Software Requirements Specification (SRS)

By Tolle Sudore

Mehmet Çağdaş AKAR – 1630524

Çağrı ASLANBAŞ – 1630607

Uğur BALTACI - 1559962

Cem EKİCİ - 1560119

Ceng491
05.12.2010

Table of Contents

Table of Figures

1. INTRODUCTION

This Software Requirements Specification is about the project named "Controlling Mobile Devices via Gesture Recognition" (CMDGR) and is written by using the IEEE 830-1998 model.

1.1. Problem Definition

With the improvement of technology, Human Computer Interaction (HCI) is getting more and more important. In order to interact with mobile devices and computers, static keys/buttons, track path devices and touch screens have developed respectively. We believe that the next concept of HCI will let people use computers or mobile devices without touching. Gesture recognition and motion tracking are the next improvement of interaction with mobile devices.

Controlling Mobile Devices is not an easy task while driving car or mobile device is located away from user via touching screen or pushing button. Our Gesture Recognition System will be the solution of getting out of an obligation of touching the screen or pushing the buttons of mobile phones in order to make a call, control the menu, take a picture etc. By detecting the hand movements of human body, there will be no necessity of touching on mobile phones.

We know that some applications related with gesture recognition are available for computers. However, current mobile devices' processors are not capable of handling this heavy work. We know that smartphones will have more powerful "new generation" processors (like Intel Atom) in near future. Due to all of these reasons, Controlling Mobile Devices via Gesture Recognition (CMDGR) will be a breakthrough innovation.

1.2. Purpose

Controlling Mobile Devices via Gesture Recognition will be implemented for mobile phone users in the world. The aim of this system is to dissolve the interaction of human hand with their mobile phones by recognizing the hand movements of the mobile phone user without touching. This system will also provide calling, sending a message, playing game, controlling the menu, managing media etc. This provides mobile phone users to manage their mobile phones easily, quickly and hands free.

1.3. Scope

Controlling Mobile Devices via Gesture Recognition System will be used on mobile phones. Because of the usability of the gesture recognition, this idea and its algorithms can be used in many other areas like game consoles, tablet PC's etc.

1.4. User and Literature Survey

All literature surveys which are given below are working on computers and also related to computers and working on high-speed processors. And also none of them uses the technique that will be used in this software. The examples market research stated above are working on only PCs

and do not give a good result. Moreover, they use extra equipment for fully functioning. As we said above, hopefully our innovative solution opens new market area for mobile devices.

- **GestureTek - 3D Depth Sensing Tracking Systems:**

GestureTek is a well-known company specialized on pattern recognition. GestureTek's depth tracking software enables users to control onscreen interaction with simple hand motions instead of a remote control, keyboard or touch screen. But this technology is used on personal computers. Also, another disadvantage of this system is that the 3D sensing is made by single depth sensing camera. Depth sensing is made by a laser, which makes the cost of the camera expensive.

- **Prime Sense – The PrimeSensor Technology**

Prime Sense's PrimeSensor technology is used to recognize gestures by getting depth information with a single camera and an IR sensor. The PrimeSensor Reference Design is built around PrimeSense's PS1080 system on a chip (SoC). The PS1080 SoC houses extremely parallel computational logic, which receives an infrared pattern as an input, and produces a VGA-size depth image of the scene.

The potential users of Controlling Mobile Devices via Gesture Recognition, as it appears with its name, will be mobile phone users. Since the system will be so easy to use, so easy that the user event won't need to touch anything, the mobile phones that using this system will be a suitable choice for all users.

- **NIPPON SYSTEMWARE's Gesture Recognition Software: DigInfo**

NSW, a company that develops computer hardware, software and various IT solutions, has developed Gesture Recognition Software which can recognize the shape and color of the operator's hand and can differentiate the movement of the hand from other moving objects in the background. NSW is using this as an input method for controlling computers via cursor as the hand moves.

- **Evoluce: Evoluce ONE**

Evoluce, a German company pioneering display technology, has developed Evoluce ONE, a 47-inch Multi-touch HD LCD Screen with gesture recognition which enables users to control multi-touch applications through gestures made right above the screen, without touching it. Hand gestures such as mid-air scrolling, rotating, stretching, shrinking, or pivoting motions are immediately registered and translated into screen actions such as pinch and zoom, screen transition or application dismissal.

1.5. Definitions and Abbreviations

Term	Definition
Visual Depth Information	Proper visual recognition of depth or the relative distances to different objects in space
Gesture Recognition	Mathematical interpretation of a human gesture by a computing device
Stereovision	Method that the human visual system uses to perceive depth
SRS	Software Requirements Specification
CMPGR	Controlling Mobile Phones via Gesture Recognition
GRE	Gesture Recognition Engine
MeeGo	A Linux-based open source mobile operating system, implemented by Intel and Nokia

1.6. References

1. IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

2. Custom 3D Depth Sensing Prototype System for Gesture Control, http://www.gesturetek.com/3ddepth/introduction.php

3. Prime Sense, http://www.primesense.com/?p=488

4. NIPPON SYSTEMWARE's Gesture Recognition Software: DigInfo, May 14th 2010. Retrieved from http://newtechs.net/2010/05/nippon-systemwares-gesture-recognition-software-diginfo/

5. Evoluce ONE Multi-Touch LCD Screen, http://www.evoluce.com/en/products/multi-touch_lcd_screen.php

1.7. Overview

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Specific Requirements section of this document is written primarily for the developers and describes in technical terms, the details of the interfaces, functional and non-functional requirements of the system.

The fourth and fifth chapters are written in order to describe the Data Model and Behavioral Model of Controlling Mobile Phones via Gesture Recognition (CMPGR), respectively.

Planning, which is described in the sixth chapter, will give information about team structure, schedule from month to month and process model for the Project.

2. OVERALL DESCRIPTION

This section of Software Requirements Specification (SRS) describes all general factors of the product and its requirements.

2.1. Product Perspective

Controlling Mobile Phones via Gesture Recognition is a system for mobile phones, which will be an innovation for current technologies like multi-touch, tracking system etc. Initially, the mobile phone user will be on main menu, right after he/she unlocked the key lock. Then there will be 5 choices for the user to select, which are “contacts”, “messages” and “media”, respectively. Wandering around these menus, submenus and making the last action will be made by special hand movements.

“Contacts” and “messages” menus are the two choices that the user interacts with the receiver by using them. User can call someone by selecting the receiver from the contacts list or can send a message (SMS, MMS or E-Mail) to someone using “messages” menu.

“Media” menu will cover all of the personal archive of the user like photos, videos, music. By special hand movements, the user can wander around these archives, listen to music and even can zoom-in or zoom-out the pictures. CMPGR System diagram can be seen on Figure-1.

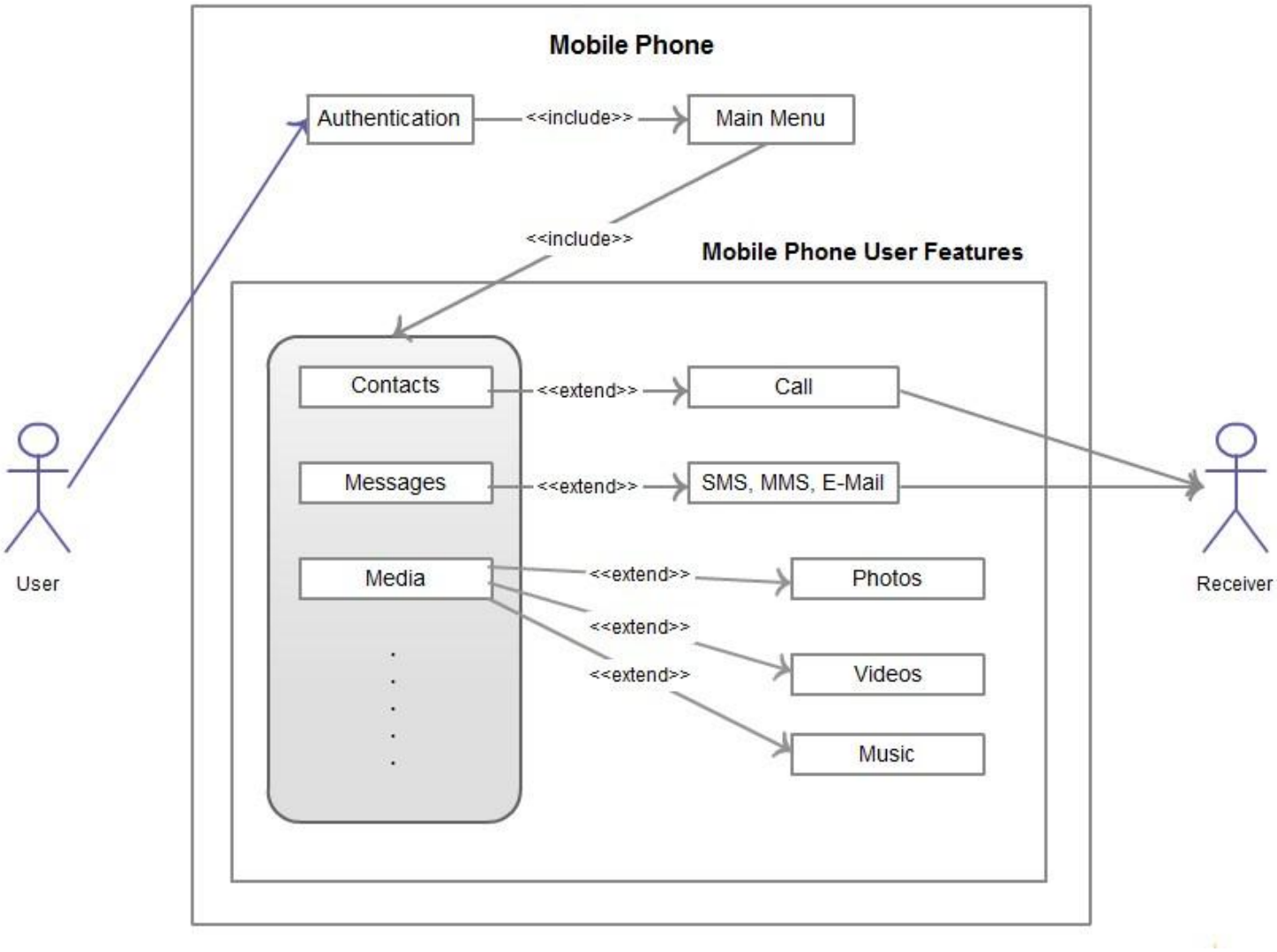

Figure-1. Use case diagram of CMPGR

2.2. Product Functions

CMPGR consists of two main parts, which are related strongly to each other: Gesture Recognition Engine and User Interface Controlling. In the meantime these two parts can also be used for other projects separately. The following functions are;

GESTURE RECOGNITION ENGINE – Track, HandLocation, HandGesture

Gesture Recognition Engine is the first part of the project. It is the first branch of the mobile devices control system. Tracking hand movements via cameras, hand location of recognized movements and gesturing of these hand movements to control the mobile devices is done by the engine. GRE is the most important part of our project. It will consist of lots of pattern classification and image processing algorithms. Below are the use case diagrams of the following functions:

***Track*:**

Movements of the user are tracked by gesture recognition engine.

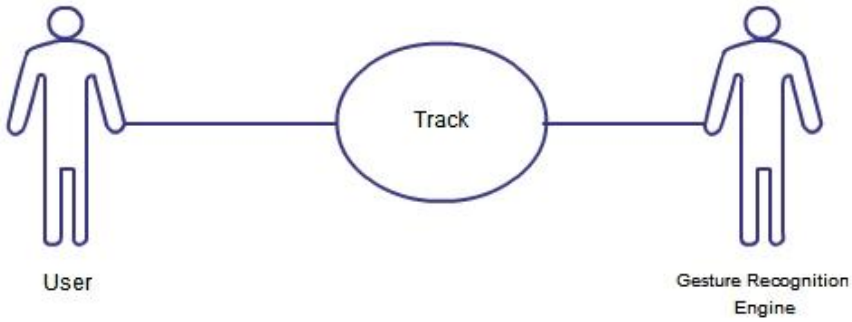

Figure-2 Track function

***HandLocation*:**

Location of the hand is sent to user interface controller from gesture recognition engine.

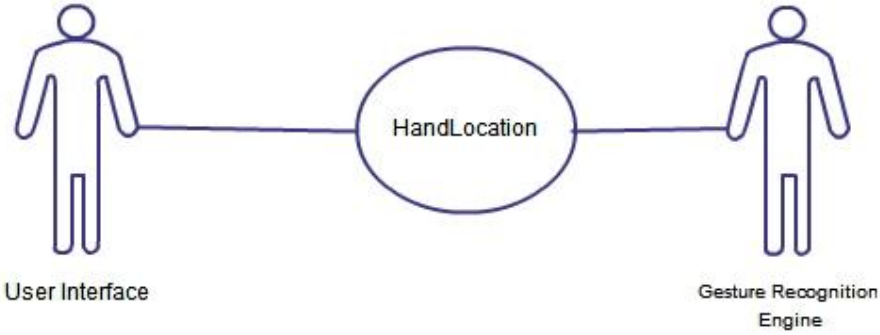

Figure-3 HandLocation function

HandGesture:

Gesture of the hand is sent to user interface controller from gesture recognition engine.

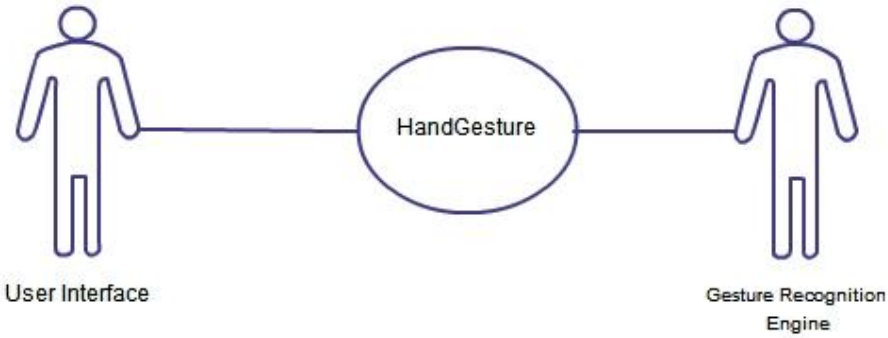

Figure-4 HandGesture function

USER INTERFACE CONTROLLER – UnlockScreen, Select, Quit, Next, Previous, Call, EndCall, Approve

User Interface Controller function is the last part of the project. It is the second branch of the mobile devices control system which has following functionalities. Unlocking the screen after the face of the user is recognized, selection of the desired menu, quitting from the menu, getting the next menu or the previous menu, approval of the action, calling a selected contact and ending the call are done by the user's special hand movements. Below are the use case diagrams of the following functions:

UnlockScreen:

Unlocking of the screen is done after the recognition of the face of the user.

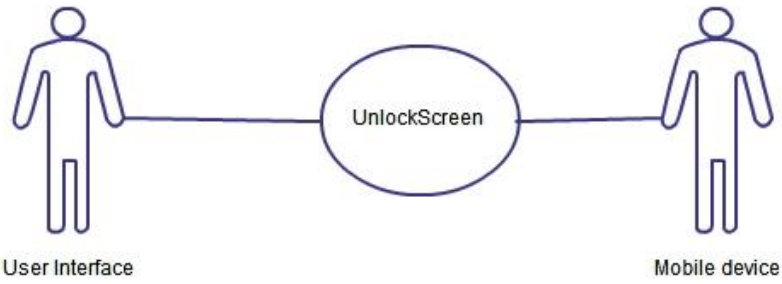

Figure-5 UnlockScreen function

.

Approve:

User can approve a desired action of the mobile device.

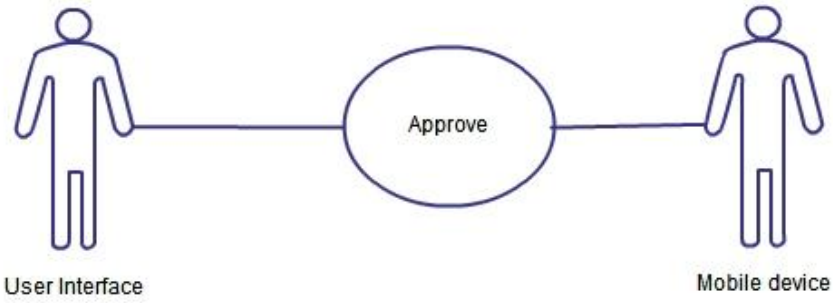

Figure-6 Approve function

Select:

User can select a desired menu of the mobile device.

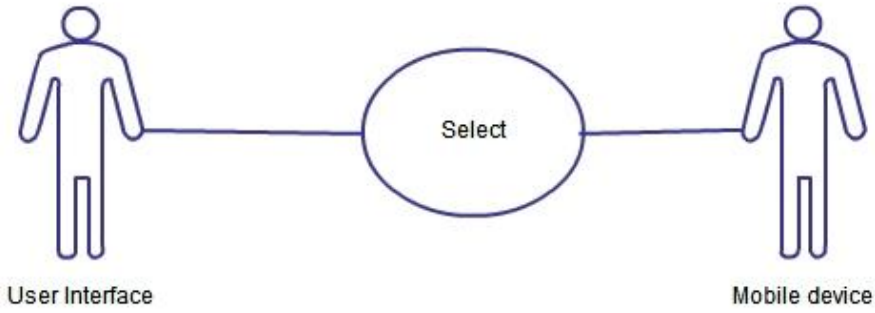

Figure-7 Select function

Quit:

User can quit from a desired menu of the mobile device.

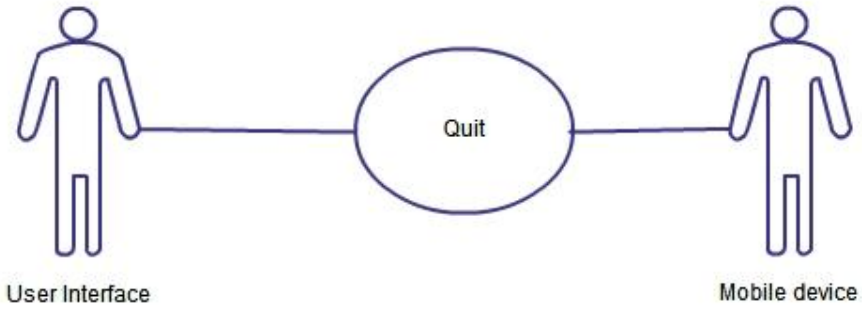

Figure-8 Quit function

Next:

User can move to next menu/image.

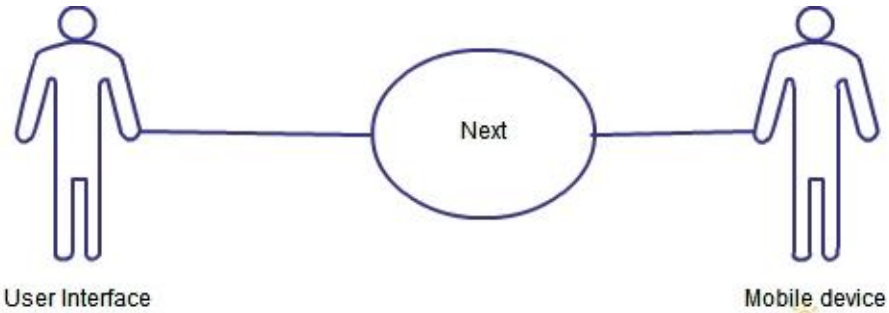

Figure-9 Next function

Previous:

User can move to previous menu/image.

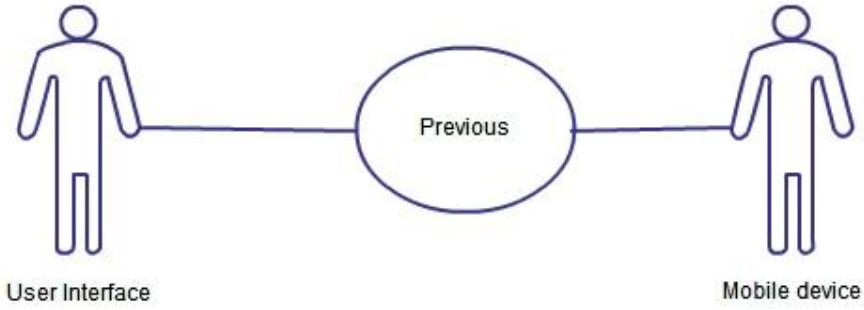

Figure-10 Previous function

Call:

User can call a desired receiver.

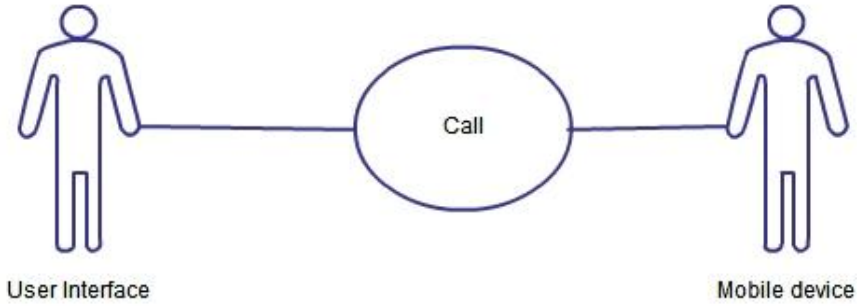

Figure-11 Call function

EndCall:

User can end the call.

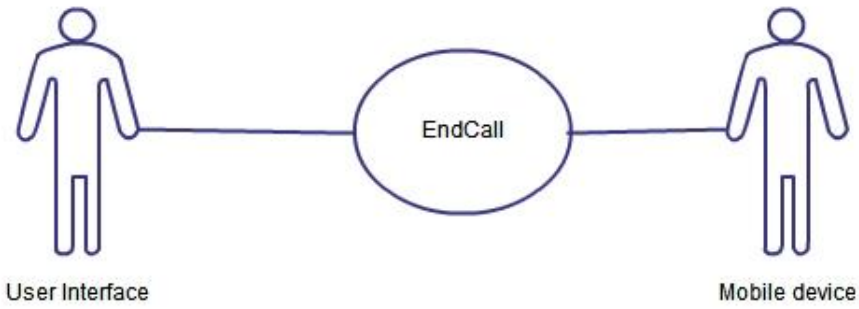

Figure-12 EndCall function

2.3. Constraints, Assumptions and Dependencies

Regulatory Policies: Damaging one of the two cameras that recognizes the hand movements of the user by using stereovision technique can cause CMBGS system to work improperly.

Hardware Limitations: In order to apply stereovision technique properly, the mobile phone used must have two cameras in front of it. The system also needs a powerful processor like Intel Atom in order to make complicated processes like processing 9-10 frames per second.

Security Consideration: Unlocking the screen will be made by recognizing the face of the user by using face recognition. Hence the mobile phone user should lock the screen unless he/she is not using the mobile phone at that moment in order to secure his/her personal information.

Criticality of the Application: In case of attempt to make hand movements with more than one hand at the same time, CMBGS might not work properly.

3. SPECIFIC REQUIREMENTS

Specific requirements are organized by use cases and functional hierarchy so that the main functions of the system will be understandable.

3.1. Interface Requirements

User interfaces consist of various menus and submenus. After unlocking the screen lock with face recognition of the user, the user will see the main menu on the screen. Main menu will have submenus named “Contacts”, “Messages”, “Media” etc. By using special hand movements, Select and Approve functions take action and user can go into another submenu. Reverse of this action is made by Quit and Approve functions in rows. In “Contacts” menu, again by using special hand movements, Call function takes action and desirable receiver can be called. Interfaces and relations of menus and submenus of a mobile phone with CMPGR system can be seen on Figure-13.

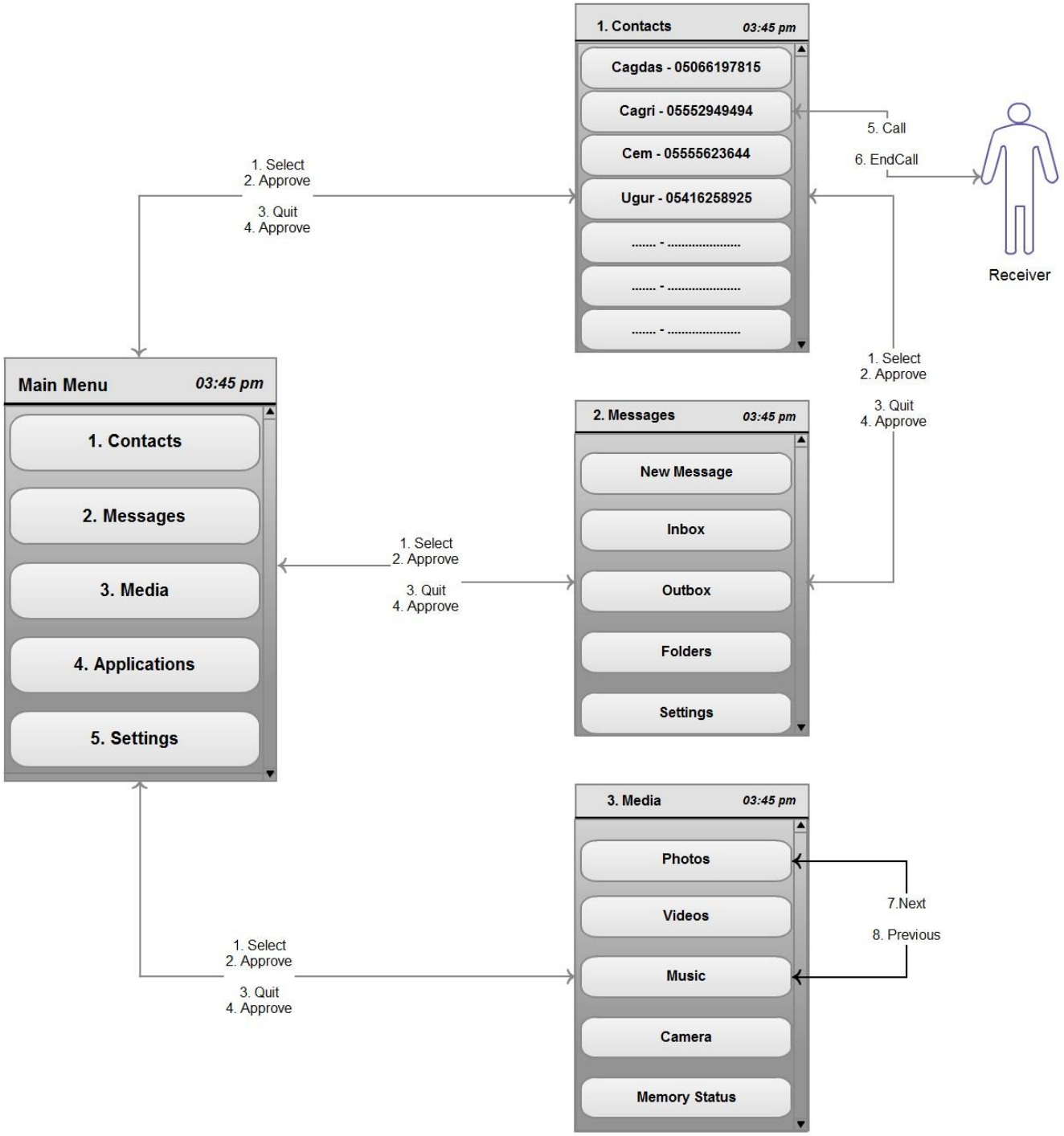

Figure-13 Relations of functions

3.2. Functional Requirements

Since CMPGR consists of two major parts named Gesture Recognition Engine and User Interface controlling, functional requirements will be examined in two chapters:

- Functional Requirements of Gesture Recognition Engine
- Functional Requirements of User Interface Controlling

3.2.1. Functional Requirements of Gesture Recognition Engine

There are three features of Gesture Recognition Engine which are described below:

3.2.1.1. Track

This feature provides the ability to track movement of the hand of the user.

3.2.1.1.1. Description and Priority

If the user moves his/her hand, tracking is the first thing to be done. Tracking of the movement is done by Gesture Recognition Engine, so this function is between user and GRE.

3.2.1.1.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) *User first unlocks the screen by authenticating of his/her face via GRE.*
2) *State or movement of the hand is recognized by cameras.*
3) *GRE tracks the recognized state or movement.*

Alternative Data Flow 1:

1) *User forgets to unlock the screen.*
2) *Gesture Recognition Engine does not work since it waits for the unlocking of the screen.*

Alternative Data Flow 2:

1) *User unlocks the screen.*
2) *The nearest object to the mobile device is not user's hand, so GRE does not work.*

Alternative Data Flow 3:

1) *User unlocks the screen.*
2) *Position of the hand cannot be detected because of the depth of the hand is out of depth borders.*
3) *GRE does not work.*

3.2.1.1.3. Functional Requirements

REQ-1: There must not be an object between the mobile device and hand of the user in order for a successful tracking.

3.2.1.2. HandLocation

This feature sends the location of the hand from GRE to User Interface.

3.2.1.2.1. Description and Priority

After the hand is tracked, this function returns the position of the hand in three dimensions.

3.2.1.2.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) *User first unlocks the screen by authenticating of his/her face via GRE.*
2) *State or movement of the hand is recognized by cameras.*
3) *GRE tracks the recognized state or movement.*
4) *GRE finds the position of the hand in three dimensions.*
5) *GRE sends the hand position to the user interface.*

Alternative Data Flow 1:

1) *User unlocks the screen.*
2) *The end position of the hand is out of depth border, GRE stops working.*

3.2.1.2.3. Functional Requirements

REQ-2: The end position of the hand is must be into the angle of sight.

3.2.1.3. HandGesture

This feature sends the gesture of the hand from GRE to User Interface.

3.2.1.3.1. Description and Priority

After the hand is tracked, this function returns the gesture of the hand.

3.2.1.3.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) *User first unlocks the screen by authenticating of his/her face via GRE.*
2) *State or movement of the hand is recognized by cameras.*
3) *GRE tracks the recognized state or movement.*
4) *GRE recognizes the gesture of the hand either in a motion or in a static state.*
5) *GRE matches the gesture with a predefined dataset.*

Alternative Data Flow 1:

1) *User unlocks the screen.*
2) *Gesture of the user is not a predefined gesture; thus, no action is done.*

3.2.1.3.3. Functional Requirements

REQ-3: Gesture of the hand must be a predefined gesture in order to use the mobile device.

3.2.2. Functional Requirements of User Interface Controlling

There are eight functions of User Interface Controlling which are described below:

3.2.2.1. UnlockScreen

This feature unlocks the screen of the mobile device.

3.2.2.1.1. Description and Priority

Unlocking screen will be done by recognizing the face of the user. Hence, this function is between user interface and mobile device.

3.2.2.1.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) *User shows his/her face to the cameras and GRE takes action.*
2) *If the face of the user is valid, screen becomes unlocked.*

Alternative Data Flow 1:

1) *User shows his/her face to the cameras and GRE takes action.*
2) *If the face of the user is not valid, screen continues to be locked.*
3) *Gesture Recognition Engine does not work since it waits for the unlocking of the screen.*

Alternative Data Flow 2:

1) *User does not show his/her face, although face recognition is activated(by a button or a scroll)*
2) *Gesture Recognition Engine does not properly work since it waits for a face to recognize.*
3) *The nearest object to the mobile device is not user's face, so GRE does not work.*

3.2.2.1.3. Functional Requirements

REQ-4: The end position of the face is must be into the angle of sight.

3.2.2.2. Select

This feature selects a desired menu.

3.2.2.2.1. Description and Priority

Selecting a desired menu will be done by recognizing of appropriate hand gesture.

3.2.2.2.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "select gesture", GRE takes action.
3) Desired menu which be identified by gesture selected.

Alternative Data Flow 1:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "select gesture", GRE takes action.
3) If the selected/desired menu does not exist, than give warning.

Alternative Data Flow 2:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than select gesture, again GRE takes action.
3) If other gesture are pre-defined for other functionality, that functionality done.

Alternative Data Flow 3:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than select gesture, again GRE takes action.
3) If other gesture is not pre-defined for other functionality, do nothing.

3.2.2.2.3. Functional Requirements

REQ-5: Select gesture must match with pre-defined gesture and location of hand must be closer object to phone.

3.2.2.3. Quit

This feature quits from a desired menu and returns back.

3.2.2.3.1. Description and Priority

Quitting from a desired menu and returning back will be done by recognizing of appropriate hand gesture.

3.2.2.3.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "quit gesture", GRE takes action.
3) Quitting from desired menu which be identified by gesture.

Alternative Data Flow 1:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "quit gesture", GRE takes action.
3) If current menu is a main menu, then do nothing and give warning.

Alternative Data Flow 2:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than quit gesture, again GRE takes action.
3) If other gesture are pre-defined for other functionality, that functionality done.

Alternative Data Flow 3:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than quit gesture, again GRE takes action.
3) If other gesture is not pre-defined for other functionality, do nothing.

3.2.2.3.3. Functional Requirements

REQ-6: Quit gesture must match with pre-defined gesture and location of hand must be closer object to phone.

3.2.2.4. Next

This feature skips left to right or up to bottom in the desired menu.

3.2.2.4.1. Description and Priority

Skipping next in desired context is not available for all menu sections and this action will be done by recognizing of appropriate hand gesture. Some examples for available menu sections are contact list, music, photos.

3.2.2.4.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "next gesture", GRE takes action.
3) Skips through left to right or up to bottom

Alternative Data Flow 1:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "next gesture", GRE takes action.
3) If current menu is not available menu for next action, then do nothing and give warning.

Alternative Data Flow 2:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than next gesture, again GRE takes action.
3) If other gesture are pre-defined for other functionality, that functionality done.

Alternative Data Flow 3:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than next gesture, again GRE takes action.
3) If other gesture is not pre-defined for other functionality, do nothing.

3.2.2.4.3. Functional Requirements

REQ-7: Next gesture must match with pre-defined gesture and location of hand must be closer object to phone.

3.2.2.5. Previous

This feature skips right to left or bottom to up in the desired menu.

3.2.2.5.1. Description and Priority

Skipping previous in desired context are not available for all menu sections and this action will be done by recognizing of appropriate hand gesture. Some examples for available menu sections are contact list, music, photos.

3.2.2.5.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "Previous gesture", GRE takes action.
3) Skips through right to left or bottom to up

Alternative Data Flow 1:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "Previous gesture", GRE takes action.
3) If current menu is not available menu for next action, then do nothing and give warning.

Alternative Data Flow 2:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than Previous gesture, again GRE takes action.
3) If other gesture are pre-defined for other functionality, that functionality done.

Alternative Data Flow 3:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than Previous gesture, again GRE takes action.
3) If other gesture is not pre-defined for other functionality, do nothing.

3.2.2.5.3. Functional Requirements

REQ-8: Previous gesture must match with pre-defined gesture and location of hand must be closer object to phone.

3.2.2.6. Call

This feature calls selected user in contact menu.

3.2.2.6.1. Description and Priority

Calling desired contact will be done by recognizing of appropriate hand gesture.

3.2.2.6.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "Call Gesture" in contact menu, GRE takes action.
3) Calling desired contact.

Alternative Data Flow 1:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user do pre-defined "Call Gesture" in contact menu, GRE takes action.
3) If current menu is not contact menu or none of contacts are chosen, then do nothing and give warning.

Alternative Data Flow 2:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than Call gesture, again GRE takes action.
3) If other gesture are pre-defined for other functionality, that functionality done.

Alternative Data Flow 3:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than Call gesture, again GRE takes action.
3) If other gesture is not pre-defined for other functionality, do nothing.

3.2.2.6.3. Functional Requirements

REQ-9: Call gesture must match with pre-defined gesture and location of hand must be closer object to phone.

3.2.2.7. EndCall

This feature ends call.

3.2.2.7.1. Description and Priority

Calling ends if user has been a ongoing call by recognizing of appropriate hand gesture.

3.2.2.7.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) User unlocked the screen according data flow of Unlock Screen.
2) Calling procedure started with appropriate gestures.
3) If user do pre-defined "End Call Gesture" in ongoing call, GRE takes action.
4) Ending call.

Alternative Data Flow 1:

1) User unlocked the screen according data flow of Unlock Screen.
2) Calling procedure started with appropriate gestures.
3) If user do pre-defined "End Call Gesture" not in ongoing call, GRE takes action.
4) Do nothing and give warning.

Alternative Data Flow 2:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than End Call gesture, again GRE takes action.
3) If other gesture are pre-defined for other functionality, that functionality done.

Alternative Data Flow 3:

1) User unlocked the screen according data flow of Unlock Screen.
2) If user does other gesture than End Call gesture, again GRE takes action.
3) If other gesture is not pre-defined for other functionality, do nothing.

3.2.2.7.3. Functional Requirements

REQ-10: EndCall gesture must match with pre-defined gesture and location of hand must be closer object to phone.

3.2.2.8. Approve

This feature approves other functionality

3.2.2.8.1. Description and Priority

It is the most important gestures, and every functionality except from next and previous functionality does approved by this functionality.

3.2.2.8.2. Stimulus and Priority

Data Flow:

Basic Data Flow:

1) User unlocked the screen according data flow of Unlock Screen.
2) Some functionality are called except next and previous.
3) If user do pre-defined "Approve Gesture", GRE takes action.
4) Called functionality approved.

Alternative Data Flow 1:

1) User unlocked the screen according data flow of Unlock Screen.
2) Next and previous section can be called.
3) If user do pre-defined "Approve Gesture", GRE takes action.
4) Do nothing and give warning.

Alternative Data Flow 2:

1) User unlocked the screen according data flow of Unlock Screen.
2) Some functionality are called except next and previous.
3) If user does other gesture than "Approve Gesture", GRE takes action.
4) If other gesture are pre-defined for other functionality, do nothing and state that waits for approving.

Alternative Data Flow 3:

1) User unlocked the screen according data flow of Unlock Screen.
2) Some functionality is called except next and previous.
3) If user does other gesture than "Approve Gesture", GRE takes action.
4) If other gesture is not pre-defined for other functionality, do nothing.

3.2.2.8.3. Functional Requirements

REQ-11: Approve gesture must match with pre-defined gesture and location of hand must be closer object to phone.

3.3. Non-functional Requirements

3.3.1. Performance Requirements

Controlling Mobile Devices via Gesture Recognition should be used by only one hand of the user, indeed. CMDGR system should run on a single core processor, by using 30% capacity of the processor. It should not exceed 256MB RAM while it is on action.

3.3.2. Design Constraints

C++ is the programming language that will be used in the software together with the OpenCV and QT libraries. Software architecture will be based on Model-View-Controller (MVC) architecture. The software designed for MeeGo and Ubuntu operating systems. MySQL will be the database server keeping the datasets and related information of mobile devices features. The software will include the reliability and portability system attributes.

4. DATA MODEL AND DESCRIPTION

In this section, the information of data models of the software is mentioned.

4.1. Data Description

Basically, 3 types of data objects will be manipulated by the software. These data objects are namely interface object, user object and engine object.

4.1.1. Data Objects

- *Interface Object*: This object contains interface information, namely information of the menu, information of the screen of mobile device to build the corresponding user interface.
- *User Object*: This object contains hand information; position and orientation of the hand. Also face information of the user will be held in user objects. Position and orientation of the hand will be updated given by data comes from camera.
- *Engine Object*: This object contains dataset of gestures.

4.1.2. Relationships

Engine object can be said as the core part of the system. User objects are created from the engine object. In order to interpret the gestures correctly, engine object uses the attributes of the user object.

Engine object also gets the state of the screen in order to start working.

Interface object also waits for the data triggering coming from the engine object.

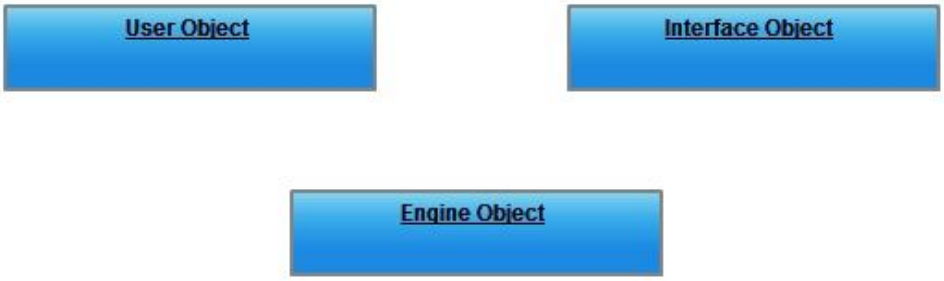

Figure-14 Data Objects

4.1.3. Complete Data Model

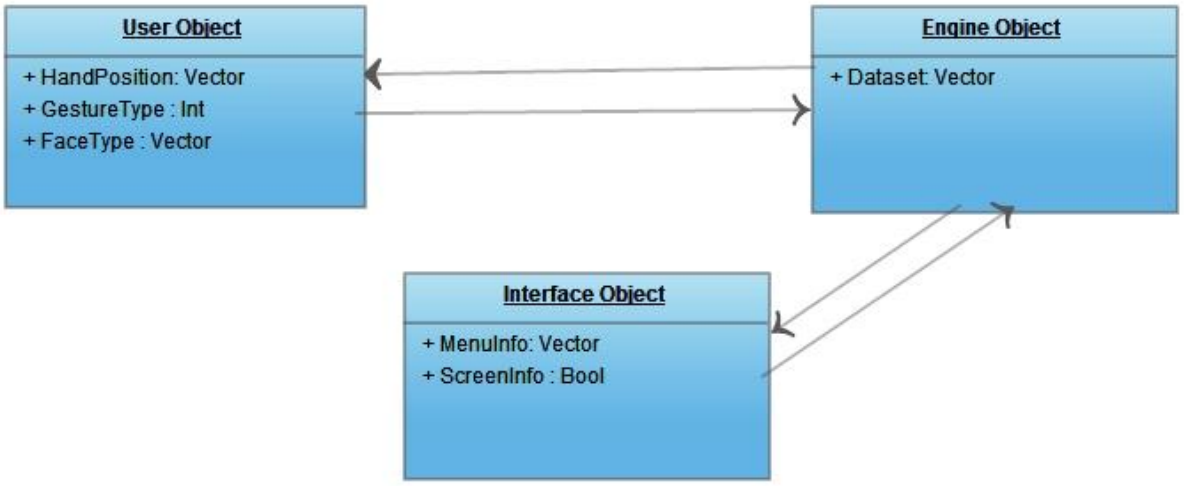

Figure-15 Complete Data Model

4.1.4. Data Dictionary

The data dictionary can be accessed via the link given below:

- http://senior.ceng.metu.edu.tr/TolleSudore/DBMS

5. BEHAVIORAL MODEL AND DESCRIPTION

5.1. Description for Software Behavior

Since CMDGR system consists of two parts, which are Gesture Recognition Engine and User Interface Controlling, description is going to be made separately.

1. Description for Software Behavior of Gesture Recognition Engine:

Engine state can be thought of as major state of this part. This state makes gesture recognition from Gesture state by using Track, HandLocation and HandGesture, respectively.

2. Description for Software Behavior of User Interface Controlling:

In order to have ability to reach states in CMDGR, successful face recognition should be done to pass over authentication state. After the authentication state, CMDGR system goes to Main Menu state, which can be considered as the main state of the system. At this step, three major states, which are Contacts, Messages and Media, will appear.

At Main Menu state, mobile phone user can go on with Contacts state and make a call by selecting the corresponding contact from the list or can go on with Messages state and can send SMS, MMS or E-Mail. After the message template is completed, next step will be Contacts. The message receiver is going to be selected from this state.

The last sub-state of Main Menu state is Media state. Here at this state, system goes to 3 states named Photos, Videos, and Music states. As their names suggest, by going on with these states, user can reach photos, videos and music, respectively.

5.2. State Transition Diagrams

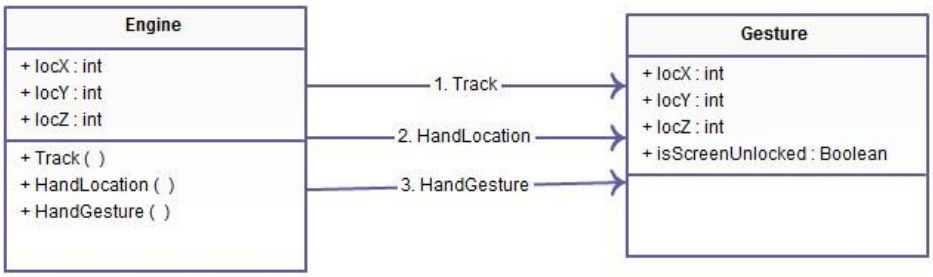

Figure-16. Transitions of functions

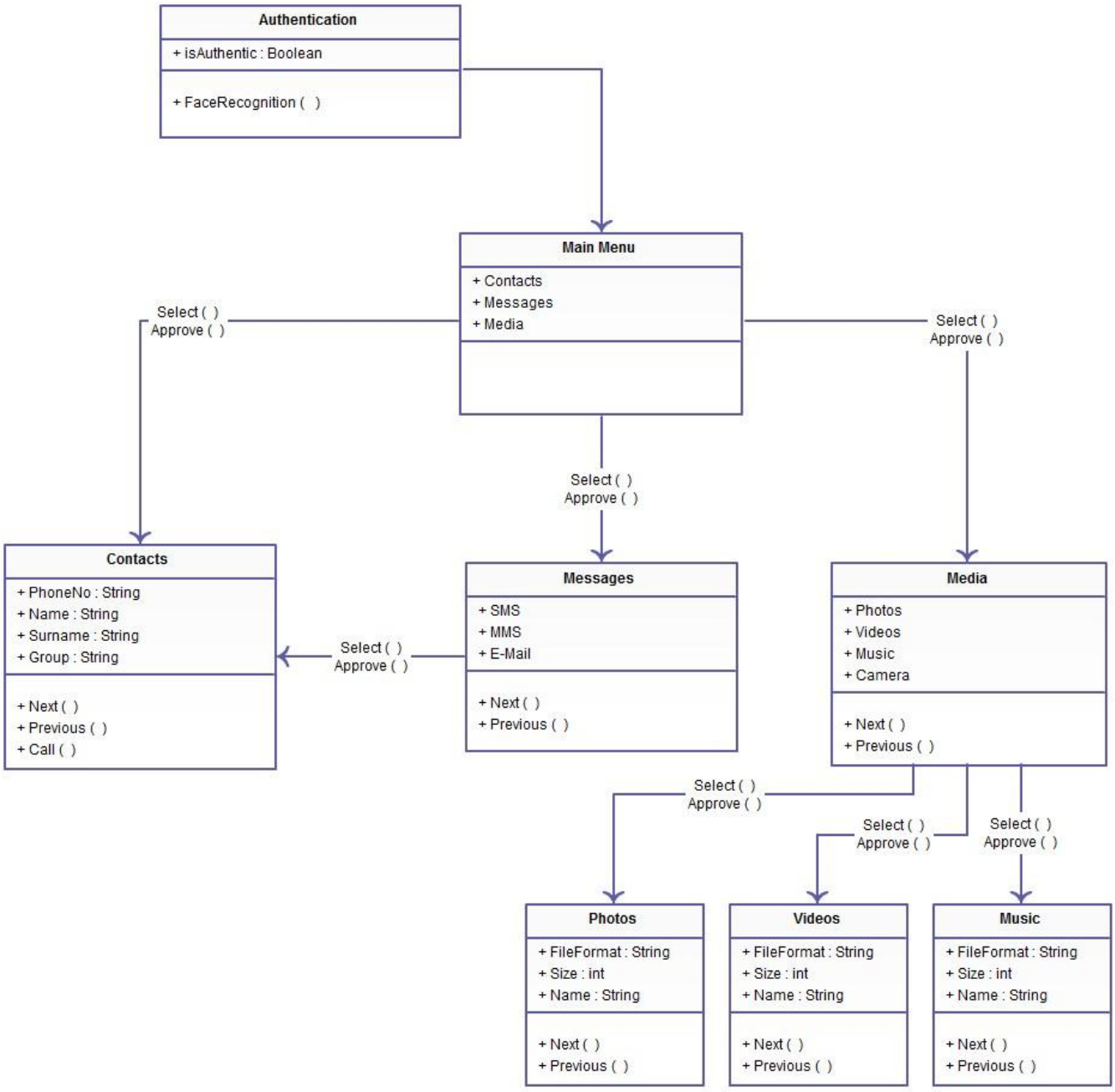

Figure-17. Complete Transition Functions

6. PLANNING

6.1. Team Structure

- Mehmet Çağdaş AKAR - Vision Researcher, Software Engineering, Engine Module Developer, Designer
- Çağrı ASLANBAŞ - Software Engineering, Interface Module Developer, Reporter
- Uğur BALTACI - Vision Researcher, Engine Module Developer, Project Leader
- Cem EKİCİ- Vision Researcher, Interface Module Developer, Public Relations, Reporter

6.2. Estimation (Basic Schedule)

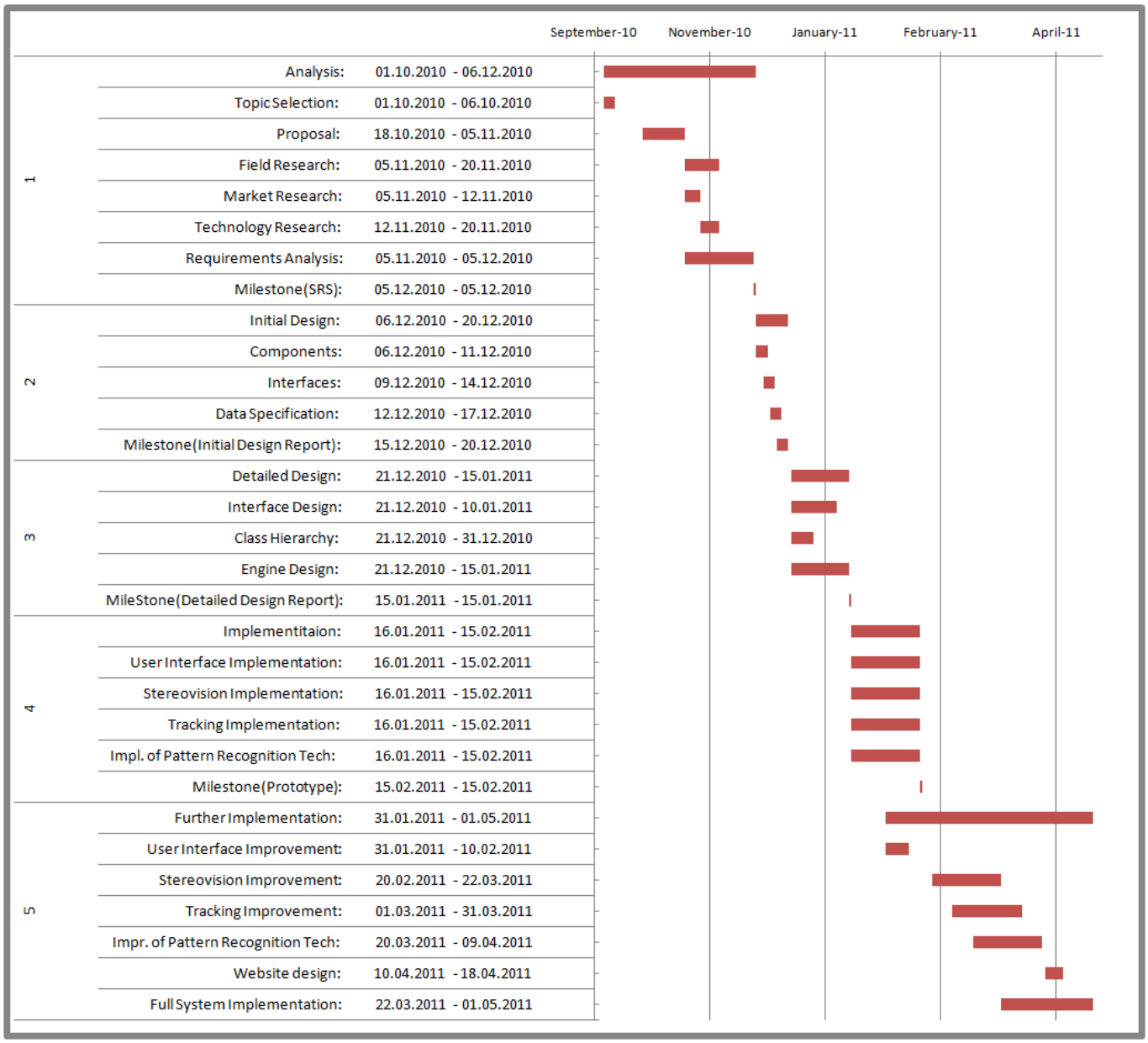

Figure-18 Schedule

6.3. Process Model

Actually, waterfall method is not the best applicable for software design process in nowadays. However, what we are asked for is more suitable for waterfall, we chose waterfall process method. Waterfall process, as you know, is a sequential process which includes conception, initiation, analysis, design, construction, testing and maintenance.

7. CONCLUSION

To conclude, we tried to reveal the requirements of Controlling Mobile Devices via Gesture Recognition in this document. Firstly, introduction and brief project description are presented. After that, overall description and product functionality are explained. Then, requirements of CMDGR are examined and demonstrated. After that data and behavioral models are introduced. Lastly, project methodology and schedule are demonstrated.

This report was very useful to focus our projects requirements and aspects. Since it's a breakthrough innovation in HCI, we expect good results for this project in terms of usability, availability, scalability, reliability and functionality.