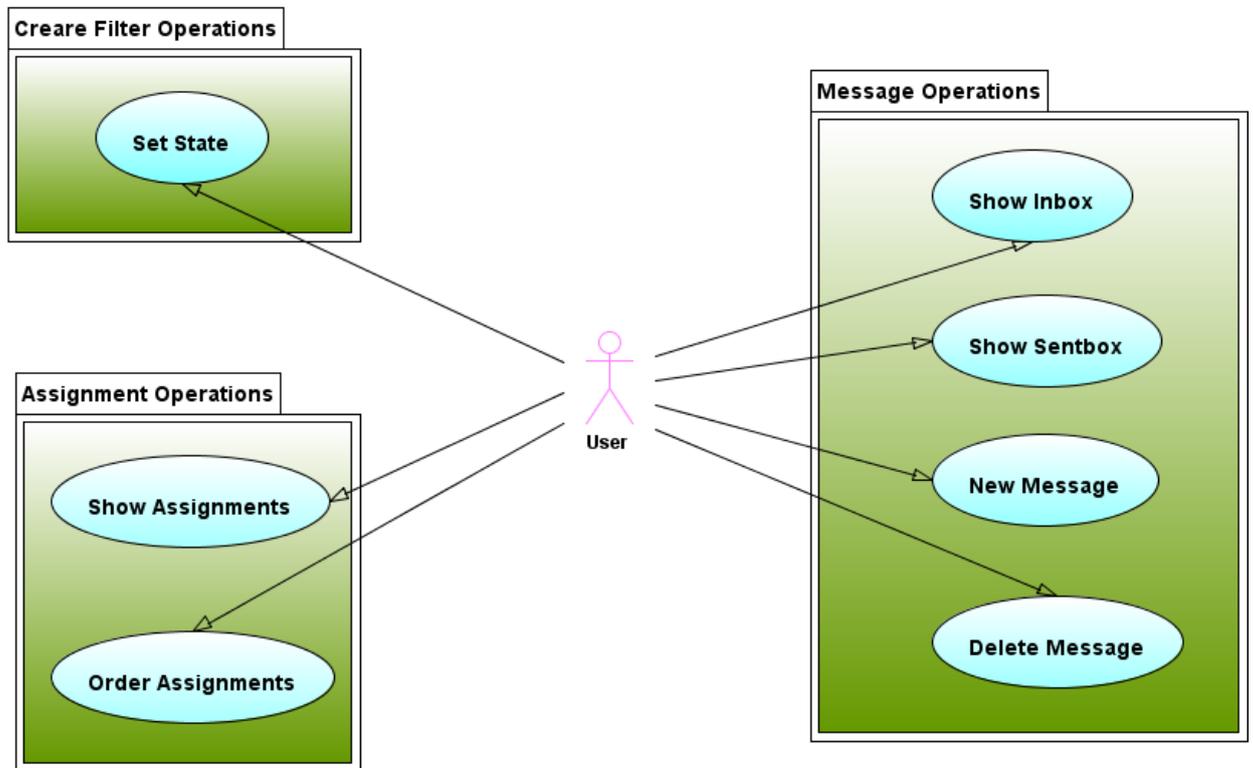


Figure3: Use Case Diagram II

2.2.1 Toolbar Operations

User can manipulate map by using the toolbar easily. Main operations for manipulating the map are zooming in- out and moving left-right-up-down. Also, user can view the region of a given coordinate by selecting the “go to specified coordinate” button.

Moreover, user can open the message window, list assignments and create filters for what is going to be shown in different modes of motion in the interface.

2.2.2 Create Filter Operations

Filtering operations are done in the create filter window. In this window, user defines filters for all the elements of the user interface. By using checkboxes user defines the state.

2.2.3 Assignment Operations

After selecting the “list assignments” option in the toolbar, user enters assignments window where h/she can view the assignments as a list. Moreover, user can order the listed assignments according to date, priority, etc.

2.2.4 Message Operations

Message operations consist of viewing the coming and sent messages, composing a new message and erasing messages. They are typically the operations done in any mailbox window.

2.3 Constraints, Assumptions and Dependencies

- The system will not track the user's position like in GPS devices. It will bring the map of the region after user enters coordinate information.
- If the user wants to view the regions that are outside the window by using "move" buttons, the map will refresh itself by bringing new info via internet connection. If internet is not available, user will be warned in less than five seconds and user will not be able to see the parts outside the region that was previously shown in the screen. When internet connection is cut, map will work in offline mode.
- The user will be able to send messages which have maximum size of 2MB.
- The application must be coded in a multi-thread way, since there will processes which work independently from user actions.
- It is assumed that the mobile device used is a PDA-like device which has bigger screen sizes than common mobile devices.
- The operations related with map are dependent to internet connection of the user has.

3. Specific Requirements

3.1 Interface Requirements

3.1.1 User Interfaces

In this part of the document the graphical user interface of the application is explained.

3.1.1.1 Map Window

The physical map of environment which user stands is shown in this window. Required information to represent the map is downloaded via the internet connection. This download operation is conducted by the system as independent from user. Specific coordinates that user wants to see can be seen by helping marker. Moreover, routes and some objects (buildings, units etc.) whose coordinates sent by command centre can be displayed on the map. User travels on the map by dragging mouse.

3.1.1.2 Toolbar Window

Buttons which are used to activate program's functions are in toolbar window. These are used for different purposes. There are several tools (buttons) in this window:

- *Zoom-in, Zoom-out tool:* This tool is used to zoom in or zoom out on the map. When the user presses this button Zoom-in or Zoom-out function is called.
- *Move left, right, up, down tool:* While traveling on the map this tool is used. Left, right, up and down are four directions which user can go. These buttons call Move function.
- *Open Messages Window button:* Messages window is opened by pressing this button. Open Messages Window function is called after the button is pressed.
- *List the Assignments button:* This button is used to show assignment list in a new window. List Assignments function is called to do this operation.
- *Go to the Specified Coordinate button:* User can determine the coordinates which s/he wants to see. There are also two text boxes for writing X and Y coordinate values next to this button. This button calls Go to Coordinate function and if it needs Download Map function.
- *Create Filter button:* User press this button if s/he wants to designate the filter which will be used for redesigning of GUI. This button opens a window used for filtering operations. Open Filter Window function is called after this button is pressed.

3.1.1.3 Messages Window

This window is a simple program resembling Thunderbird which user can read messages sent by command and send messages. There are several tools in this window:

- *Inbox tool:* User sees messages sent by command centre by using this tool with title and name of the sender. Unread messages' titles are shown as bold. Show Inbox function is called after the tool is clicked.
- *Sentbox tool:* User sees messages sent by him/her to the command centre by using this tool with title and name of the receiver. Show Sentbox function is called after the tool is clicked.
- *New Message tool:* This tool opens a new window to write a new message. New window has also some features which are one send message button, one text plain and two text boxes. Text boxes are for mail address and title. After message is written into text plain, this button is pressed and the message is sent. This tool calls New Message function.

- *Delete Message tool*: If user wants to delete a message, s/he selects message that is wanted to be deleted by clicking check box next to the message title, and then press the delete button. After this button is pressed Delete Message function is called.

3.1.1.4 Assignments Window

Assignments which are assigned by command centre via internet connection are shown in this window as a table. Uninvestigated assignments are shown as bold. Table has some features about the assignment. These are assign date, due date, description, priority, personal need, equipment need, route, target coordinate, target personal and strength of target. Also, each assignment has a name and these features are represented next to the name as a table. User can arrange assignments in order with respect to some features which are due date, priority, target coordinate and strength of target. User click on the feature name and table contents are arranged according to this entity. This operation is done by Order Assignments function.

3.1.1.5 Create Filter Window

In this window, user defines filters for all other windows buttons and entities. There are three tabs for each window in this system. Each tab has all buttons or entities in the specified window. Firstly, toolbar tab has only buttons of toolbar window, since entities are not important as buttons, and three check boxes next to name of the button. These check boxes are the same in all three tabs and they are stationary, walking and running. These indicate the status of user. If user ticks one of them or more, this means that user sees this button during only these statuses. For instance; user ticks stationary box of 'create filter' button in the toolbar, then user sees this button only in stationary mode, while running or walking s/he doesn't see this button. Secondly, tab for message window has all buttons and entities in this window with check boxes. Thirdly, Assignments tab has only entities in this window with check boxes, since there is no button there. Define State of Button is called by buttons' check boxes and Define State of Entity is called by entities' check boxes. Moreover, end of the entities or buttons list there is a check box to set filters as a default. After user selects the status of entities or buttons, s/he should press the Set Filter button.

3.1.2 Hardware Interfaces

A laptop with webcam required so that the application can run and communicate with the command centre via network connection.

3.1.3 Software Interfaces

For communication interfaces, our software must accept common interfaces which provides basic methods so that even if we simulate the wireless communication the previous code will not be changed. The simulation modules and the real modules which communicating with wireless kit must implement those interfaces that mentioned.

3.1.4 Communication Interfaces

The protocols needed in wireless communication must be implemented such that the location information, if present the user message must be encoded in wireless signal. Also the message coming via wireless connection must be parsed in order to enable visualization of the user on the map.

3.2 Functional Requirements

3.2.1 Download Map

3.2.1.1 Description

This is the first thing the system must do to show map on the map window. At the first coordinate entry made by the user, system downloads the map of the related area to the device with respect to these coordinates via internet connection. If the coordinates which is entered by user are already in the related map, the system skips this download operation.

3.2.1.2 Stimulus/Response Sequences

3.2.1.2.1 Basic Data Flow

- i. User enters the coordinate values
- ii. User clicks the 'Go to' button
- iii. The system starts automatically download operation.
- iv. The system shows downloaded map on the map window.

3.2.1.3 Requirements of the Function

REQ-1: The device must be connected to the internet.

REQ-2: The server providing the map data must be available.

3.2.2 Zoom in/out

3.2.2.1 Description

This is one of the main features of our map viewing capabilities. After the user has determined desired area by entering coordinates, at default the user sees the whole specified area. Then the user using his/her mouse or the slide-box in the tool box can zoom in or zoom out the map that had been downloaded.

3.2.2.2 Stimulus/Response Sequences

3.2.2.2.1 Basic Data Flow

- i. User clicks on the map where to zoom in.
- ii. User uses the mouse wheel on the map to zoom in or out.

3.2.2.2.2 Alternative Data Flow

- ii. User uses the slide box in the tool box to zoom in or out.

3.2.2.3 Requirements of the Function

REQ-3: The related map must be downloaded first when internet connection is present.

3.2.3 Move left/right/up/down

3.2.3.1 Description

This is one of the features of our map viewing capabilities. After the user has determined desired area by entering coordinates, at default the user sees the whole specified area. Then the user using his/her mouse or the direction buttons in the tool box can move around the map that had been downloaded.

3.2.3.2 Stimulus/Response Sequences

3.2.3.2.1 Basic Data Flow

- i. User clicks on the map.
- ii. User uses the mouse on the map to move around.

3.2.3.2.2 Alternative Data Flow

- ii. User uses the direction buttons in the tool box to move around.

3.2.3.3 Requirements of the Function

REQ-3: The related map must be downloaded first when internet connection is present.

3.2.4 Open Messages

3.2.4.1 Description

The user uses this feature to be able to manage messages window. When the user wants to read messages sent by command centre or send message to command centre, s/he clicks Messages button and reach the messages window. After the window is opened, user can use features of that window.

3.2.4.2 Stimulus/Response Sequences

3.2.4.2.1 Basic Data Flow

- i. User clicks Messages button.
- ii. A new window which has features to manage messages is popped up.
- iii. User uses these features on this window.

3.2.4.2.2 Alternative Data Flow

- iii. User click cancel button and close the window.

3.2.4.3 Requirements of the Function

REQ-4: Messages button must be present on the tool box.

3.2.5 List Assignments

3.2.5.1 Description

The user uses this feature to be able to manage assignments window. When the user wants to see assignments assigned by command centre, s/he clicks Assignments button and reach the assignments window. After the window is opened, user can use features of that window.

3.2.5.2 Stimulus/Response Sequences

3.2.5.2.1 Basic Data Flow

- i. User clicks Assignments button.
- ii. A new window which has features to manage assignments is popped up.
- iii. User uses these features on this window.

3.2.5.2.2 Alternative Data Flow

- iii. User click cancel button and close the window.

3.2.5.3 Requirements of the Function

REQ-5: Assignments button must be present on the tool box.

3.2.6 Create Filter

3.2.6.1 Description

The user uses this feature to be able to designate filters of each window's buttons and entities. When the user wants to determine filters, s/he clicks Create Filter button and reach the filter window. Initially, all status of entities and buttons are set default. After the window is opened, user can select windows and define filters according to status of her/him (stationary, walking and running).

3.2.6.2 Stimulus/Response Sequences

3.2.6.2.1 Basic Data Flow

- i. User clicks Create Filter button.
- ii. A new window which has features to designate filters is popped up.
- iii. User uses these features on this window.

3.2.6.2.2 Alternative Data Flow

- iii. User click cancel button and close the window.

3.2.6.3 Requirements of the Function

REQ-6: Create Filter button must be present on the tool box.

3.2.7 Show Inbox

3.2.7.1 Description

This is one of the features of messages window which shows messages which are sent by command centre. When the user wants to see inbox and read messages, s/he clicks the Show Inbox button and sees messages with titles. By clicking on the title of message user can read the messages.

3.2.7.2 Stimulus/Response Sequences

3.2.7.2.1 Basic Data Flow

- i. User clicks Show Inbox button.
- ii. Messages are shown with their titles, senders and dates on the window.
- iii. User selects a title which s/he wants to read and this message is seen on the window.

3.2.7.3 Requirements of the Function

REQ-7: Messages window must be already opened.

REQ-8: Show Inbox button must be present.

3.2.8 Show Sentbox

3.2.8.1 Description

This is one of the features of messages window which shows messages which are sent by user. When the user wants to see sentbox and read messages, s/he clicks the Show Sentbox button and sees messages with titles. By clicking on the title of message user can read the messages.

3.2.8.2 Stimulus/Response Sequences

3.2.8.2.1 Basic Data Flow

- i. User clicks Show Sentbox button.
- ii. Messages are shown with their titles on the window.
- iii. User selects a title which s/he wants to read and this message is seen on the window.

3.2.8.3 Requirements of the Function

REQ-7: Messages window must be already opened.

REQ-8: Show Sentbox button must be present.

3.2.9 New Message

3.2.9.1 Description

The user uses this feature of messages window to write a new message. For doing this, user clicks the New Message button and a new window to write a message is popped up. This window has text plain and text boxes to write a message text, title and address. After user write message text, s/he should press the Send button to send the message.

3.2.9.2 Stimulus/Response Sequences

3.2.9.2.1 Basic Data Flow

- i. User clicks New Message button.
- ii. A new window is popped up.
- iii. User writes address which message is sent to.
- iv. User writes title of message.
- v. User writes message text.
- vi. User presses the Send button and the message is sent.

3.2.9.2.2 Alternative Data Flow

- vi. User clicks cancel button and close the window.

3.2.9.3 Requirements of the Function

REQ-7: Messages window must be already opened.

REQ-9: New Message button must be present.

3.2.10 Delete Message

3.2.10.1 Description

The user uses this feature of messages window to delete a message. For doing this, user first enters the Inbox or Sentbox and chooses a message by ticking check box next to the title of message and then press the Delete button. Afterwards, that selected message is no more seen on window.

3.2.10.2 Stimulus/Response Sequences

3.2.10.2.1 Basic Data Flow

- i. User opens Inbox or Sentbox.
- ii. User selects a message by ticking check box next to title.
- iii. User presses the Delete button.
- iv. The selected message is removed message database.

3.2.10.3 Requirements of the Function

REQ-7: Messages window must be already opened.

REQ-10: Inbox or Sentbox must be present on the window.

3.2.11 Order Assignments

3.2.11.1 Description

This is the only feature of assignments window that is used by user to order assignments with respect to some entities. These entities which are used to order assignments are due date, priority, target coordinate, and strength of target. User clicks one of them on the assignments table and assignments are arranged with respect to clicked entity.

3.2.11.2 Stimulus/Response Sequences

3.2.11.2.1 Basic Data Flow

- i. User opens Assignments window.
- ii. User clicks an entity to order assignments.
- iii. Assignments are arranged according to this clicked entity.

3.2.11.3 Requirements of the Function

REQ-11: Assignments window must be already opened.

REQ-12: At least more than one assignment must be assigned and present.

3.2.12 Describe Priorities

3.2.12.1 Description

This is a major feature of the system in assignments part to designate filters of window. In the Assignments window, user ticks check boxes next to entity name or button name at which status s/he wants to see them. After choosing statuses of button and entity, user presses the Set Filter button and filters are designated.

3.2.12.2 Stimulus/Response Sequences

3.2.12.2.1 Basic Data Flow

- i. User opens Create Filter window.
- ii. User clicks check boxes next to entity or button names which are wanted to define the visibility at statuses (stationary, walking and running).
- iii. User presses Set Filter button and filters are set.

3.2.12.3 Requirements of the Function

REQ-13: Create Filter window must be already opened.

3.2.13 Redesign GUI

3.2.13.1 Description

This is the most important feature of the system which is done automatically. The images taken by camera are processed and environmental conditions are described according to this operation. For example, there are three statuses of user motion; they are stationary, walking and running. Moreover, the amount of light in this environment is determined by processing images. After these are determined, the system changes the color of GUI in order that user see comfortably GUI. For instance; if the environment is dark, the system brighten the GUI. Furthermore, the system redesigns the GUI such as button visibility and size by combining the information which is taken from Create Filter window.

3.2.13.2 Stimulus/Response Sequences

3.2.13.2.1 Basic Data Flow

- i. User runs the application.
- ii. User opens Create Filter window and sets the filters.
- iii. System processes the image taken from camera and redesign GUI by combining filter information.

3.2.13.2.2 Alternative Data Flow

- iii. System processes the image and redesign GUI with using filter information defined by user or default filter.

3.2.13.3 Requirements of the Function

REQ-14: Camera must be connected to the device which program runs on.

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

Performance is an important issue for our-system because our aim is to redesign GUI dynamically according to environment. Our database should be big enough to keep map information, messages and assignment list. However our serve time should also be fast. Moreover, software should be fast enough to respond in real time. Furthermore, Internet connection speed of device should be high enough to get assignments and messages from command centre and to download map information instantly.

3.3.2 Design Constraints

3.3.2.1 Hardware Requirements

- The device shall have ... MB of disc capacity to store the map of ... km² area
- The device shall have at least 667 MHz processor to process the data efficiently, fire the map-related-functions, and run the related Java applications.
- The device shall have at least 512 MB of RAM.
- The device shall have an appropriate graphic processor unit to visualize the map.
- The device shall support wireless technology to communicate and download the map information.
- The device shall have a camera sensor to take image and to describe environmental condition.

3.3.2.2 Software Requirements

- The device shall have an operating system installed, which can be Windows or Linux.
- The device shall have our software installed.
- The device shall support JDK 1.5 and later environment.
- The device shall have some external libraries such as Swing (expected to grow while development)

3.3.2.3 Software System Attributes

i. Security:

Network communication should be made in encrypted way in order to avoid network attacks in the system. Therefore, assignments and messages are sent after data is encrypted. This system guarantees that any information will not be stolen by unauthorized people.

ii. Maintainability :

The application should be banned for modification by users and allowed for extensions by developers. It should be implemented in a way that modification will not be necessary. However it should be open for extensions such as keeping database for the past locations of the users, newer network connection technologies or its simulations, newer communication technologies. In other words, our system should accept common interface for all service it uses, the new ones should implement these interfaces in order not to modify previously coded parts.

iii. Portability:

The application should be platform independent. It should work in Windows and Linux successfully and also in mobile devices. By using Java for implementing this application we handle most of this issue.

iv. Availability/Reliability:

The application should represent all the information which comes from server in the map, message and assignment sections instantly, allowing dynamic changes whenever it executes. User and command center should be available to communicate with command center. Application guarantees that wrong information comes from users will not cause software crashes.

v. Scalability:

Since available network management algorithms are designed to work on fixed or relatively small wireless networks and our communication technology has not been decided yet the exact number can vary. Our application is for using of one military team leader. However, this system is not an application such that many users can not connect to device simultaneously. Application is suitable for all soldiers who are commanded.

4. Data Description:

This section describes attributes of data objects and relationship between them with a data dictionary and tables to overcome confusions. These data objects are made under the consideration of getting rid of unnecessary attributes and normalization factors.

4.1 Data Objects

4.1.1 Map Data

At this table, system takes maps from server and after filtering with particular filters, shows them in GUI. Since this table always needs server interaction, we attempted to create as less attribute as we could. Some web companies such as Google provides map to use in their application. In order to take maps from servers of the companies, some information has to be sent. Coordinate values of the current position is primary key value to take maps from servers. In addition, Route of the current assignment information (coordinates of the assignment) is also seen on map.

Entities

CoordinateMap (Primary)

RouteCoordinates

Name:	CoordinateMap
Description:	Represents coordinate values of any point that showing on map. Area which centered by this coordinate is displayed. Initially, it is set coordinate chosen by user.

Name:	RouteCoordinates
Description:	Represents set of coordinate values of assigned route. Each coordinate is displayed on Map and connected with respect to the assigned order.

Table1 : Entities of Map

4.1.2 Toolbar Data

At this table, system waits for an input to change its current state. Users can change coordinate values by using move left, move right, move up, move down. Therefore, this new coordinate values are set coordinate of Map objects. Similarly users can change visibility amount of map by using zoom in, zoom out features. In order to change this feature, Factor can be incremented or decremented with special values with respect to the zoom in or zoom out.

Entities

UpdatedCoordinate (Primary)

Factor

Name:	UpdatedCoordinate
Description:	Represents updated version of coordinate due to Move Left, Right, Up, Down. It's also assigned to CoordinateMap variable of Map Object.

Name:	Factor
Description:	Represents a multiplier to display Map. It is changed by user with Zoom

	In/Out.
--	---------

Table2 : Entities of Toolbar

4.1.3 Message Data

At this table, users can read taken message or send a message to the command center. It's so similar to general purpose mail program. Coming messages are listed and user chooses whatever s/he wants. These messages are stored at internal database. Users can delete message from database using Message interface. Coming messages are listed with respect to the priority by default.

Entities

ReachDate (Primary)

ReachTime (Primary)

Sender

TitleOfComing

Priority

ContentsOfComing

Name:	ReachDate
Description:	Represents date of the incoming message.

Name:	ReachTime
Description:	Represents time of the incoming message.

Name:	Sender
Description:	Represents address or name of the sender of incoming message.

Name:	TitleOfComing
Description:	Represents title of incoming message.

Name:	Priority
Description:	Represents a value assigned by sender which shows priority of the incoming message. Messages are listed with respect to the priority.

Name:	ContentsOfComing
Description:	Represents body of the incoming message. It consists of set of characters.

Table3 : Entities of Message

4.1.4 Assignment Data

Command center can assign a duty to user. When a duty is assigned, then user sees properties of this assignment by using this application. Each assigned assignments are listed.

User can change order of the listed assignments with respect to the entities. Priority is default entity to order all uncompleted assignments.

Entities

Time:

AssignDate (Primary)

AssignTime (Primary)

DueDate

DueTime

Description:

AssignmentDescription

Priority

PersonalNeeded

EquipmentNeeded

RouteInfo

Target:

TargetCoordinate

TargetPersonal

StrengthOfTarget

Name:	AssignDate
Description:	Represents assign date of duty.

Name:	AssignTime
Description:	Represents assign time of duty.

Name:	DueDate
Description:	Represents finish date of the duty.

Name:	DueTime
Description:	Represents finish time of the duty.

Name:	AssignmentDescription
Description:	Represents description and some useful information of related to duty.

Name:	Priority
Description:	Represents a value assigned by command center which shows priority of the duty. Assignments are listed with respect to the priority.

Name:	PersonalNeeded
Description:	Represents expected count of personal to finish assigned duty.

Name:	EquipmentNeeded
Description:	Represents expected count and properties of equipments to finish assigned

	duty.
Name:	RouteInfo
Description:	Represents information of objects and obstacles on the route
Name:	TargetCoordinate
Description:	Represents coordinate value of target.
Name:	TargetPersonal
Description:	Represents count of personal of target.
Name:	StrengthOfTarget
Description:	Represents strength of target.

Table4 : Entities of Assignment

4.1.5 New Message Data

. In order to compose a new message and send it, user chooses new message part. New message part consists of three basic features which are address information, title and contents of the message.

Entities of New Message

SendDate (Primary)

SendTime (Primary)

Address

TitleOfNewMessage

ContentsOfNewMessage

Name:	Address
Description:	Represents address of the message to be sent.
Name:	TitleOfNewMessage
Description:	Represents title of the message to be sent.
Name:	ContentsOfNewMessage
Description:	Represents body of the outgoing message. It consists of set of characters.

Table5 : Entities of New Message

4.1.6 Create Filter Data

Main purpose of the project is being awareness on environmental changes and then dynamically redesigning GUI to provide more usability. In order to do this, some properties are hidden or sizes of the some properties are changed. Default behavior is identified but user

can change this behavior for own purposes. At this part, user creates own filter and chooses behavior of the properties.

Entities

ActiveTab

All other entities and buttons.

Name:	ActiveTab
Description:	Represents which tab is active tab and visualized on GUI. Tabs are Message, Assignment and Toolbar.

Table6 : Entity of Create Filter

4.2 Relationships

4.2.1 Map – Server Relation (isOnline)

Server sends map with respect to the coordinate value. This relation has an extra entity called isConnected, which shows status of server connection.

Entity

CoordinateMap (Primary)

isConnected

Name:	isConnected
Description:	Represents status of network connection.

Table7 : Entity of isOnline Relation

4.2.2 Map – Toolbar Relation (UpdateMap)

Toolbar contains some map-related functions such as Zoom in/out, move left, right, up, down. By using these functions, maps vision capability and area is changed. If obtained coordinate values by using move functions are out of the downloaded map, system wants new map by updated coordinates. This relation has an extra entity called isOutOfBound. It's value depends on whether UpdatedCoordinate is in or out of map.

Entity

CoordinateMap (Primary)

UpdatedCoordinate (Primary)

isOutOfBound

Name:	isOutOfBound
Description:	Represents UpdatedCoordinate is whether in downloaded map or not.

Table8 : Entity of updateMap Relation

4.2.3 Map – Assignment Relation (ShowRoute)

Route of the current assignment and target coordinate is showed on map.

Entities

CoordinateMap (Primary)

AssignDate (Primary)

AssignTime (Primary)

IsAssignmentCompleted

Name:	IsAssignmentCompleted
Description:	Represents current assignment is finalize or not. It is set automatically when due time is reached.

Table9 : Entity of showRoute Relation

4.2.4 Toolbar – Message Relation (ShowNotification)

When new message comes from command center, a notification is seen on toolbar, and user reaches this message by choosing this notification.

Entities

UpdatedCoordinate (Primary)

ReachDate (Primary)

ReachTime (Primary)

HasNewNotification

Name:	HasNewNotification
Description:	Represents existence of a new message or new assignment.

Table10 : Entity of showNotification Relation

4.3 Complete Data Models and ER Diagrams

4.3.1 Map Data Model:

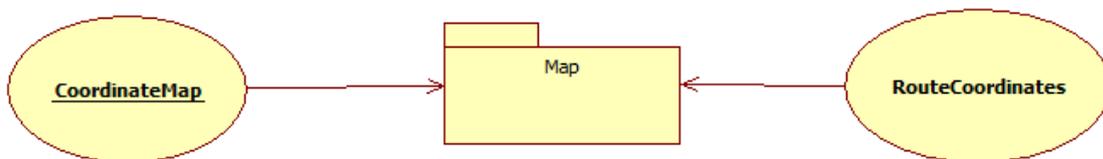


Figure4: Map Data Model

4.3.2 Toolbar Data Model:

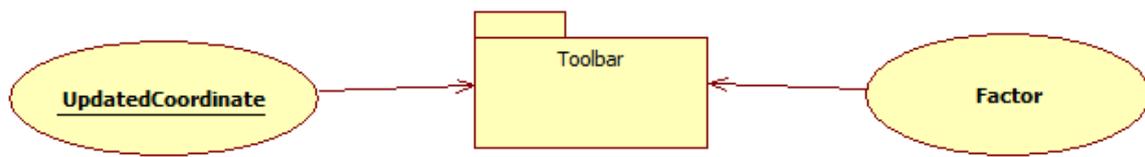


Figure5: Toolbar Data Model

4.3.3 Message Data Model:

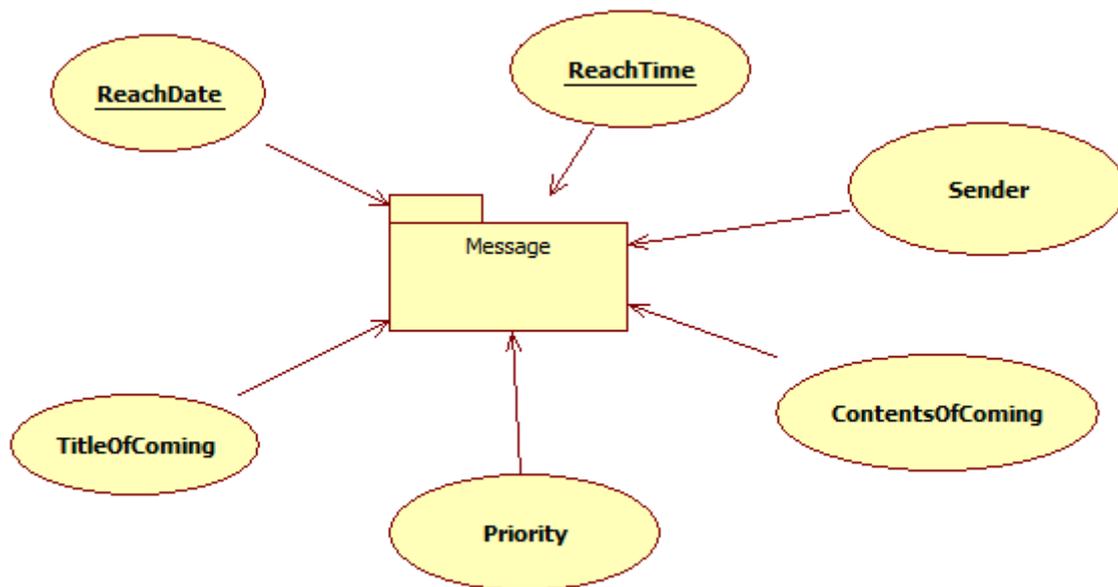


Figure6: Message Data Model

4.3.4 Assignment Data Model:

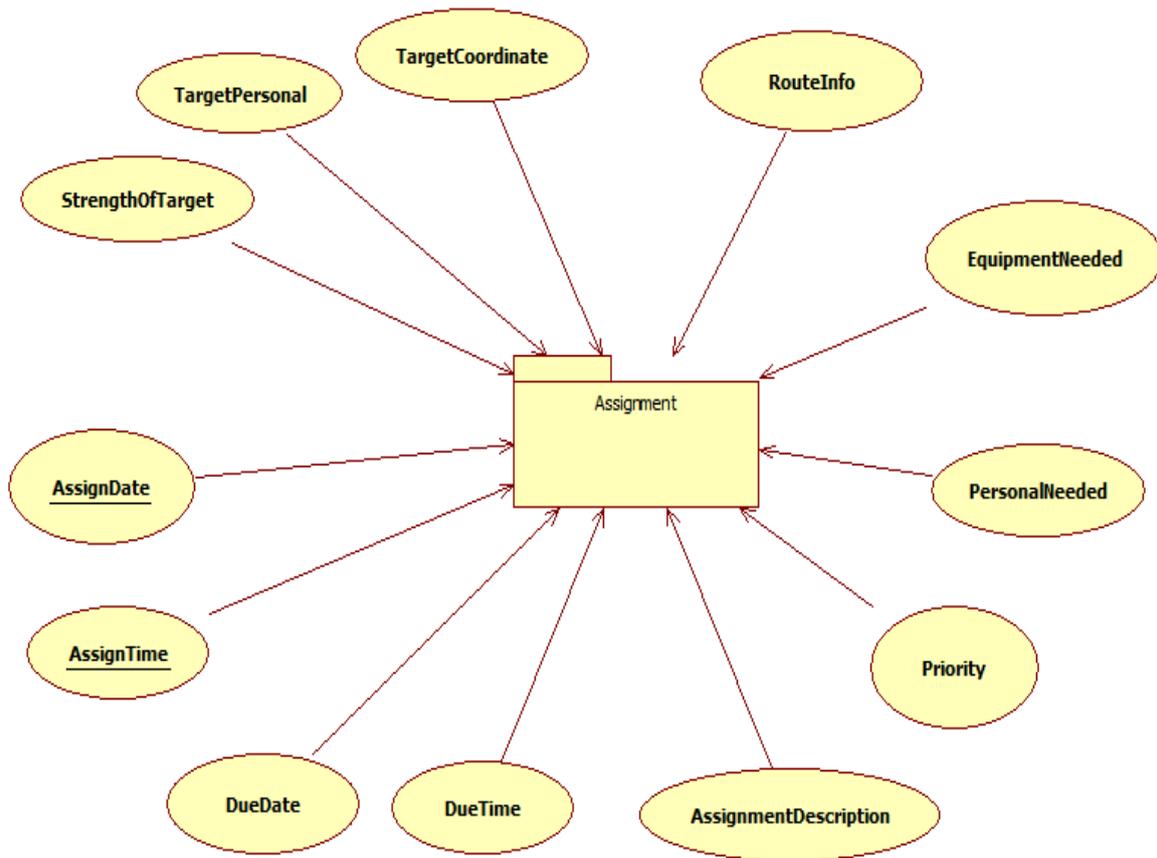


Figure7: Assignment Data Model

4.3.4 New Message Data Model:

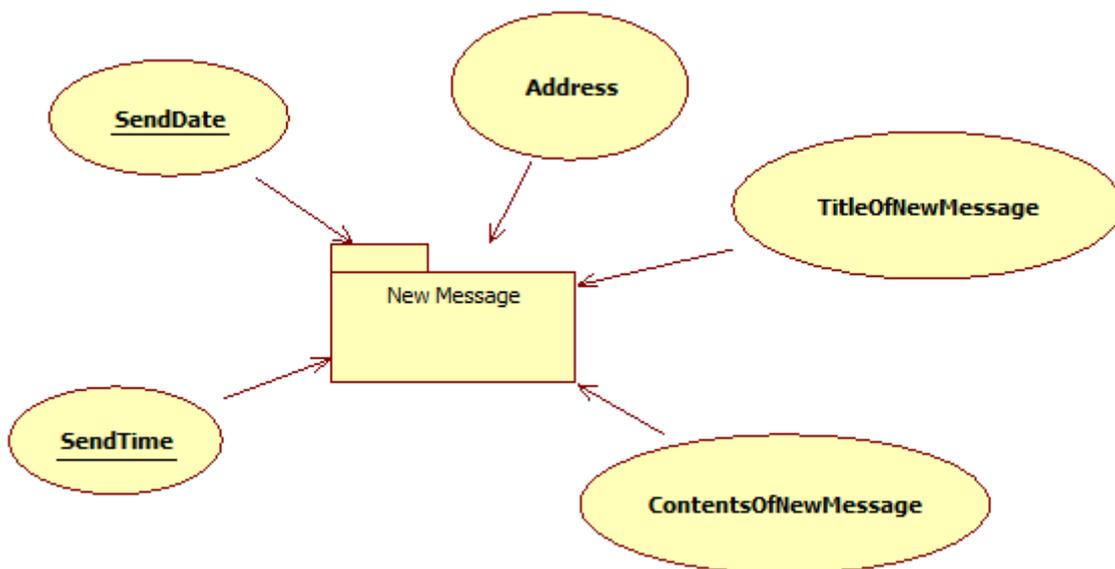


Figure8: New Message Data Model

4.3.5 Create Filter Data Model:

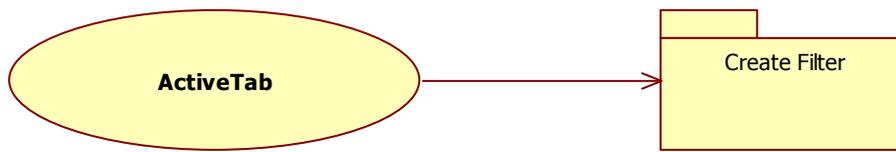


Figure9: Create Filter Data Model

4.3.6 isOnline ERD:

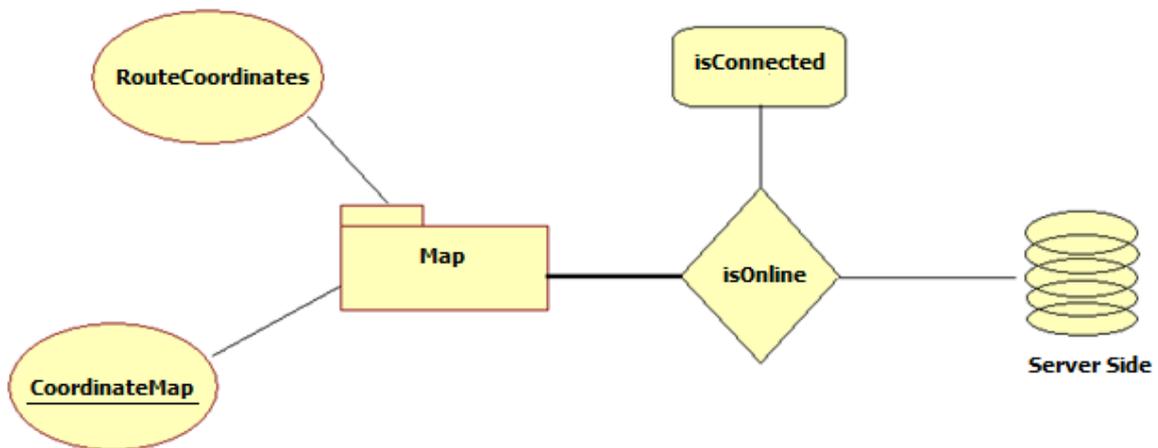


Figure10: isOnline ER Diagram

4.3.7 UpdateMap ERD:

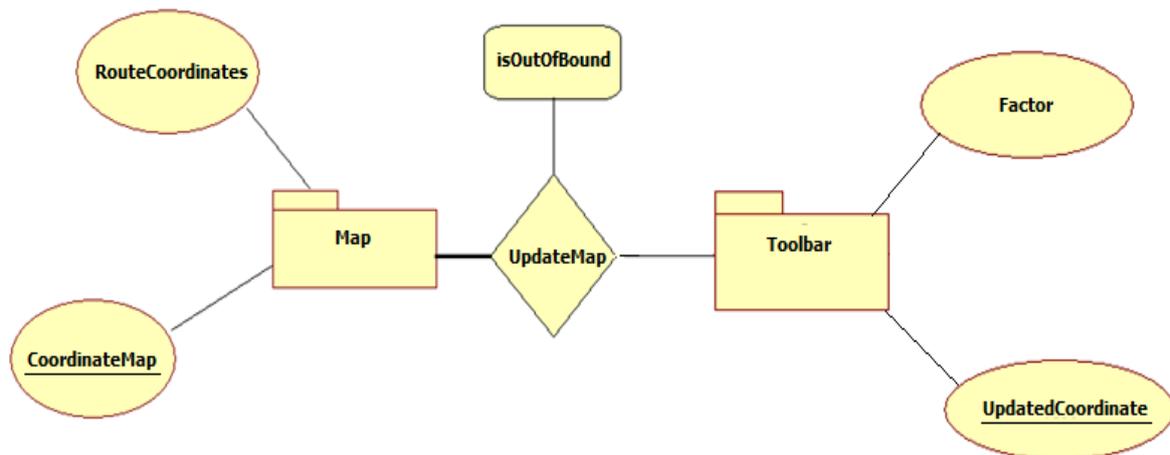


Figure11: updateMap ER Diagram

4.3.8 showRoute ERD:

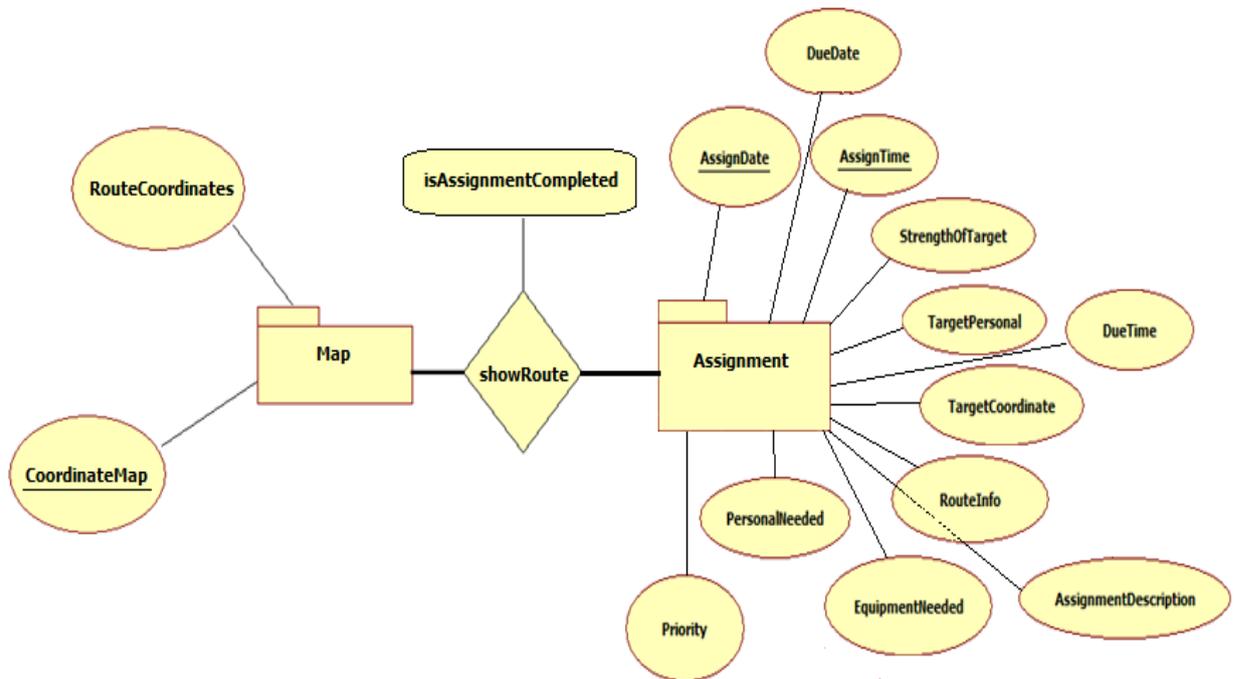


Figure12: showRoute ER Diagram

4.3.9 showNotification ERD:

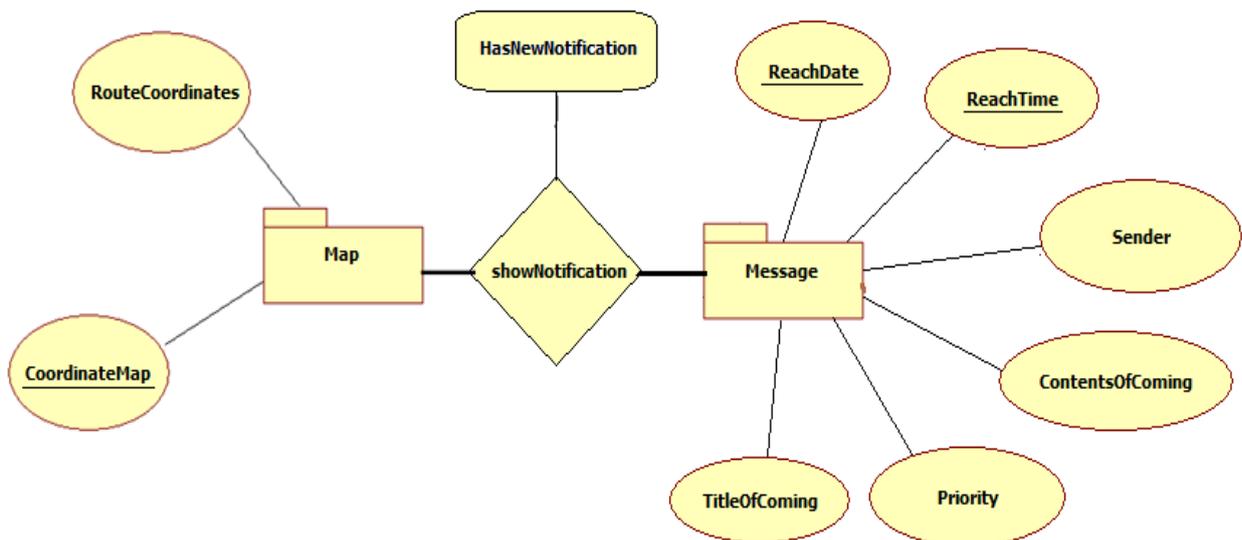


Figure13: showNotification ER Diagram

5 Behavioral Model and Description

5.1 Description of the Software Behavior

When the software is launched in the mobile device, user enters the standby state immediately. In this state user will see an interface which has a map and toolbar. When user gives a coordinate to the system, the map will be brought via internet connection and show some previously selected information on the map. Also, the user interface adapts itself according to the environment and user's motion independently from users actions, that are system will automatically redesign the graphical user interface according to the data taken from camera sensors. These two operations are done by the system's itself.

In the standby state, user has options to do operations related with toolbar, map, messages, filters and assignments. When user uses one of the buttons in the toolbar, the system enters the toolbar operations state and system will response back which button was initialized. According to this response, other state transitions will be made. The following table given below shows these transitions in a detailed way:

Previous State	Condition	Next step
Start	None	Standby
Standby	If sensor data is received	Adapt
Standby	If coordinates are given	Update
Standby	If toolbar buttons are activated	Toolbar Operations
Standby	If map related element is active	Map Operations
Standby	If filter related element is active	Filtering Operations
Standby	If message related element is active	Message Operations
Standby	If assignments related element is active	Assignment Operations
Standby	If application is terminated	Finish
All next states except finish	If demanded operation is done	Standby
All states	If any failure occurs	Standby(warn user)

Table11 : Software Behavior Matrix

5.2 State Transition Diagrams

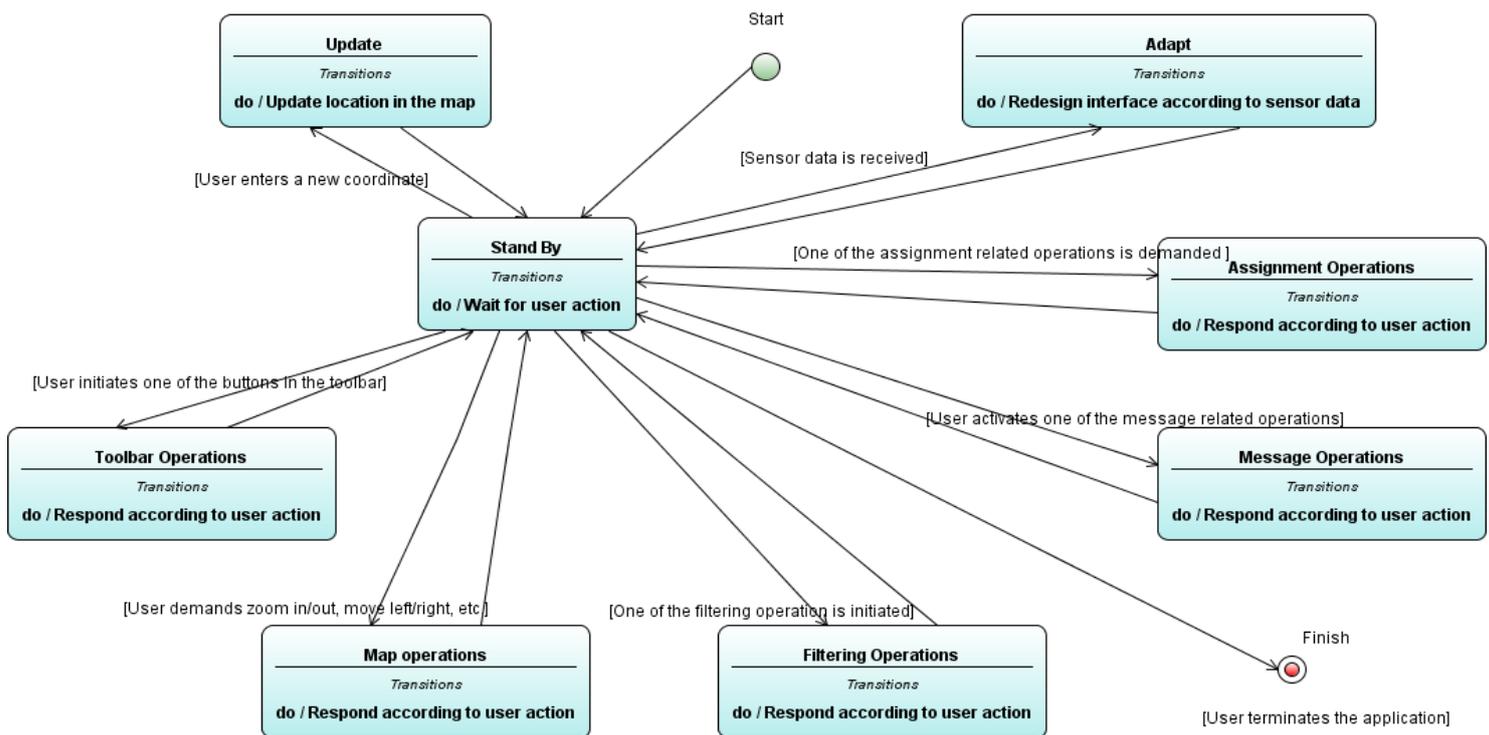


Figure14: State Transition Diagram

6. Planning

6.1 Team Structure

Ali Karakaya : Optimist, Initiator
 Aydın Atay : Devil's Advocate, Summarizer
 Oğuzhan Onkal : Gate Keeper, Devil's Advocate
 Yunus Emre Işıklar : Recorder, Time Keeper

6.2 Estimation (Basic Schedule)

	December			January				February			March			April		May						
	6-12	13-19	20-26	27-2	3-9	10-16	17-23	24-30	31-6	7-13	14-20	21-27	28-6	7-13	14-20	21-27	28-3	4-10	11-17	18-24	25-1	2-31
	Learn Java Swing				Implementing GUI Functions				Image Handling			Camera Handling										
All Team Members	Prepare Initial Design Report		Design look of GUI		Prototype Demo		Simulate mission, message and map handling				Solve Performance Issues		Software Demo		Final Presentation							
	Get ready for Team Presentation		Prepare Detailed Design Report		Make GUI changeable by using a menu for developers				Test Application													

6.3 Process Model

We will follow an incremental process model. We will do requirement analysis and design, then we will implement a prototype. Afterwards we will gradually increase the quality of the product. We decided that Spiral Model best suits our needs.

7. Conclusion

Preparation of this SRS, for determining requirements and functions which we will use in application gave us a valuable experience, ideas and clear boundaries about how to implement our application. This document is not strict, i.e. some parts may change. Lots of the parts will be mentioned and summarized at initial and detailed design documents.