

ANDROID MOBILITY SUITE



By

Muhammad Usman

Isma Batool

Muneeb Ahmed Bhatti

Submitted to the Faculty of Computer Science, Military College of Signals
National University of Sciences and Technology, Rawalpindi in partial fulfillment for

The requirements of a B.E Degree in Computer Software Engineering

June 2014

CERTIFICATE

Certified that the contents and form of project report entitled “**Android Mobility Suite**” submitted by Muhammad Usman, Isma Batool, and Muneeb Ahmed Bhatti have been found satisfactory for the requirement of the degree.

Supervisor: _____

Lec. Waseem Iqbal

Department of CS

Military College of Signals

ABSTRACT

ANDROID MOBILITY SUITE

Mobile phone is basic need of modern society. During the busy schedule and working on PC it is annoying to check cell phone for important messages and calls. So we provide excellent solution to this problem one does not need to check phone while working on pc, he can control his phone form PC after making connection between the two. This application provides features of mobile phone on PC when both are connected through Wi-Fi. The System consists of two parts. An android application and a Java desktop application. The aim of the system is to bring limited functionality of android phone on PC so that the user can directly control the features of his android phone from PC. The PC module will provide interface to user so that the user can perform the tasks form his computer. After successful pairing with the android phone it will provide certain controls to the user that will allow the user to perform certain tasks. It is a Java desktop application. The GUI is built using javafx along with XML and CSS. The cell phone module is the core. It will process all the requests that will be sent from the computer. It will perform all the requested functions and will send the requisite data to the computer. It is an android application. The features that are included are initiate call, send and receive SMS, application management, call logs, import contacts, remote Camera, file exploration, backup and authentication mechanism. The system uses a TCP connection for transmitting data over Wi-Fi. Sockets are used in order to achieve the communication tasks. The system is compatible to various devices and operating systems. From android 2.3 gingerbread to android 4.4 Kit Kat. It is also compatible to windows operating system. Multithreading has been implemented so the user can perform multiple functions at a time. The application is a complete suite and in useful for increasing work efficiency by providing a centralized management interface.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To all the Reformed Institutions of Pakistan, dedicated to Allah and His kingdom and,

In particular to the faculty, staff and students of Military College of Signals

ACKNOWLEDEMENTS

We would like to express our profound gratitude to our parents for their prayers and continuous moral support throughout degree program in general and our project in particular. We are very much grateful to our Project Supervisor Asst. Prof. Bilal Rauf for his support and critique which helped us reach new levels of perfection. Likewise we would like to thank co Project Supervisor Lec. Wasim Iqbal for taking up our project in absence of Asst. Prof. Bilal Rauf and helping us in every possible manner. We are highly indebted to all the faculty members who throughout the degree trained, mentored and guided us to achieve new levels of professionalism which is the hallmark of this prestigious institute.

Table of Content

| | |
|---|-----------|
| 1. Introduction | 1 |
| 1.1 Purpose | 1 |
| 1.1.1 PC Module: | 1 |
| 1.1.2 Cell phone Module: | 1 |
| 1.2 Background | 2 |
| 1.3 Problem Statement | 2 |
| 1.4 Objectives | 3 |
| 1.5 Deliverables | 3 |
| 1.6 Technological requirements | 3 |
| 1.6.1 Operating Systems | 3 |
| 1.6.2 Software Packages | 3 |
| 1.6.3 Hardware Components | 4 |
| 2 Literature Review | 4 |
| 2.1 Introduction | 4 |
| 2.1.1 Background | 4 |
| 2.2 Overall Description | 4 |
| 2.3 Similar Software and work | 5 |
| 2.3.1 AirDroid | 5 |
| 2.3.2 Samsung Kies | 7 |
| 2.4 Related Papers/Articles | 8 |
| 3 System Requirements | 14 |
| 3.1 Introduction | 14 |
| 3.2 Product Perspective | 15 |
| 3.3 Basic Assumptions | 16 |
| 3.4 Operating System | 16 |
| 3.5 Operating Conditions | 17 |
| 3.6 System Features | 17 |
| 3.7 Nonfunctional requirements | 26 |
| 3.8 Software Quality Attributes | 27 |
| 4 Design and Development | 30 |
| 4.1 Introduction | 30 |
| 4.2 Architectural Model | 31 |
| 4.2.1 Client | 31 |
| 4.2.2 Server | 31 |
| 4.3 Design Pattern | 32 |
| 4.4 Logical View | 33 |
| 4.4.2 Sequence Diagrams | 47 |
| 4.5 Implementation View | 53 |
| 4.5.1 System Class Diagram | 53 |
| 4.5.2 System Classes Description | 54 |
| 4.6 Dynamic View | 64 |
| 4.6.1 Activity Diagrams | 64 |
| 4.7 User Interface design | 72 |
| 5 System Implementation | 76 |
| 5.1 Tools and Technologies | 76 |
| 5.1.1 Eclipse SDK | 76 |
| 5.1.2 JavaFX Scene Builder | 76 |
| 5.2 Software Implementation | 77 |
| 6. Project Analysis and Evaluation | 80 |
| 6.1 Testing | 80 |
| 6.1.1 Testing Introduction | 80 |
| 6.1.2 Testing Levels | 80 |
| 6.1.3 Unit Testing | 81 |
| 6.1.4 Integration Testing | 88 |

| | | |
|----------|---|------------|
| 6.1.5 | System Testing | 101 |
| 6.1.6 | Acceptance Testing | 104 |
| 6.2 | Summary | 114 |
| 6.3 | Results and Analysis | 114 |
| 6.3.1 | Results and Analysis Introduction | 114 |
| 6.3.2 | Results | 115 |
| 6.3.3 | Analysis | 115 |
| 7 | Conclusion and Future Work..... | 115 |
| | Appendix A: Glossary | 116 |
| | Appendix B: Bibliography..... | 119 |
| | Appendix C: User Manual | 120 |

Table of Figures

| | |
|---|----|
| Figure 2-1: The Web Browser Interface of AirDroid on PC | 6 |
| Figure 2-2: The Mobile application Interface of AirDroid..... | 6 |
| Figure 2-3: Interface of SnapPea | 7 |
| Figure 2-4: Samsung Kies Web Interface | 8 |
| Figure 4-1: Client Server Distribution | 31 |
| Figure 4-2: Client Server Architecture..... | 32 |
| Figure 4-3: System Use Case Diagram | 33 |
| Figure 4-4: Application Management Use Case..... | 34 |
| Figure 4-5: Call-Log Use Case | 36 |
| Figure 4-6: Import Contacts Use Case..... | 37 |
| Figure 4-7: SMS Use Case | 39 |
| Figure 4-8: Initiate Call Use Case..... | 41 |
| Figure 4-9: Explore Files Use Case | 42 |
| Figure 4-10: Remote Camera Use Case | 44 |
| Figure 4-11: Data Backup Use Case..... | 45 |
| Figure 4-12: Authentication Sequence Diagram..... | 47 |
| Figure 4-13: Import Contacts Sequence Diagram | 47 |
| Figure 4-14: Send SMS Sequence Diagram | 48 |
| Figure 4-15: Call Log Sequence Diagram..... | 48 |
| Figure 4-16: Application Management Sequence Diagram | 49 |
| Figure 4-17: Receive SMS Sequence Diagram | 50 |
| Figure 4-18: Data Backup Sequence Diagram | 51 |
| Figure 4-19 File Exploration Sequence Diagram | 51 |
| Figure 4-20: Initiate Call Sequence Diagram | 52 |
| Figure 4-21: Remote Camera Sequence Diagram | 52 |
| Figure 4-22: System Class Diagram | 53 |
| Figure 4-23 System Activity Diagram..... | 64 |
| Figure 4-24:App Management Activity Diagram..... | 65 |
| Figure 4-25 Import Contacts Activity Diagram..... | 66 |
| Figure 4-26 Call-Log Activity Diagram..... | 66 |
| Figure 4-27 Initiate Call Activity Diagram | 67 |
| Figure 4-28 Receive SMS Activity Diagram | 67 |
| Figure 4-29 Data Backup Activity Diagram..... | 68 |
| Figure 4-30 File Exploration Activity Diagram | 69 |
| Figure 4-31 Send SMS Activity Diagram | 70 |
| Figure 4-32: Remote Camera Activity Diagram..... | 71 |
| Figure 4-33: Main Window UI Design | 72 |
| Figure 4-34: Call WindowUI Design..... | 73 |
| Figure 4-35: SMS Window UI Design..... | 73 |
| Figure 4-36: File Exploration UI Design | 74 |
| Figure 4-37: Contacts Window UI Design..... | 74 |
| Figure 4-38: Logs Window UI Design..... | 75 |
| Figure 4-39: Apps Window UI Design | 75 |

1. Introduction

1.1 Purpose

“Android Mobility Suite” will allow users to control features of any android smartphone from their computer using Wi-Fi. It will be a stand-alone application through which user can control their Android smartphone from their PC, enabling user to perform his/her tasks directly from PC instead of going to the phone. The application will have two modules.

1.1.1 PC Module:

The PC module will provide interface to user so that the user can perform the tasks form his computer. After successful pairing with the android phone it will provide certain controls to the user that will allow the user to perform certain tasks.

1.1.2 Cell phone Module:

The cell phone module will be the core. It will process all the requests that will be sent from the computer. It will perform all the requested functions and will send the requisite data to the computer.

The application when installed on computer and mobile phone provide access to the all significant features of mobile phone through computer. The main features are

- User can initiate call from his PC
- User can send and receive SMS using his PC
- Data Backup
- File Exploration and Data Management

- Call logs management
- Import contacts.
- User can use camera of mobile phone from PC
- Application management
- An effective authentication mechanism is used before establishing the connection

1.2 Background

The root cause of creating this application is that there is no product in the market which provides all features of controlling a mobile phone from PC. There are some products in the market with limited functionalities. Android Mobility Suite is integrating all significant features of those products and adding new features which make it a complete suite.

1.3 Problem Statement

Suppose you are working on your PC and you receive frequent phone calls and text messages on your phone. Now it is very frustrating sometimes to switch between the two, maximizing the effort to complete a task and time is wasted. Switching between interfaces can also be difficult sometimes. File management is not intuitive. There is no other way to sync your device with PC if the data cable is lost. There is no easy way to continuously backup your data of phone.

Keeping in mind these facts we are developing an application that maximizes effort and time required to complete a task. Provides efficient file management and provides continuous and faster way to keep backup of your data.

1.4 Objectives

Objective is to develop Windows based application which will control any android smartphone using WLAN. It will be a stand-alone application through which user can control their Android smartphone from their PC. Both devices will be connected to each other through WI-FI Signals. The application will provide control of the android device from the PC enabling user to perform his/her tasks directly from PC instead of going to the phone.

1.5 Deliverables

First Progress Report: including SRS Document

Second Progress Report: including System Design (Hardware and Software)

Third Progress Report: Demo of integrated system

Final Report: including complete documentation of the system and user manual.

1.6 Technological requirements

1.6.1 Operating Systems

- Windows
- Android

1.6.2 Software Packages

- Java Net beans 8
- Eclipse SDK
- Scene Builder

1.6.3 Hardware Components

- Personal Computer(s)
- Android Phone

2 Literature Review

2.1 Introduction

2.1.1 Background

Android is the leading operating system in smart phones. With a market share of more than 80 percent it is the leading operating system in smart phones and tablets. With the major brands like Sony, HTC and Samsung developing android based devices it is the most widely used operating system. As of September 3, 2013, 1 billion Android devices have been activated.

Android is an operating system based on the Linux kernel, and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is open source and Google releases the source code under the Apache License. This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. As of May 2013, 48 billion apps have been installed from the Google Play store.

2.2 Overall Description

The concept of this software originates when effectively using your android smart phone while working on your PC. Usability, time management, improved efficiency is the aim.

The detailed description of the software is presented in the software requirement specification document above.

2.3 Similar Software and work

2.3.1 AirDroid

The application allows user to manage the phone from his computer. Using a web browser the user can use some of the features of his phone from his/her PC.

The main features of the application are:

1. Desktop SMS.
2. Files and media management
3. Find phone using GPS.
4. Manage applications
5. Call logs and clipboard

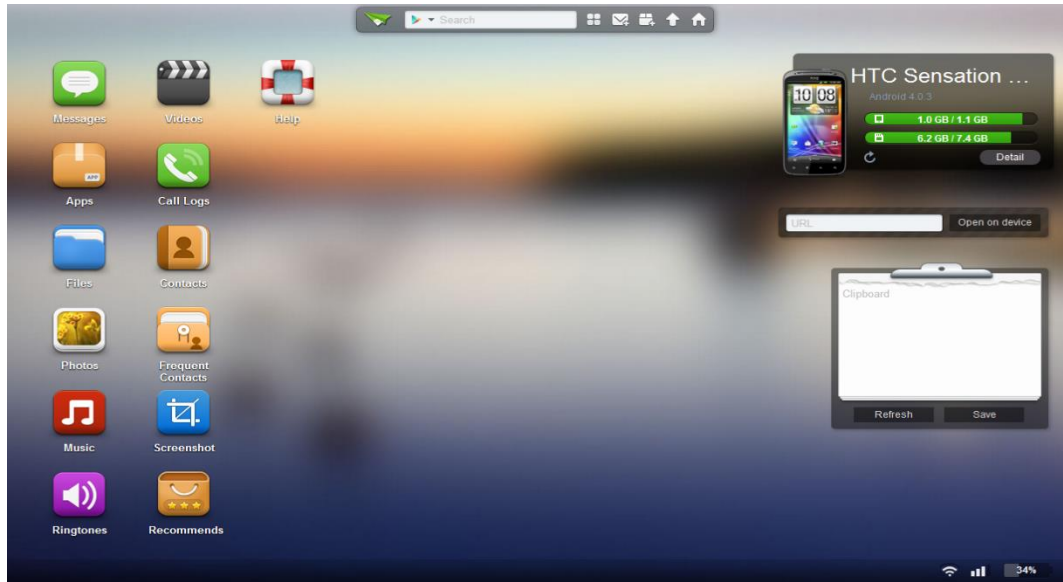


Figure 2-1: The Web Browser Interface of AirDroid on PC

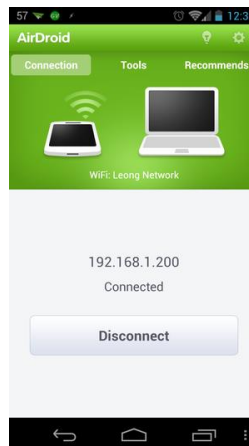


Figure 2-2: The Mobile application Interface of AirDroid

SnapPea is another product that allows you to use the functionality of your android device from your computer. The main features are:

1. Text messages
2. Instant access to pictures
3. Apps
4. Contacts

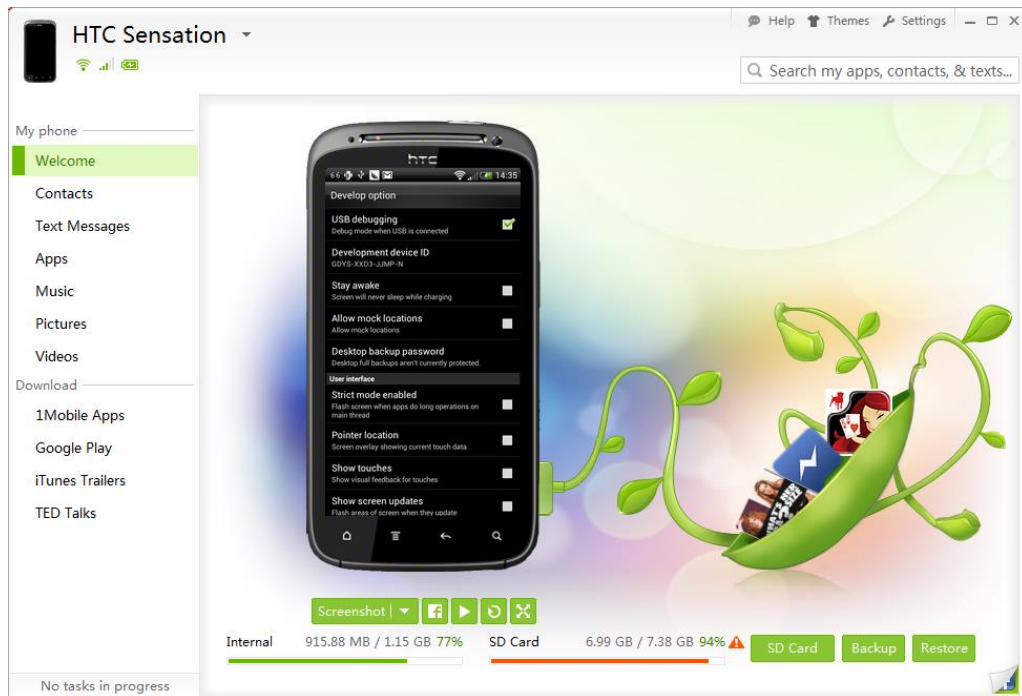


Figure 2-3: Interface of SnapPea

2.3.2 Samsung Kies

Samsung Kies is an application that allows you to manage media from your PC. The main features are:

1. Manage music
2. Manage photos
3. Manage videos

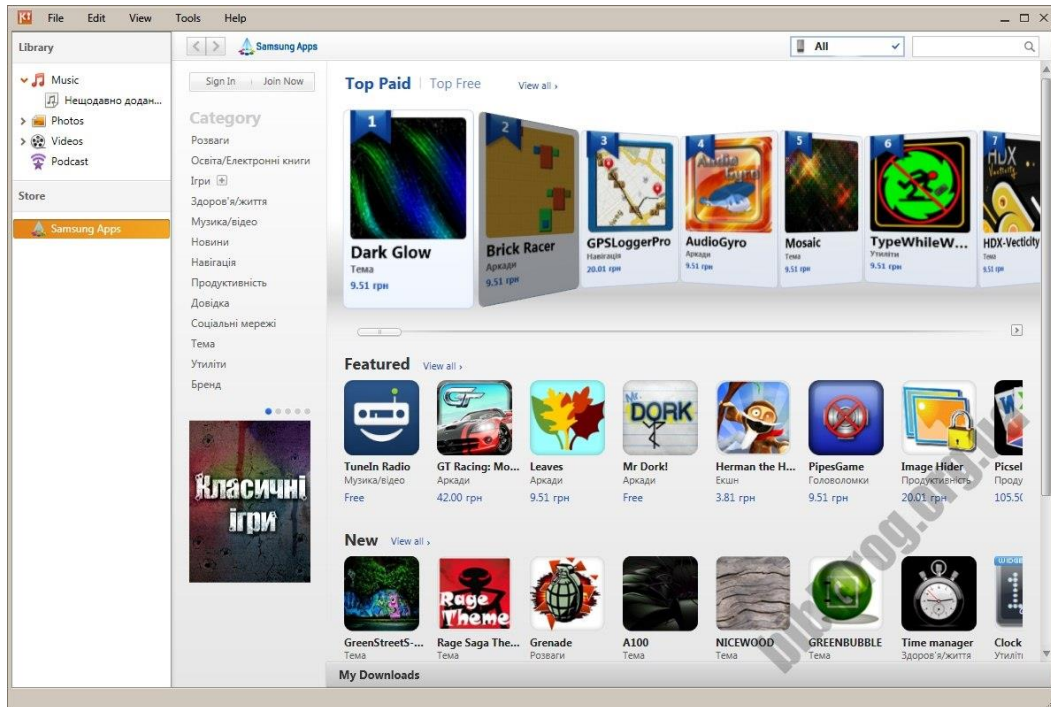


Figure 2-4: Samsung Kies Web Interface

2.4 Related Papers/Articles

2.4.1 The Android Operating System [1]

Butler, G., Finley, D., & Kasavana

The paper gives in an overall view of Android operating system. It covers the following topics:

- Android history
- Differences of Android from other mobile operating system
- Linux kernel
- Libraries and Dalvik virtual machine
- Android based devices and their specifications
- Android applications framework

It also provides detail on android's market share along with advantages.

2.4.2 Managing the Android Activity Lifecycle [2]

Article on Android Developer Website explaining different states and android application can be and how to handle events pertaining to application state.

2.4.3 What's the big IDE? Comparing Eclipse and NetBeans [3]

Webpage on Comparison of IDE options for our Android Application Development Project.

2.4.4 Development Techniques for Android Platform Mobile Device Application [4]

Ivan Njunjic

This thesis focuses on Android application development techniques needed to implement a mobile application. This thesis provides a unique approach to Android development due to its single focus on the data. The topics of interest in this paper were:

- Android application environment
- Android application structure
- Android application file structure
- Development techniques
- Android API's access and understanding.

2.4.5 Android Application Fundamentals [5]

Developers.android

This article is a complete guide to understand android application structure and fundamentals. It explains all the application components by detail along with examples.

The topics covered are:

- Android application lifecycle
- Activities
- Services
- Content providers
- Broadcast receivers
- Android manifest file
 - User permissions
 - API levels
 - API libraries

It gives a complete guide about these topics explaining with different examples and code segments.

2.4.6 Android application files [5]

Developers.Android

This article deeply discusses the files and directories that android application is composed of. The author explains all the files in a very good way which is easy to learn and implement. The android application consists of the following files

2.4.6.1 Activity.java

This file contains all the java code that runs the application. It contains all the business logic and the related java code. It can be of any name.

2.4.6.2 Activity.xml

It contains the xml code of the application. The GUI of the application is built using xml. The views, lists, content providers all the made using xml. It can be of any name.

2.4.6.3 Androidmanifest.xml

This file is one of the most important files of the app. It contains the related permissions the app is going to use. The permissions to access specific hardware and software e.g. camera, GPS etc.

This article also gives an overview of the folders and directories of content e.g. pictures that is to be used in the android application.

2.4.7 Sockets programming in Java [6]

Qusay H. Mahmoud

This is a complete guide to sockets and socket programming in java. As require network communication so it is necessary that we understand how to communicate between two devices programmatically. This article provides deep understanding to sockets, ports and how to program them and communicate effectively. The topics covered are:

- TCP and UDP introduction
- Programming sockets
- Using ports
- Java libraries that are required for socket programming
- Client server introduction

2.4.8 Android working with camera [7]

Ravi Tamada

A tutorial that covers how to access and work with android camera in a number of ways. It also describes how to handle and store images taken. The tutorial clearly explains:

- Accessing phone camera
- Permissions required
- Taking and storing pictures

2.4.9 Views in Android [8]

Views are responsible for measuring, layout and drawing themselves and their child elements. Everything that draws on the UI of the Android application utilizes the view class and is an instance of the view class. Views are also responsible for saving their UI state and handling touch events. You can also create a new view from existing View Objects.

The following topics related views are covered here:

- Responsibility of views
- Reasons for creating views
- Defining views

2.4.10 Android Drag and Drop Tutorial [9]

As of Android 4.0 drag and drop of views is supported. You register a listener on the views which can be dragged and you define other views as possible drop targets. Following is covered here

- Registering a listener with the instance of View.
- Passing data from one View to other View.
- Dragging operations in a View.

2.4.11 Android Animations Overview [10]

Android 3.0 introduced the **Properties Animation API** which allow to change object properties over a predefined time interval. The API allows to define for arbitrary object properties a start and end value and apply a time-based change to this attribute. This API can be applied on any Java object not only on Views.

Classes in Properties Animation API

The superclass of the animation API is the Animator class. Typically the ObjectAnimator class is used to modify the attributes of an object.

You can also add an AnimatorListener class to your Animator class. You can use this listener to perform actions before or after a certain animation by calling the Listener object. Using animate() method on a View object is used to perform simultaneous animations. You can set different attributes of animation like the duration.

2.4.12 Android Logging Tutorial [11]

The Android system uses a centralize logging for all logs. The application programmer can also write log messages. The tooling to develop Android applications allows you to define filters for the log statements you are interested in. In correctly configured Eclipse IDE you can enable LogCat View to see different event logged by connected Android Device.

2.4.13 Android ListView [12]

The display of elements in a lists is a very common pattern in mobile applications. The user sees a list of items and can scroll through them. Individual list item can be selected, you can perform different operation by selecting the List Item. The List View can be binded to List using Adapters. So that data in a list can be shown on the list view.

2.4.14 Threading in Java [13]

An article that gives you a complete overview of threading in java. Multithreading is an important part of large projects so it is necessary that we have a deep understanding of this topic. We have to implement threading in order to increase performance. The article covers the following topics in detail:

- Introduction to threading
- Lifecycle of thread
- Implementing runnable
- Thread synchronization
- Thread control
- Inter thread communication

3 System Requirements

3.1 Introduction

This chapter describes the requirements specifications for Android Mobility Suite.

3.2 Product Perspective

The existing applications which allow to control mobile phone from computer contain different features which are integrated in Android Mobility Suite and some new and necessary features are added which were not in any of existing applications. The ancestors of Android Mobility Suite are:

| | |
|--------------------------|---|
| Air Droid | Available on Google Play Store with a rating of 4.5/5 with approximately 5 – 10 million installations per month. |
| Android Commander | Android File, application and device manager, very popular with developer community. |
| Samsung Kies | Samsung's desktop software to transfer music, contacts, and photos, update software, and sync wirelessly from your phone or tablet. |

In order to cope up with the increasing efficiency demands the mobility suite is intended to prove itself as an optimum tool for effective data and task management and also improving not only the productivity but also the improving the effectiveness and efficiency. Comparison between our project and some of the similar products available in market is given below.

| | AMS (our system) | AirDroid | Android Commander | Remote Phone Call |
|------------------|------------------|----------|-------------------|-------------------|
| Call | Yes | No | No | Yes |
| SMS | Yes | Yes | No | Yes |
| File Exploration | Yes | Yes | Yes | No |
| Backup | Yes | No | Yes | No |
| Remote Camera | Yes | Yes | No | No |
| Contacts | Yes | Yes | No | No |
| Logs | Yes | No | No | No |
| App Management | Yes | No | No | No |

Table 3-1: Comparison of AMS with Similar Applications

It is a complete system whose one module is installed on computer and other on mobile. The system model consists of a computer and a cell phone, it provides GUI for both applications. A java based desktop application is run on the pc whereas an android application runs on the android device. The communication between the two devices is through Wi-Fi.

3.3 Basic Assumptions

The project is based on following assumptions:

- Android based mobile will be available.
- Windows pc will be available.
- Wi-Fi connection will be present for the connection between mobile and computer.
- User will be familiar with the windows and android.

3.4 Operating System

To run the software user be will required to have windows operating system on pc and smart phone with android operating system.

3.5 Operating Conditions

Wi-Fi should be available to run the application successfully.

3.6 System Features

3.6.1 Initiating a phone call from computer

Description and Priority

This feature will allow user to initiate a call form mobile phone by giving instructions from computer using the user interface. The priority of this feature is 9 (note 9-highest).

Stimulus/Response Sequences

Normal Flow:

| | |
|-----------------|--|
| Stimulus | User clicks the icon of call on computer. |
| Response | System gives the option to choose from contact list or dial pad. |
| Stimulus | User clicks on a contact or enters a number from dial pad. |
| Response | System displays the contact information in case a contact is selected and number in case the number dialed is not a contact. |
| Stimulus | User clicks DIAL button. |
| Response | System initiates the call and displays the status of the call as dialing. |

Alternate Flow:

- 1- If the user wants to go back and presses the back button in selecting contact or dialing number window the system will go the main window of module.
- 2- If the cancel key is pressed after the call is initiated the system will cancel the ongoing call.
- 3- If the close button is pressed the system should close the module.

Functional Requirements

Requirement 1: The system should allow the user to initiate the call.

Requirement 2: The system should show a list of contacts to user when requested.

Requirement 3: The system should show the dial pad to the user when requested.

3.6.2 Sending SMS from computer

Description and Priority

This feature allows user to send SMS directly from the computer using his mobile services.

The priority of this feature is 9.

Stimulus/Response Sequences

Normal Flow:

Stimulus **User clicks the icon of SMS on computer.**

Response System opens the interface of the SMS module.

Stimulus **User clicks on text box for writing the message.**

Response System allows user to write a message in that text box.

Stimulus **User clicks on the phone number field.**

Response System allows user to either enter the new number or to select number from contact list.

Stimulus **User clicks on SEND button.**

Response System gives the status of sending SMS and sends message to that particular number.

Alternate Flow:

- 1- If the close button is pressed the system should close the SMS module.
- 2- If the back button is pressed while selecting a contact or dialing the recipient's number the system will bring the user back to the main window of module.

Functional Requirements

Requirement 1: The user should be able to send text message.

Requirement 2: The user should have some area to write the text message.

Requirement 3: The user should be able to choose recipient from contact list or enter a number.

3.6.3 Import Contacts

Description and Priority

This feature allows user to import contacts from phone to computer's hard disk. The priority of this feature is 8.

Stimulus/Response Sequences

Normal Flow:

Stimulus **User clicks on contacts button**

Response System shows a list of all the contacts of the phone.

Stimulus **User clicks on import contacts button**

Response System gives a dialog box asking the location where the user wants to save the contacts on computer's hard disk.

Stimulus **User gives the location and presses the save button**

Response System saves the contacts to the given location and prompts the user that contacts have been saved.

Alternate Flow:

- 1- If the close button is pressed the system should close the module.
- 2- If the back button is pressed while selecting location the system should bring the user to the main window of module.

Functional Requirement

Requirement 1: System should show all the contacts present on the phone.

Requirement 2: System should allow the user to import contacts.

Requirement 3: User should be able to choose the location where he/she wants to save contacts.

3.6.4 Data backup

Description and Priority

This feature allows the user to create a backup of the data on the phone on his computer.

The priority of this feature is 7.

Stimulus/Response Sequences

Normal Flow:

Stimulus User clicks on create backup button

Response System gives a dialog box asking the location where the user wants to create the backup on computer's hard disk.

Stimulus **User gives the location and presses the create button**

Response System creates the backup at the given location and prompts the user that backup has been created.

Alternate Flow:

- 1- If the close button is pressed the system should close the module.
- 2- If the back button is pressed while selecting location the system should bring the user to the main window of module.

Functional Requirements

Requirement 1: The user should be able to create backup of his phone's data.

Requirement 2: The system should allow user to choose the location where to create backup.

3.6.5 Remote Camera

Description and Priority

This feature allows the user to get a real time picture from the phone's camera on the PC.

The priority of this feature is 7.

Stimulus/Response Sequences

Normal Flow:

Stimulus **User clicks on remote camera button**

Response System will show window having image view and button of take picture

Stimulus User clicks on take picture button

Response System will take an image from the phone's camera and show it on the PC

Alternate Flow:

1- If the close button is pressed the system should close the module.

Functional Requirements

Requirement 1: The user should be able to view the image from the phone's camera.

Requirement 2: The system should allow switching of front/rear Camera (if equipped).

3.6.6 Call logs management

Description and Priority

This feature will allow the user to effectively manage the call logs. It will show all the call logs which will include missed, dialed and received calls. The priority of this feature is 6.

Stimulus/Response Sequences

Normal Flow:

| | |
|-----------------|---|
| Stimulus | User clicks on call logs button |
| Response | System displays the call logs which include missed, received and dialed calls. |
| Stimulus | User clicks the missed calls/ received call/ dialed calls button |
| Response | System displays the missed/ dialed/ received call separately on different pages/tabs. |

Alternate Flow:

- 1- If the close button is pressed the system should close the module.
- 2- If the back button is pressed in a selected tab then the system will go to the main window of the module.

Functional Requirements

Requirement 1: The user should be able to view the collective call logs.

Requirement 2: The user should be able to view missed, received and dialed calls separately.

3.6.7 Application management

Description and Priority

This feature allows the user effectively manage the applications on the phone. The user will be able to view all the installed and running applications on phone. The user will also be able to uninstall an application. The priority of this feature is 6.

Stimulus/Response Sequences

Normal Flow:

| | |
|-----------------|--|
| Stimulus | User clicks on application management button |
| Response | System displays a list of all the installed applications on phone |
| Stimulus | User clicks the running tab |
| Response | System displays all the running applications along with the amount of resources they are using. |
| Stimulus | User clicks on an installed application |
| Response | System gives the option of uninstalling the application. |
| Stimulus | User clicks uninstall |
| Response | System uninstalls the application on the phone and prompts the user that the application has been uninstalled. |

Alternate Flow:

- 1- If the close button is pressed the system should close the module.
- 2- If the back button is pressed in a selected tab then the system will go to the main window of the module.

Functional Requirements

Requirement 1: The system should provide a list of all the applications installed on the phone.

Requirement 2: The system should display all the running applications along with the amount of resources being used.

Requirement 3: The system should allow the user to uninstall an application.

3.6.8 File exploration and data management

Description and Priority

This feature allows the user to explore the files and folders present on the phone memory and also copy, cut, paste and delete files. The priority of this feature is 5.

Stimulus/Response Sequences

Normal Flow:

Stimulus **User clicks on file exploration button**

Response System opens a dialog containing options phone memory and external memory card.

Stimulus **User clicks the phone/memory card button**

Response System displays all the files and folders present in the phone memory/external memory card.

Stimulus **User clicks on folder**

Response System opens the folder showing sub folders and files present in that folder

Alternate Flow:

- 1- If the close button is pressed the system should close the module.
- 2- If the back button is pressed system should take the user on step back i.e. to the previous window.

- 3- If the user presses the home button the system should take the user to the main window.

Functional Requirements

Requirement 1: The system should show all files and folders present in the phone memory or external memory card when the option is selected.

Requirement 2: The user be able to go into folders and sub-folders.

3.7 Nonfunctional requirements

3.7.1 Performance Requirements

3.7.1.1 System Response

The system should response to the user's action within approximately 5-10 seconds or more depending on the hardware and connection speed between the two modules. Response here implies that the task started should be initiated in this duration otherwise an error should be indicated.

3.7.1.2 Time of Data Backup

Though for data backup and export of contacts the time cannot be defined here depending upon different file sizes or size of contacts. Also this heavily depends on the underlying hardware and connection speed.

3.7.1.3 Safety Requirements

During file/contacts transfer or backup, the changing of data will be prevented in order to prevent data inconsistency therefore the user must wait for the task to complete.

3.7.2 Security Requirements

Authorization Passcode is generated by application on the smart phone. The user has to enter that Code into application in the PC to initiate & Authorize Connection.

3.8 Software Quality Attributes

Some of the quality attributes identified includes:

- **Adaptability**

As already mentioned the system should be compatible to devices running Android version 2.3.3 (gingerbread) to Android version 4.2 (Jelly bean). Updating the Android OS version will not have any impact on the working of the system.

- **Availability**

The application will be available all the time, provided mobile and PC are in working state, also the connection does not break between the two.

- **Backup**

The software will be creating the backup of the phone if it is being requested by the user. So in case data is lost on the phone it can be recovered

- **Capacity**

The system shall provide storage up to storage capacity of PC. In case of limited space it should prompt the user.

- **Compatibility**

The application is compatible with windows and android. It is also compatible with their older versions. Minimum version of android should be gingerbread and minimum windows version should be windows Vista.

- **Competitive edge**

Android control mobility suite is integrating all significant features of those products and adding new features which make it a complete suite. There are some products in the market with limited functionalities the .to view the comparison refer to figure (a).

- **Compliance**

Not applicable

- **Documentation**

A video tutorial and a user manual would be available along with software.

- **Efficiency**

The system takes approximately 5-10 seconds to respond to the user's request depending on the underlying hardware and connection speed.

- **Flexibility**

The layout/architecture of the application will be flexible enough so if addition or change can be done of needed.

- **Interoperability**

Semantic interoperability is provided by the system that allows exchange of information between mobile and pc.

- **Maintainability**

Whenever there is a change in requirement or bug found, the application will be easily maintainable.

- **Portability**

In API, portability can be defined as “compatibility of application with platform (Android’s version) upgraded or downgraded versions. In Android’s platform when an up gradation is done, application requires some changes for compatibility with new version. Extensive testing will be required for verification & validation to address compatibility issues. Supported base android version will be Android 2.3.3 with support up to Android 4.2.

- **Reliability**

System will have extensive error handling mechanism. Any task if not executed properly or changing state within approx.10-15 seconds will generate an error.

- **Robustness**

If the connection between the user and the system is broken prior to a request being either confirmed or canceled the system shall prompt the user that the connection has been interrupted.

- **Scalability**

One user at a time is allowed.

- **Security**

System will be secure as there will be an effective authentication mechanism along with encrypted transmission over the communication channel.

- **Stress**

The system shall be capable of handling extremely high or stressful load.

- **Start-up time**

Not applicable because there is no specific requirement.

- **Timeliness**

Not applicable

- **Usability**

The user interface of the application will be designed user friendly and self-explanatory. Although if required minimum training of 5-10 minutes will be enough for the understanding of the system.

4 Design and Development

4.1 Introduction

This chapter describes Software Design Document for the Android Mobility Suite (AMS).

4.2 Architectural Model

This section provides a detailed and comprehensive architectural overview of the system.

Client server pattern is followed in which PC acts as a client and sends connections request on TCP server on mobile.



Figure 4-1: Client Server Distribution

4.2.1 Client

Client is PC. It runs applications of Mobile phone by sending TCP request and fetching the requested information from Android phone.

4.2.2 Server

Server is Android phone, listens the request form the client continually and receives the request packets form the client and processes them and sends the data to client. Following are the sequence of events that happen.

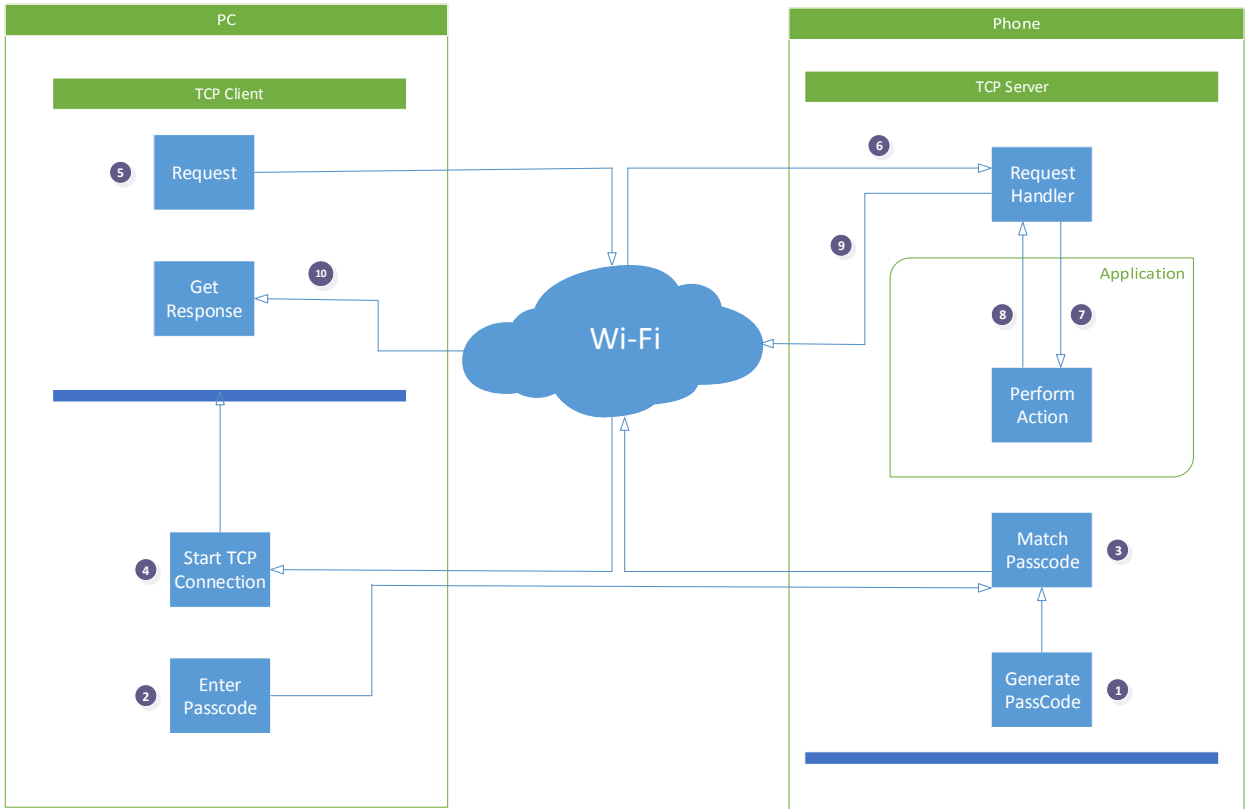


Figure 4-2: Client Server Architecture

4.3 Design Pattern

The design pattern used here is **Facade Pattern**. It provides a unified interface to a set of interfaces in a subsystem and defines a higher-level interface that makes the subsystem easier to use. A facade exposes simplified functions that are mostly called and the implementation conceals the complexity that clients would otherwise have to deal with. In general the implementation uses multiple packages, classes and function there in. Well written facades make direct access of other classes rare.

In our system the facade class is the main window. It provides a fairly simple interface and delegates the client request to the particular subsystem or class. All the other classes

constitute the systems that provide different functionalities. The main window class provides an interface to access and utilize all the functionality of the system.

4.4 Logical View

4.4.1 Use Case Diagrams

4.4.1.1 System Use Case Diagram

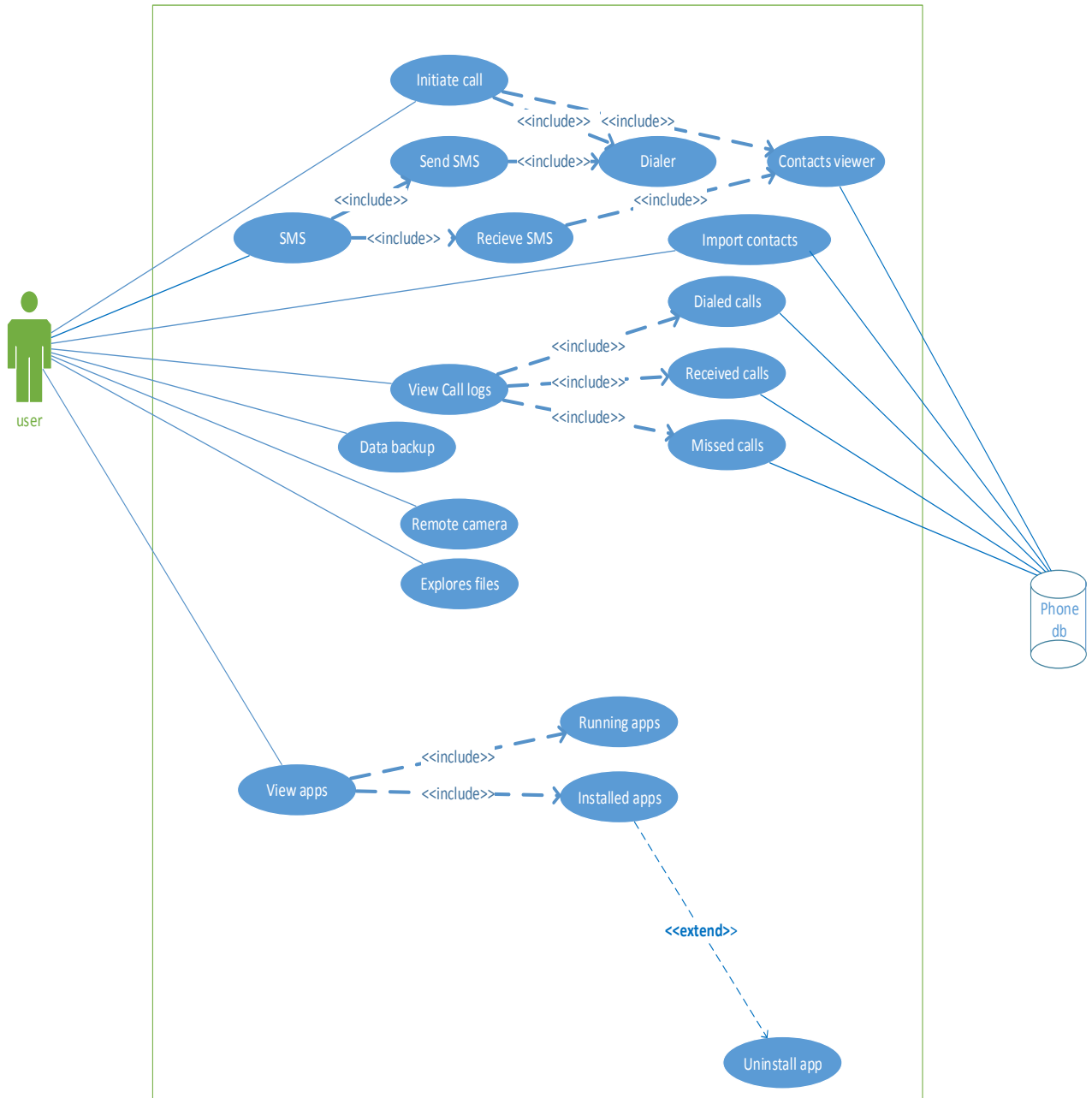


Figure 4-3: System Use Case Diagram

4.4.1.2 Application Management Use Case Diagram

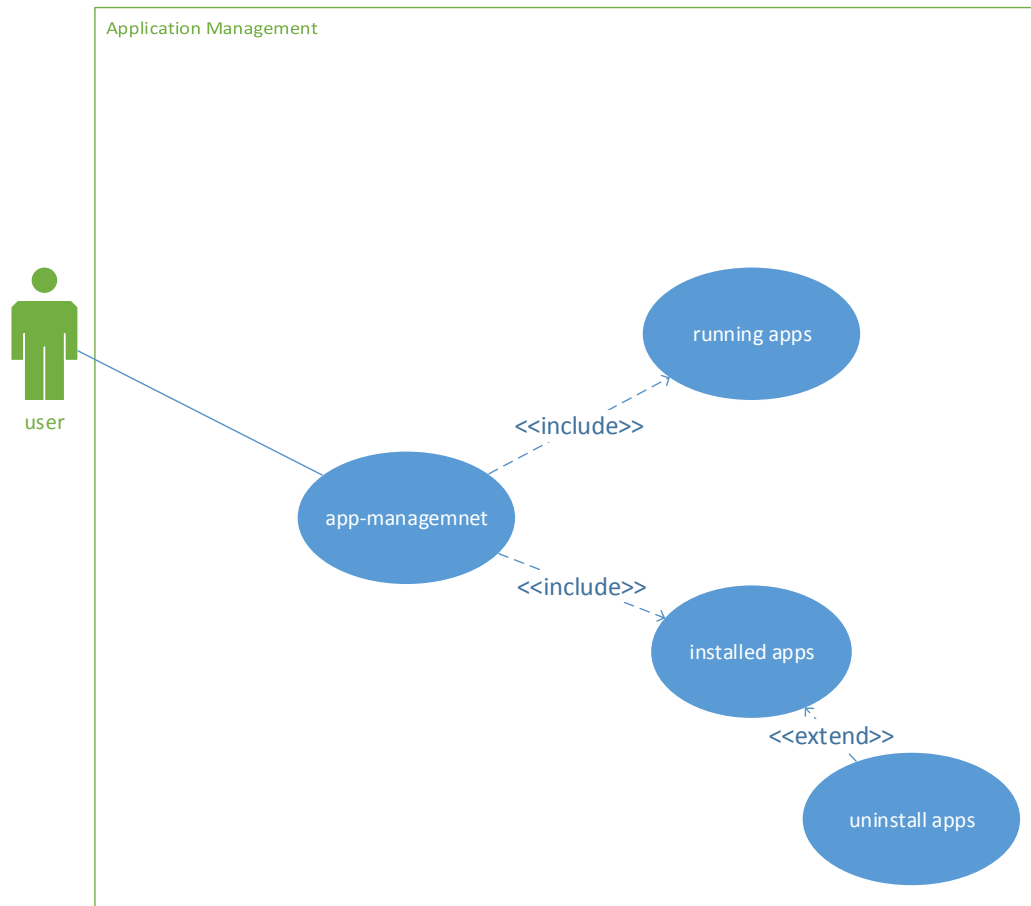


Figure 4-4: Application Management Use Case

4.4.1.2.1 Use Case Description

This use case describes how the user views running and installed applications of his android phone on his PC and how uninstalls the installed applications form PC.

4.4.1.2.1.1 Actors

User.

4.4.1.2.1.2 Pre-Conditions

1. User must have AMS installed on his PC/Laptop/ and Mobile.

2. User must have Wi-Fi hotspot available.

4.4.1.2.1.3 Basic Flow of Events

1. User starts the AMS application.
2. User connects the PC with android phone.
3. User clicks on the icon of application management
4. Application management has two buttons one for viewing running applications and other for installed applications.
5. User clicks on installed applications button
6. System gives list of all installed applications of phone
7. User clicks on an app a new screen opens showing a button to uninstall the app
8. User clicks the button and system prompts that application will be completely uninstalled.
9. User clicks ok
10. Application is uninstalled
11. User clicks on running applications button
12. Views all the running applications

4.4.1.2.1.4 Alternate Flow

1. If the close button is pressed the system should close the apps module.

4.4.1.2.1.5 Post-Conditions

1. Successful Completion

User is able to view updated list of running and installed applications. The selected application should be uninstalled.

4.4.1.3 Call-log Use Case Diagram

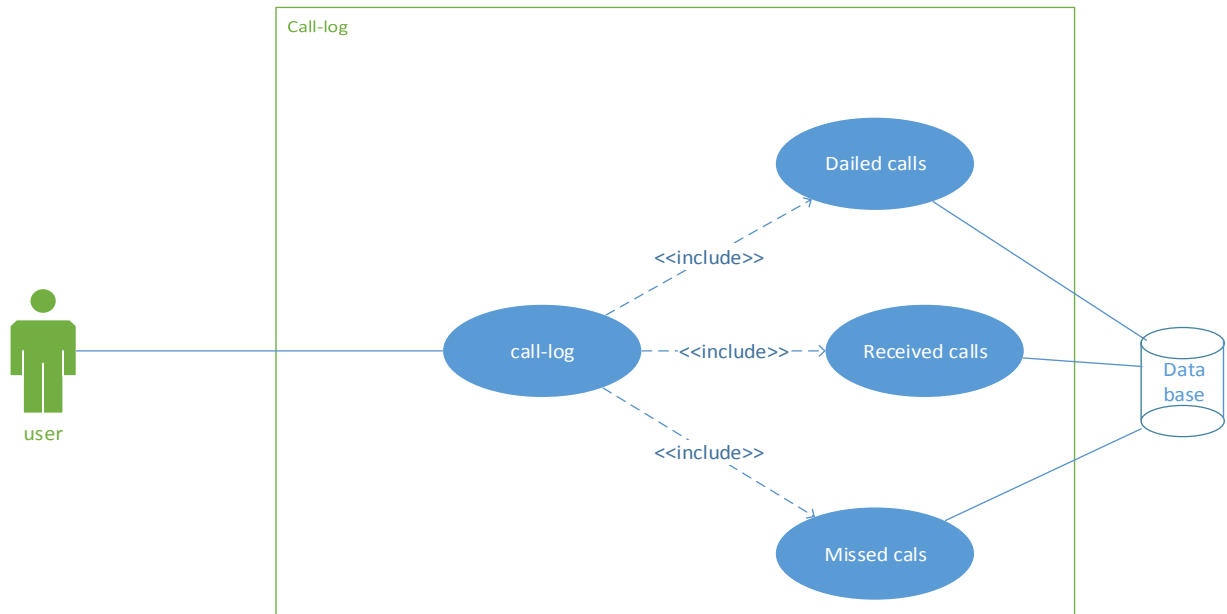


Figure 4-5: Call-Log Use Case

4.4.1.3.1 Use Case Description

This use case describes how the user is able to see dialed, received and missed calls of android phone on PC with duration and date.

4.4.1.3.1.1 Actors

1. User.
2. Database

4.4.1.3.1.2 Pre-Conditions

1. User must have application installed on his PC/Laptop and Mobile.
2. User must have Wi-Fi hotspot available.

4.4.1.3.1.3 Basic Flow of Events

1. User starts the AMS application.

2. User connects the PC with android phone.
3. User clicks on the icon of call-log.
4. Application takes all the call record of phone from its database.
5. A list of dialed received and missed calls opens on the PC.

4.4.1.3.1.4 Alternate Flow

1. If the close button is pressed the system should close the logs module.

4.4.1.3.1.5 Post-Conditions

1. Successful Completion

User is able to view the call logs.

4.4.1.4 Import contacts Use Case Diagram

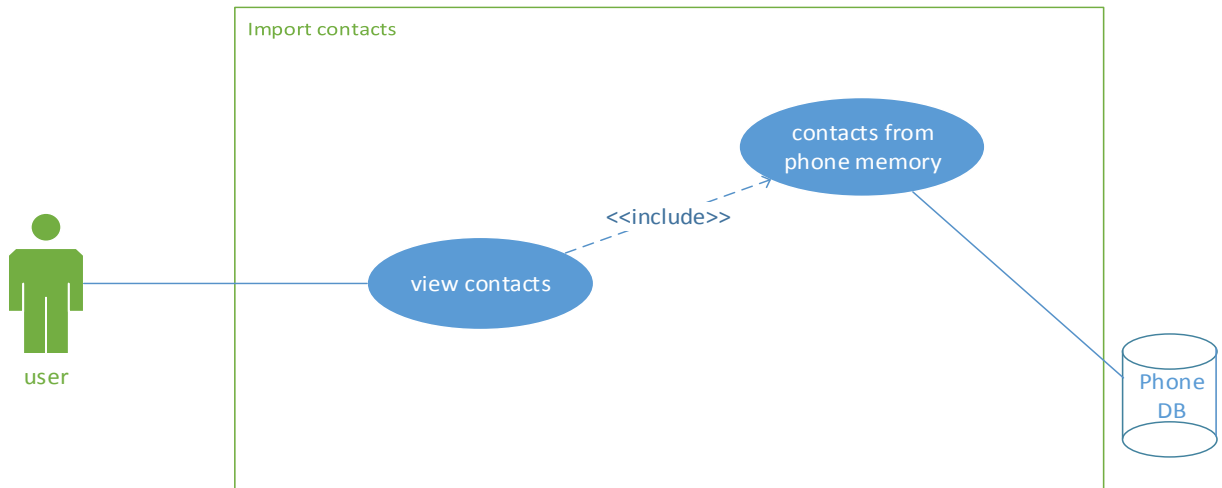


Figure 4-6: Import Contacts Use Case

4.4.1.4.1 Use Case Description

This use case describes how the AMS gives contacts details of mobile phone on PC.

4.4.1.4.1.1 Actors

1. User
2. Database

4.4.1.4.1.2 Pre-Conditions

1. User must have application installed on his PC/Laptop android phone.
2. User must have Wi-Fi hotspot available.

4.4.1.4.1.3 Basic Flow of Events

1. User starts the AMS application.
2. User connects the PC with android phone.
3. User clicks on the icon of contacts.
4. Application takes all the contacts record of phone from its database.
5. All contacts are displayed.

4.4.1.4.1.4 Alternate Flow

1. If the close button is pressed the system should close the contacts module.

4.4.1.4.1.5 Post-Conditions

1. Successful Completion

User is able to get the list of contacts with names and numbers.

4.4.1.5 SMS Use Case Diagram

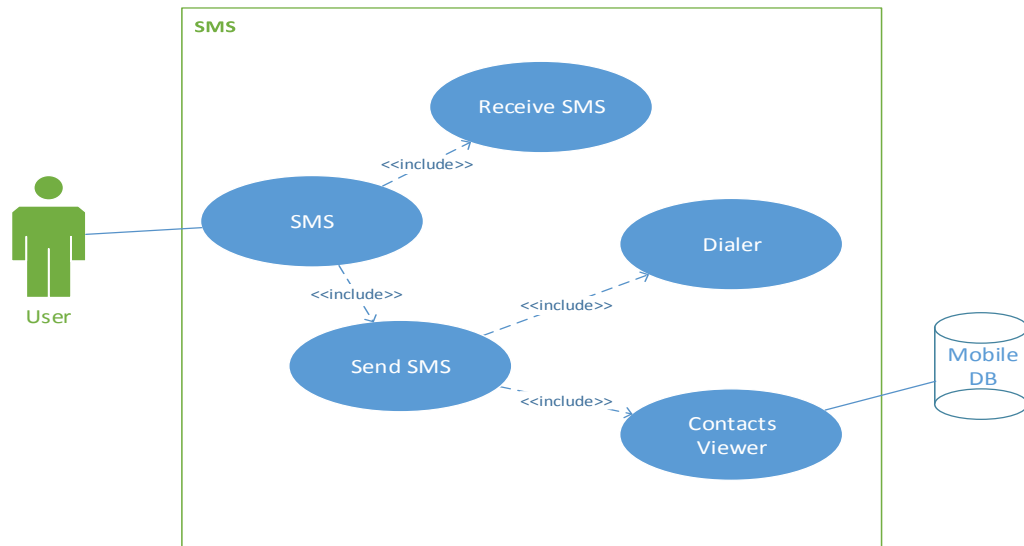


Figure 4-7: SMS Use Case

4.4.1.5.1 Use Case Description

This use case describes how the AMS allows the user to send and receive SMS form PC.

4.4.1.5.1.1 Actors

1. User
2. Database

4.4.1.5.1.2 Pre-Conditions

1. User must have application installed on his PC/Laptop android phone.
2. User must have Wi-Fi.
3. User should have credit in mobile.
4. GSM signals should be available.

4.4.1.5.1.3 Basic Flow of Events

1. User starts the AMS application.
2. User connects the PC with android phone.

3. System receives the SMS and updates the variables and phone's database.
4. User views the message along with the sender's name.
5. User clicks send SMS button.
6. User types the message.
7. Inserts mobile number of receiver either through dialer or form contacts viewer.
8. User sends the message.

4.4.1.5.1.4 Alternate Flow

1. If the close button is pressed the system should close the SMS module.

4.4.1.5.1.5 Post-Conditions

Successful Completion

User is able to receive the message successfully

Unsuccessful Completion

User is unable to send the SMS.

4.4.1.6 Initiate Call Use Case Diagram

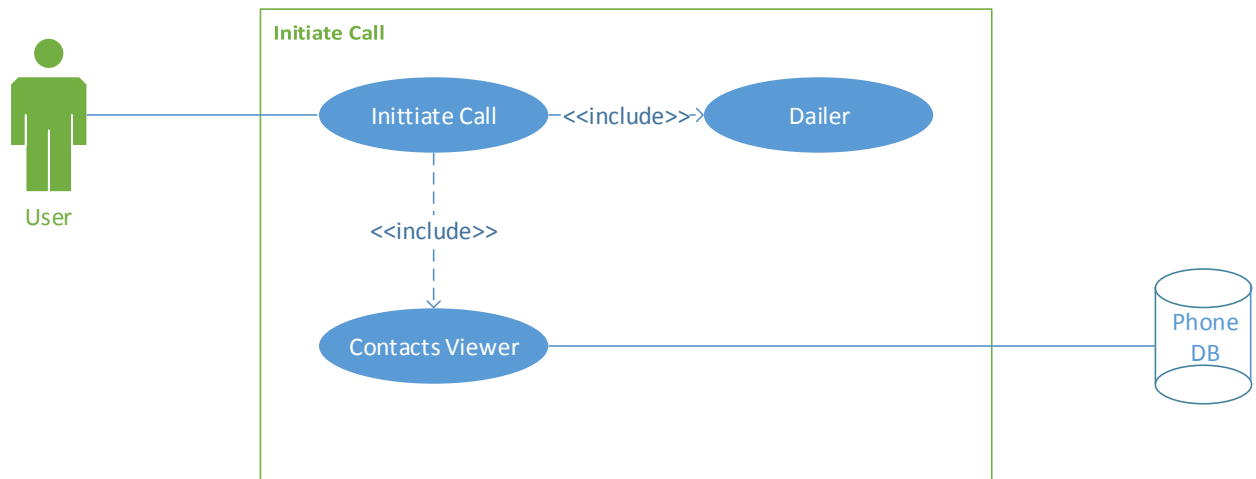


Figure 4-8: Initiate Call Use Case

4.4.1.6.1 Use Case Description

This use case describes how the AMS allows the user to initiate Call form pc.

4.4.1.6.1.1 Actors

1. User
2. Database

4.4.1.6.1.2 Pre-Conditions

1. User must have application installed on his PC/Laptop android phone.
2. User must have Wi-Fi hotspot available.
3. GSM signals should be available.
4. User should have credit in mobile.

4.4.1.6.1.3 Basic Flow of events

1. User starts the AMS application.
2. User connects the PC with android phone.
3. User clicks initiate call.

4. System retrieves the data that is required form phone database.
5. User either dials a number from dialer or selects a number from contacts viewer.
6. User makes the call.

4.4.1.6.1.4 Alternate Flow

1. If the close button is pressed the system should close the call module.
2. If the cancel button is pressed the ongoing call should be stopped.

4.4.1.6.1.5 Post-Conditions

1. Successful Completion

User is able to make the call successfully

2. Unsuccessful Completion

User is unable to make the call.

4.4.1.7 Explore Files Use Case Diagram

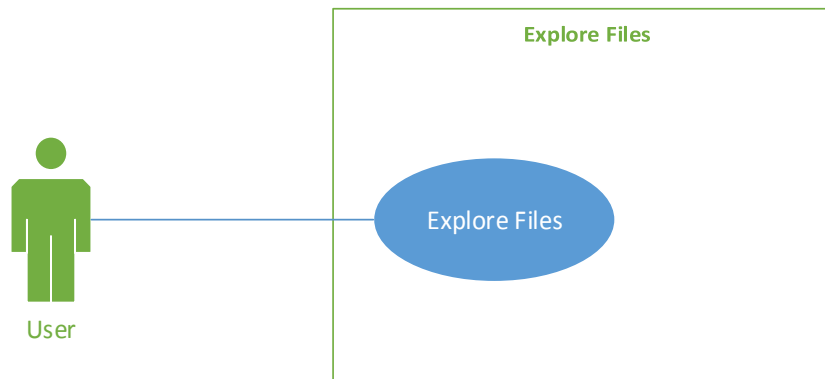


Figure 4-9: Explore Files Use Case

4.4.1.7.1 Use Case Description

This use case describes how the AMS allows the user to view, cut, copy, paste and delete files of android phone from PC.

4.4.1.7.1.1 Actors

User

4.4.1.7.1.2 Pre-Conditions

1. User must have application installed on his PC/Laptop android phone.
2. User must have Wi-Fi hotspot available.

4.4.1.7.1.3 Basic Flow of Events

1. User starts the AMS application.
2. User connects the PC with android phone.
3. User clicks on explore files button.
4. User click phone or external memory form the dialogue
5. Upon selecting the corresponding folders and files are displayed.
6. User can delete, cut, paste and copy the folders.

4.4.1.7.1.4 Alternate Flow

1. If the close button is pressed the system should close the files module.

4.4.1.7.1.5 Post-Conditions

1. Successful Completion

User is able to view copy paste delete and paste files successfully.

4.4.1.8 Remote Camera Use Case Diagram

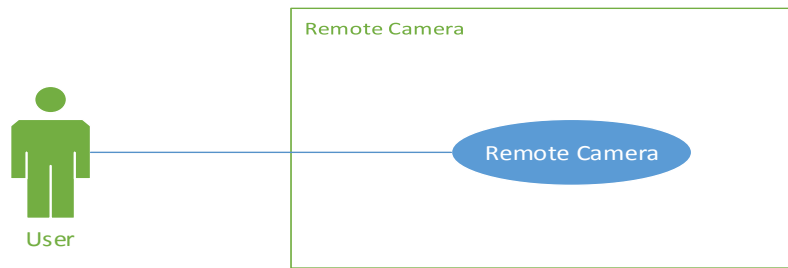


Figure 4-10: Remote Camera Use Case

4.4.1.8.1 Use Case Description

This use case describes how the AMS allows the user to take the picture from a phone using PC.

4.4.1.8.1.1 Actors

1. User

4.4.1.8.1.2 Pre-Conditions

1. User must have application installed on his PC/Laptop android phone.
2. User must have Wi-Fi hotspot available.

4.4.1.8.1.3 Basic Flow of Events

1. User starts the AMS application.
2. User connects the PC with android phone.
3. User clicks capture image button.
4. User views the image that is captured

4.4.1.8.1.4 Alternate Flow

1. If the close button is pressed the system should close the camera module.

4.4.1.8.1.5 Post-Conditions

1. Successful Completion

User is able to view the image captured.

4.4.1.9 Data Backup Use Case Diagram

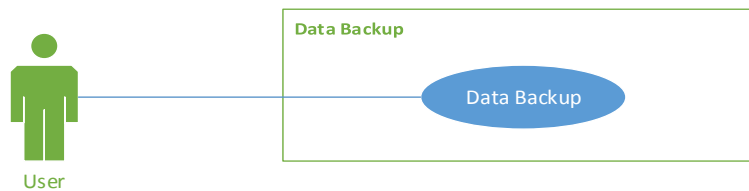


Figure 4-11: Data Backup Use Case

4.4.1.9.1 Use Case Description

This use case describes how the AMS allows the user to create backup of mobile phone using PC.

4.4.1.9.1.1 Actors

User

4.4.1.9.1.2 Pre-Conditions

1. User must have application installed on his PC/Laptop android phone.
2. User must have Wi-Fi hotspot available.

4.4.1.9.1.3 Basic Flow of events

1. User starts the AMS application.
2. User connects the PC with android phone.
3. User clicks data backup button.
4. System gives a dialog box asking the location where the user wants to create the backup on computer's hard disk
5. User selects the location and backup is created at that location.

4.4.1.9.1.4 Alternate Flow

- 1- If the close button is pressed the system should close the data backup module.

4.4.1.9.1.5 Post Conditions

1. Successful Completion

User is able to create the backup

4.4.2 Sequence Diagrams

4.4.2.1 Authentication Sequence Diagram

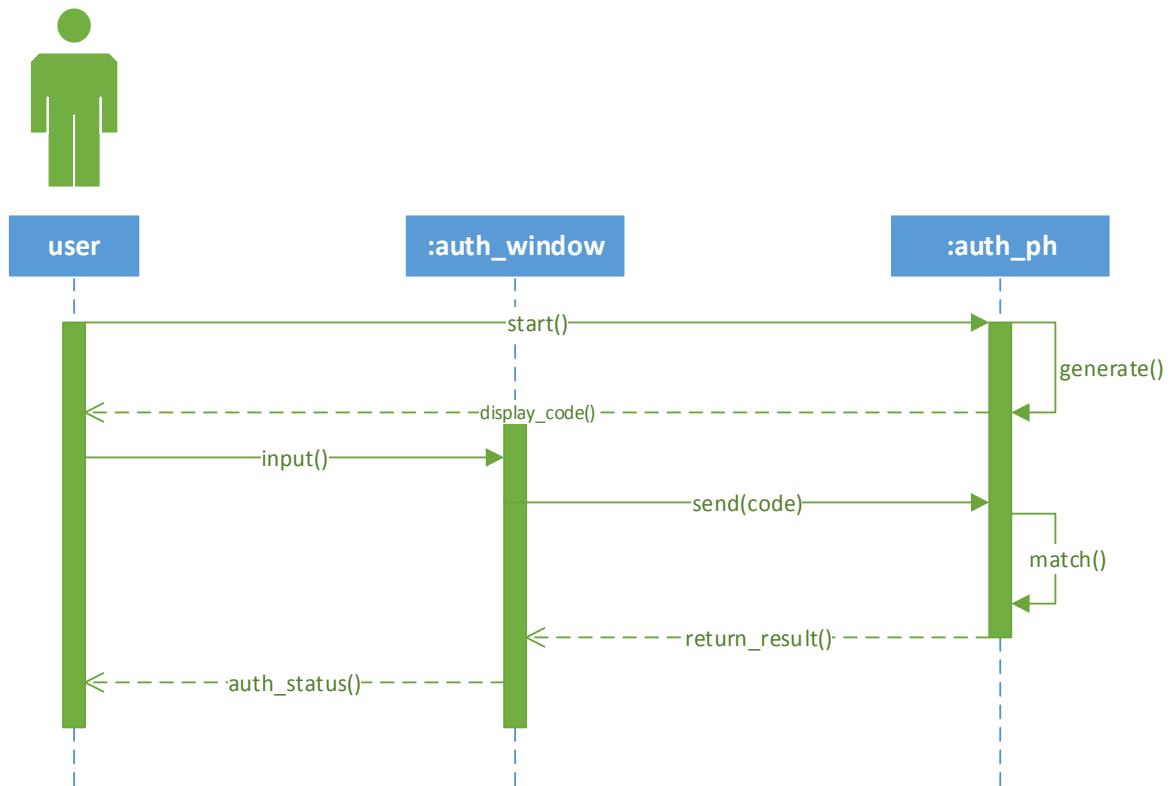


Figure 4-12: Authentication Sequence Diagram

4.4.2.2 Import Contacts Sequence Diagram

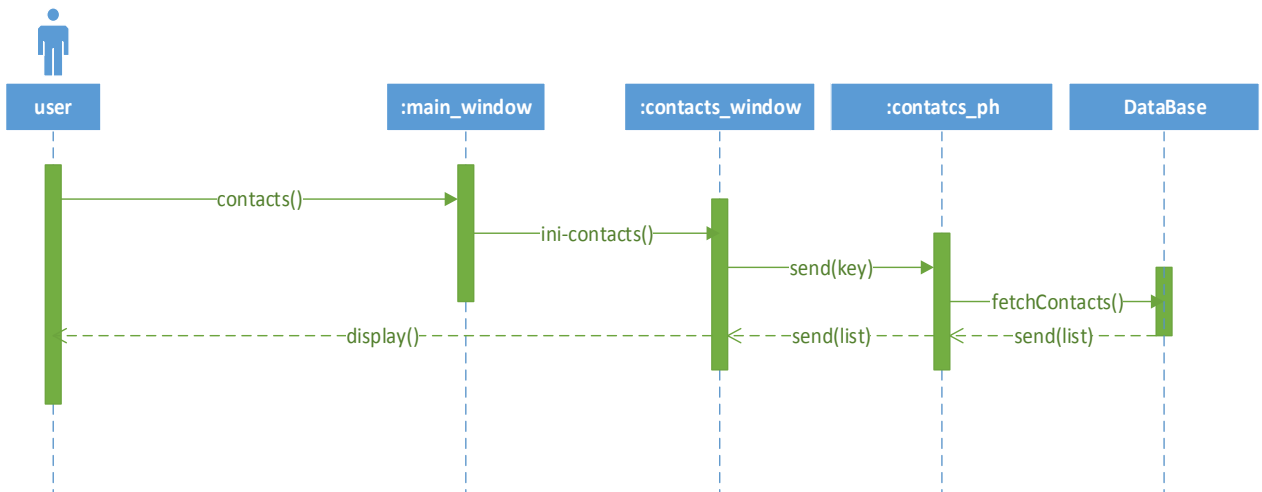


Figure 4-13: Import Contacts Sequence Diagram

4.4.2.3 Send SMS

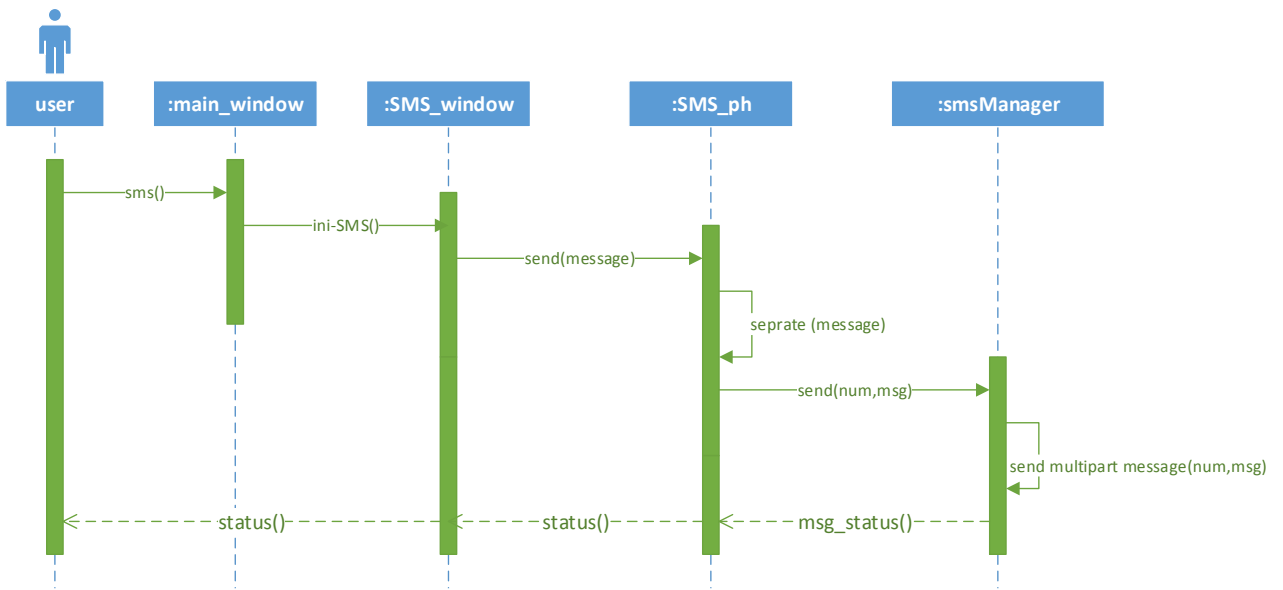


Figure 4-14: Send SMS Sequence Diagram

4.4.2.1 Call-log Sequence Diagram

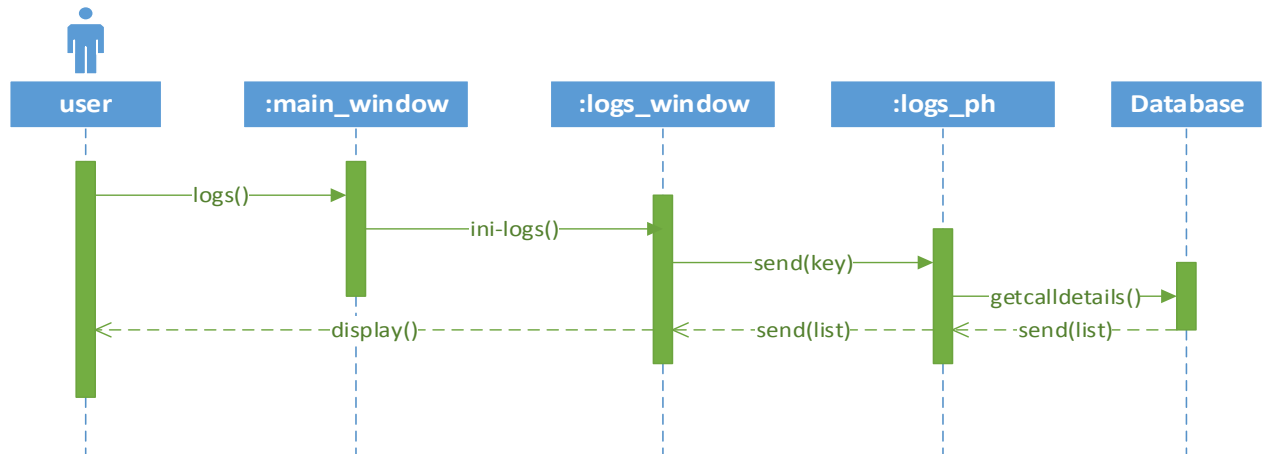


Figure 4-15: Call Log Sequence Diagram

4.4.2.2 Application management Sequence Diagram

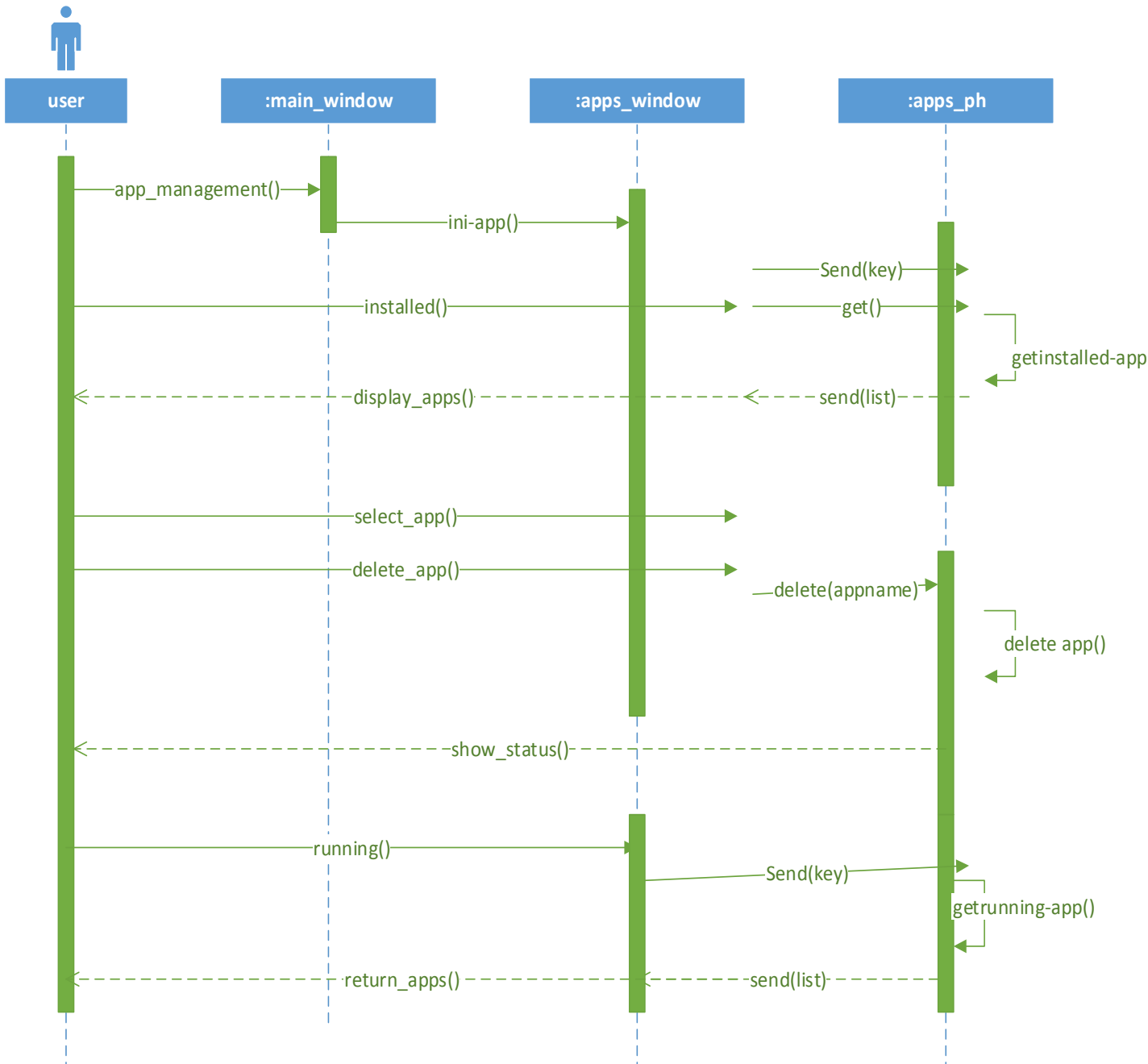


Figure 4-16: Application Management Sequence Diagram

4.4.2.3 Receive SMS

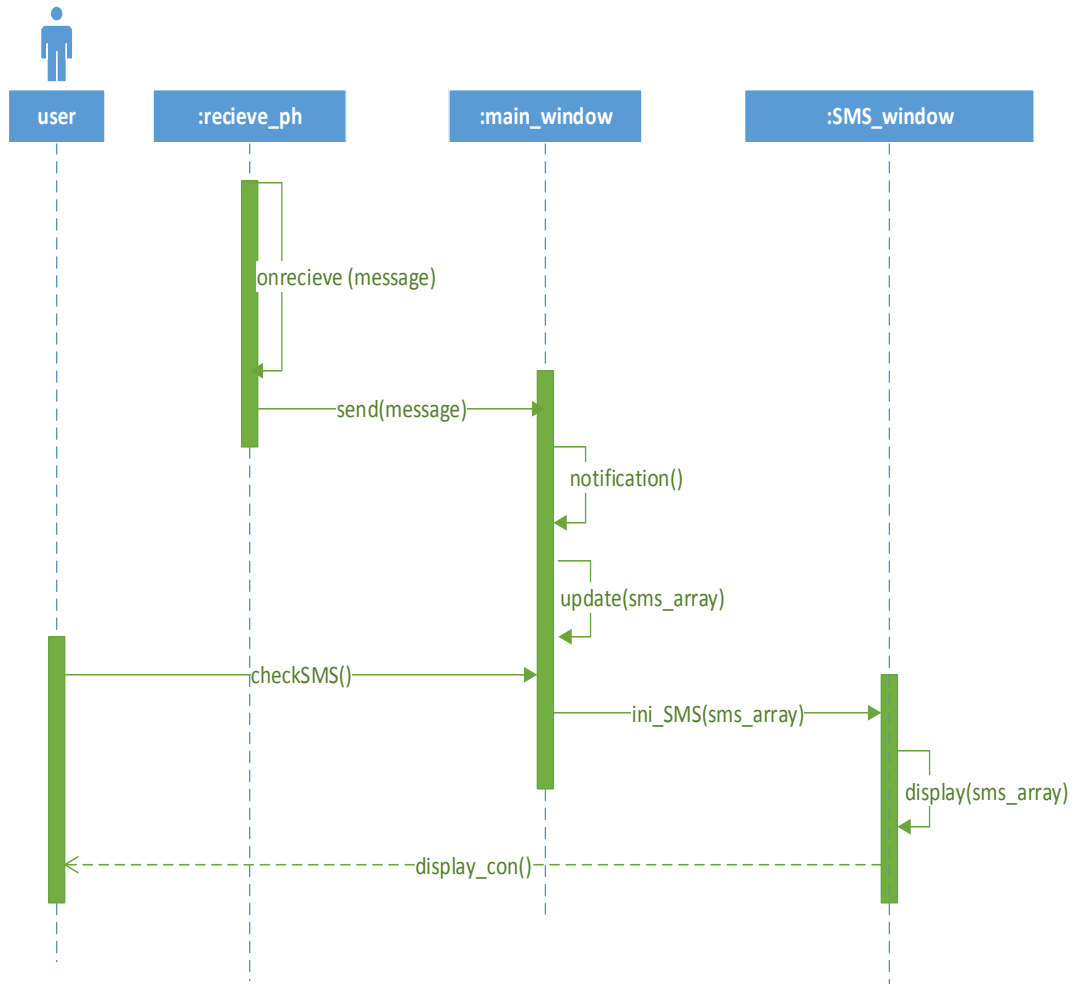


Figure 4-17: Receive SMS Sequence Diagram

4.4.2.4 Data Backup

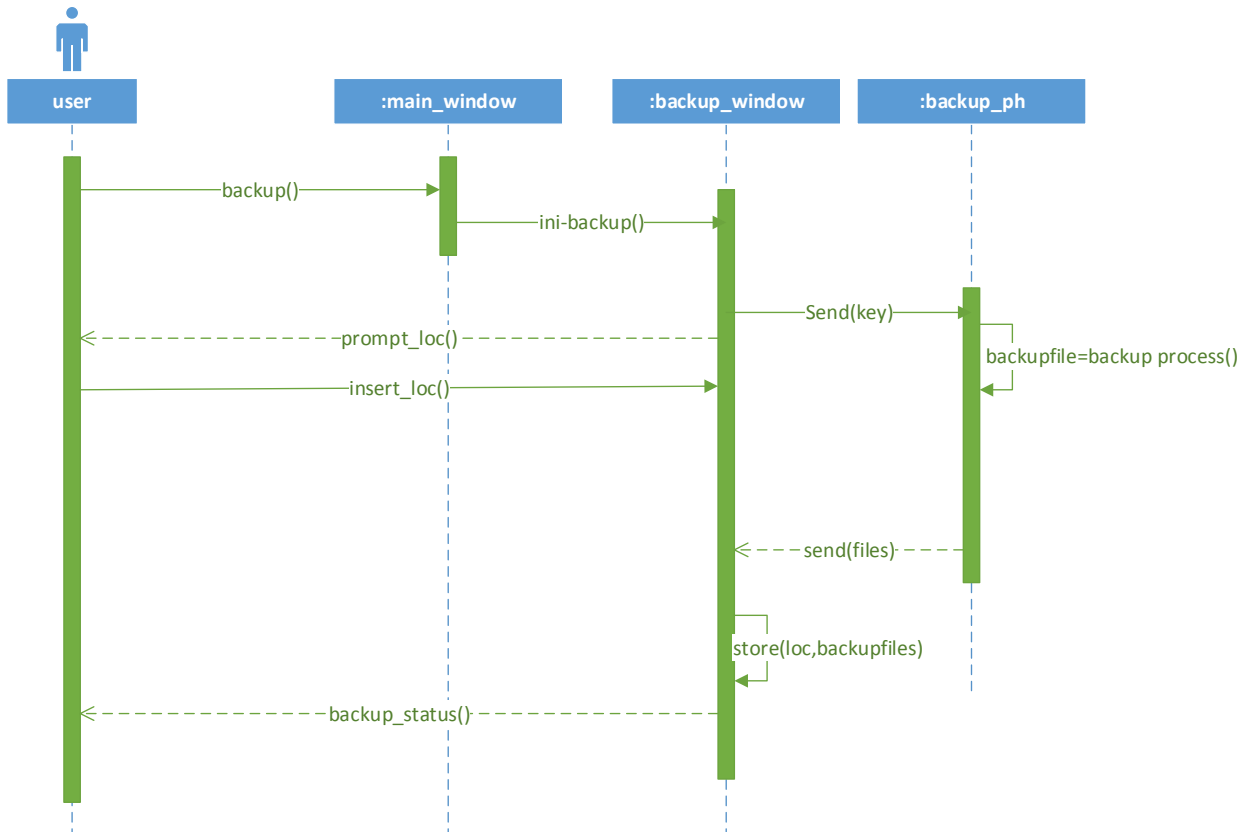


Figure 4-18: Data Backup Sequence Diagram

4.4.2.5 Explore file

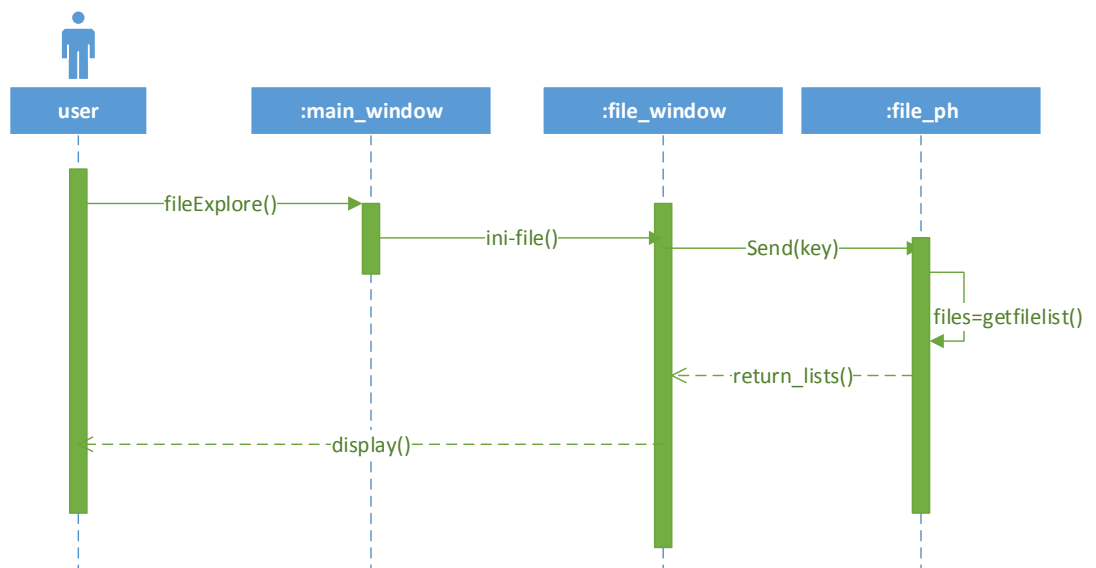


Figure 4-19 File Exploration Sequence Diagram

4.4.2.6 Initiate Call

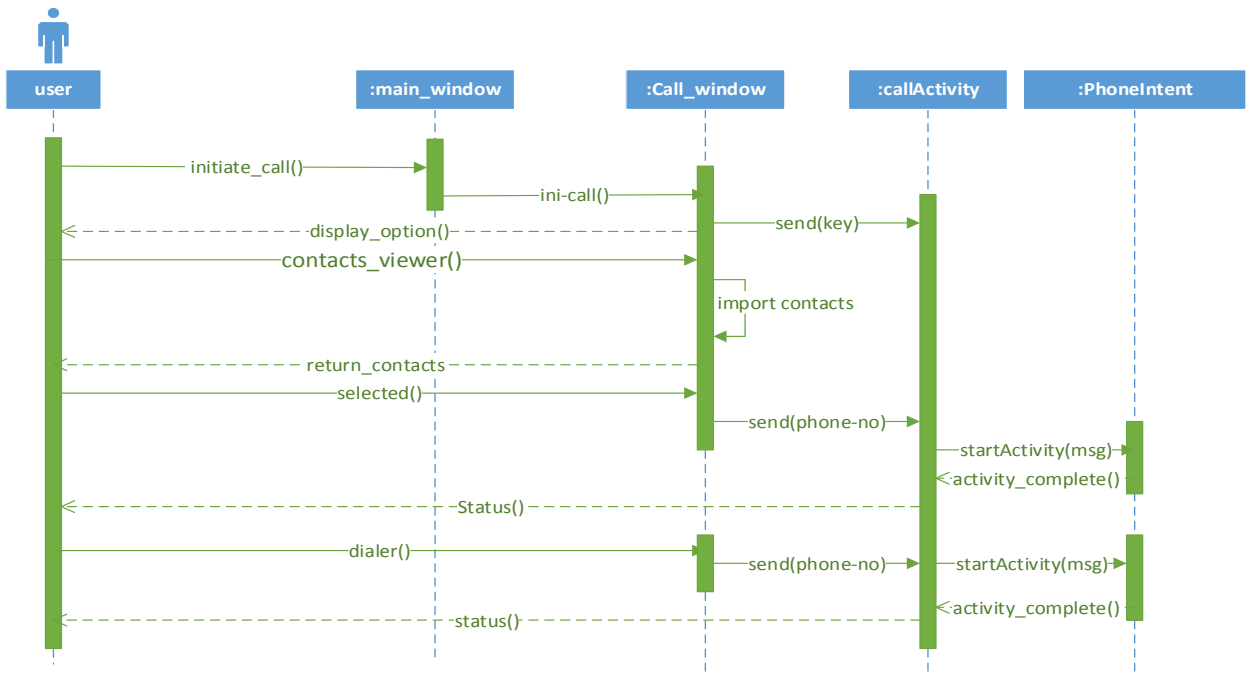


Figure 4-20: Initiate Call Sequence Diagram

4.4.2.7 Remote Camera

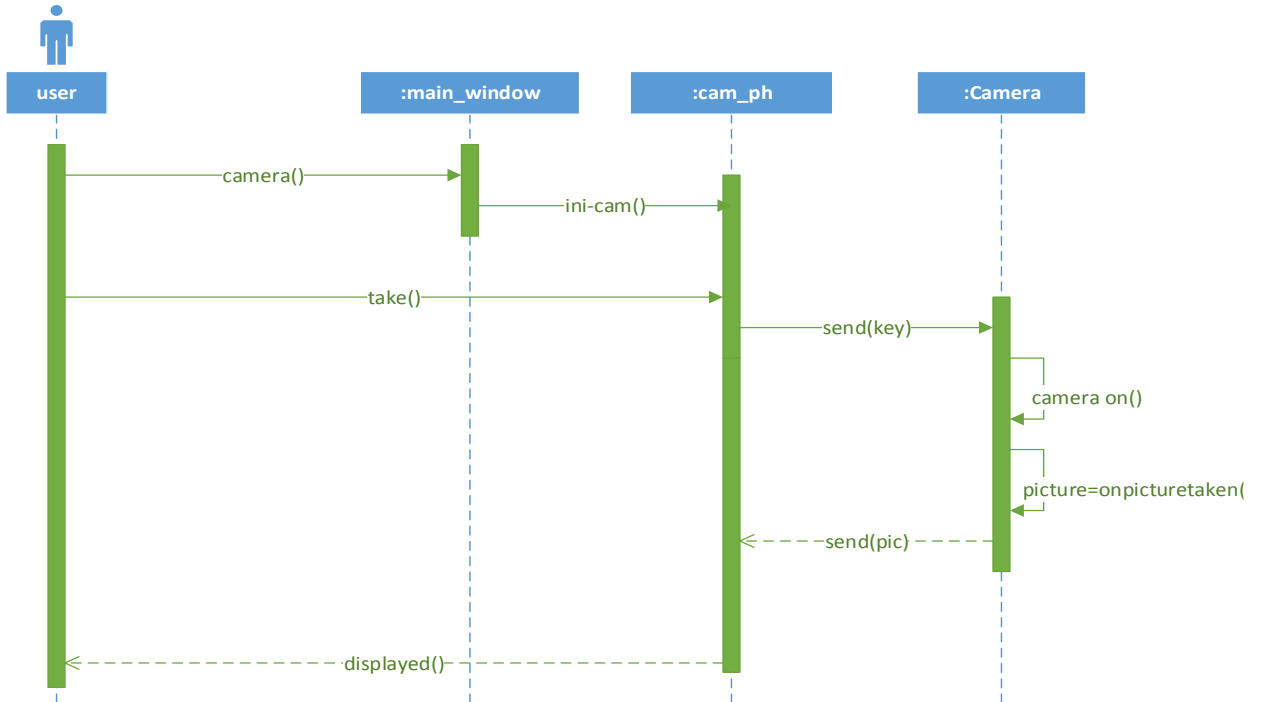


Figure 4-21: Remote Camera Sequence Diagram

4.5 Implementation View

4.5.1 System Class Diagram

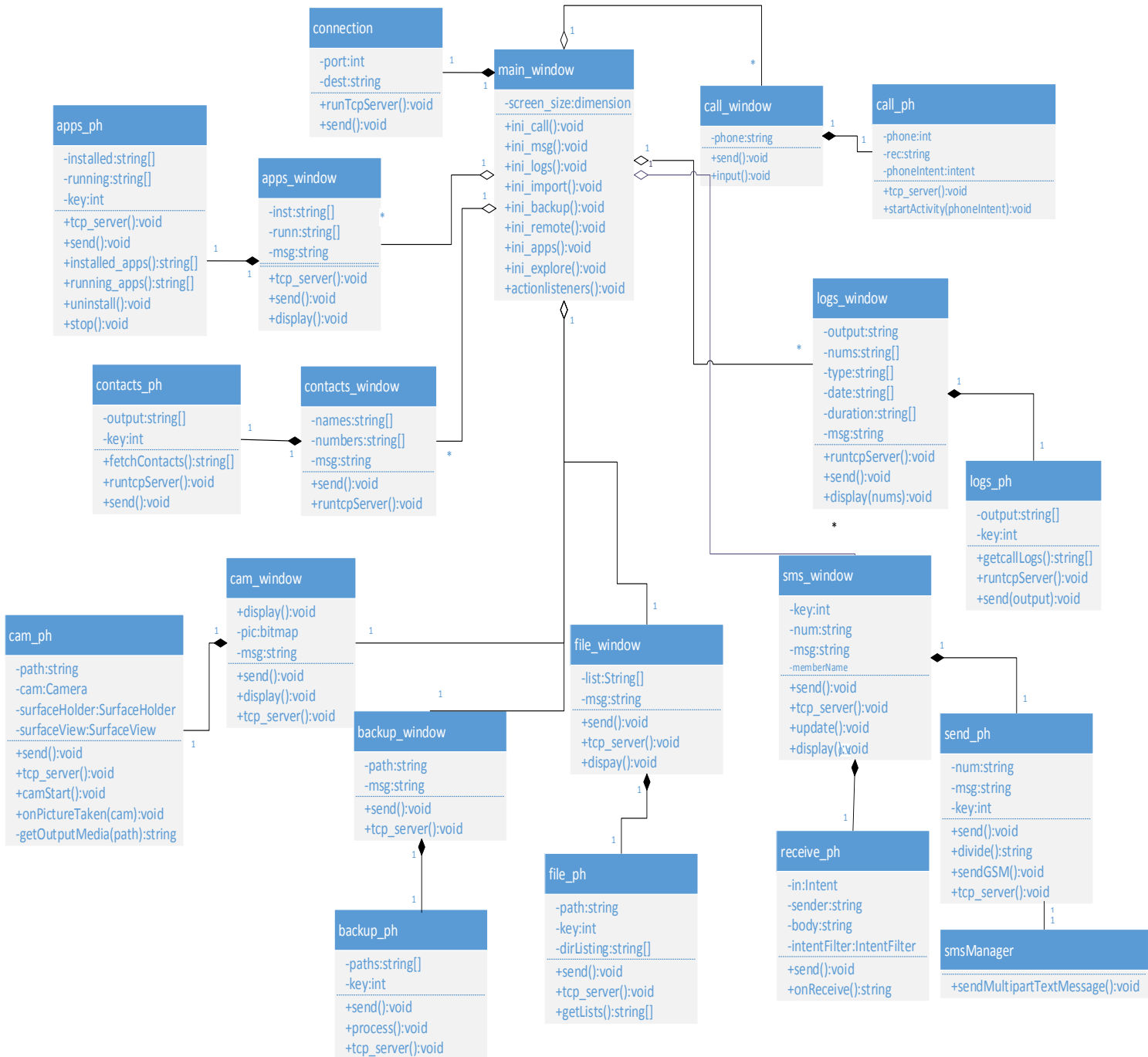


Figure 4-22: System Class Diagram

4.5.2 System Classes Description

4.4.2.8 main_window Class

| Name | main_window | |
|--------------------|---|---|
| Description | This is the main class and contains the user interface of the pc module. It initializes when the application starts. It contains all the functions to call the different interfaces of the application. | |
| Functions | Methods Name | Description |
| | ini_call() | It starts the call window and invokes the required GUI. |
| | ini_msg() | It starts the messaging window and invokes the GUI. |
| | ini_logs() | It starts the logs window and invokes the required GUI. |
| | ini_import() | It starts the contacts window and invokes the GUI. |
| | ini_remote() | It starts the camera window and invokes the GUI. |
| | ini_apps() | It starts the apps window and invokes the required GUI. |
| | ini_explore() | It starts the files window and invokes the GUI. |

| | | |
|--|-------------------|---|
| | actionlisteners() | Behaviours of buttons and GUI objects are defined in this function. |
| | ini_backup() | It starts the backup window and invokes the GUI. |

4.4.2.9 Connection Class

| Name | | Connection |
|-------------|------------------|--|
| Description | | This class is responsible for starting and maintaining the tcp connection between the two devices. It also sends the request codes for initiating different functionalities. |
| Functions | Method Name | Description |
| | run_tcp_server() | It contains the listener that listens to the particular port and receives the traffic. |
| | send(): | Sends messages over socket connection |

4.4.2.10 apps_window

| Name | | apps_window |
|-------------|-------------|---|
| Description | | This class contains the listeners and GUI of the PC module of app management. |
| Functions | Method Name | Description |

| | | |
|--|---------------|--|
| | tcp_server(): | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | display() | Displays the data into the required format. |
| | send(): | Sends messages over socket connection. |

4.4.2.11 apps_ph

| Name | apps_ph | |
|-------------|---|--|
| Description | This class contains the listeners and functions for retrieval of applications that constitute the phone module. | |
| Functions | Method Name | Description |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |
| | installed_apps() | Returns a list of installed apps. |
| | running_apps() | Returns a list of running apps. |
| | uninstall() | Deals with uninstalling an application. |

4.4.2.12 contacts_window

| Name | contacts_window | |
|-------------|--|--|
| Description | This class contains the listeners and GUI of the PC module of importing contacts | |
| Functions | Methods | Description |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection |
| | display(): | Displays the data into the required format. |

4.4.2.13 contacts_ph

| Name | contacts_ph | |
|-------------|--|--|
| Description | This class contains the listeners and GUI of the PC module of importing contacts | |
| Functions | Method Name | Description |
| | fetchContacts() | Returns an array of contacts present in phone |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |
| | display() | Displays the data into the required format. |

4.4.2.14 cam_window

| Name | cam_window | |
|-------------|---|--|
| Description | This class contains the listeners and GUI of the PC module of remote camera. It also contains the functions for processing the picture and displaying it. | |
| Functions | Method Name | Description |
| | display() | It displays the picture received in a specified format. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |

4.4.2.15 cam_ph

| Name | cam_ph | |
|-------------|--|--|
| Description | It contains the variables and methods to initialize and take a picture from the phone's camera and store it. It also sends the picture over the socket connection. | |
| Functions | Method Name | Description |
| | camStart() | Initializes and starts the camera. |
| | onPictureTaken(cam) | Deals with taking and storing the picture in phone's memory. |
| | getOutputMedia(path):. | Retrieves the picture from specified path to be sent over socket connection |
| | send(): | Sends messages over socket connection. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |

4.4.2.16 backup_window

| Name | backup_window | |
|-------------|--|--|
| Description | This class contains the listeners and GUI of the PC module of data backup. | |
| Functions | Method Name | Description |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |

4.4.2.17 backup_ph

| Name | backup_ph | |
|-------------|--|--|
| Description | It contains the methods for extracting files and sending files over the network to the PC module | |
| Functions | Method Name | Description |
| | process() | Extracts specified files from phone to be sent. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |

4.4.2.18 file_window

| Name | file_window | |
|-------------|--|--|
| Description | This class contains the listeners and GUI of the PC module of file exploration | |
| Functions | Method Name | Description |
| | display() | It displays the files and folders into specified distinguishable format. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |

4.4.2.19 file_ph

| Name | file_ph | |
|-------------|---|--|
| Description | It deals with retrieving the directory listings of files and folders and sending them over the socket connection. | |
| Functions | Method Name | Description |
| | getLists() | Returns a list of all the file paths present. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |

4.4.2.20 sms_window

| Name | sms_window | |
|-------------|--|--|
| Description | This class contains the listeners and GUI of the PC module of sms. | |
| Functions | Method Name | Description |
| | display() | Shows the specific GUI. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |
| | update() | Updates the particular conversation thread and variables associated to it. |

4.4.2.21 receiver_ph

| Name | receiver_ph | |
|-------------|---|---|
| Description | This class deals with receiving the GSM text message and transmitting it to the computer. | |
| Functions | Method Name | Description |
| | onrecieve() | Deals with reception of GSM message. Contains the GSM listener and processes the message into a specified format. |
| | send() | Sends messages over socket connection. |

4.4.2.22 send_ph

| Name | send_ph | |
|-------------|--|---|
| Description | Deals with the sending of text message over GSM. | |
| Functions | Method Name | Description |
| | divide() | Processes the message received form PC into specific format that is required for message to be sent over GSM. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |
| | sendGSM() | Sends message over GSM. |

4.4.2.23 logs_window

| Name | logs_window | |
|-------------|--|--|
| Description | This class contains the listeners and GUI of the PC module of call logs. | |
| Functions | Method Name | Description |
| | display() | Displays the logs in specific GUI with particular controls. |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |

4.4.2.24 logs_ph

| Name | logs_ph | |
|-------------|---|--|
| Description | Deals with retrieval and processing of call logs. | |
| Functions | Method Name | Description |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | send() | Sends messages over socket connection. |
| | getCallLogs() | Returns a list of call logs. |

4.4.2.25 call_window

| Name | call_window | |
|-------------|---|--|
| Description | This class contains the GUI and functionality of initiate call. | |
| Functions | Method Name | Description |
| | input() | Deals with user input. |
| | send() | Sends messages over socket connection. |

4.4.2.26 call_ph

| Name | call_ph | |
|-------------|--|--|
| Description | This class deals with initiating the phone call. | |
| Functions | Method Name | Description |
| | tcp_server() | It contains the listener that listens to the particular port, receives the traffic and processes it. |
| | startActivity() | Starts the call. |

4.6 Dynamic View

4.6.1 Activity Diagrams

4.6.1.1 System Activity

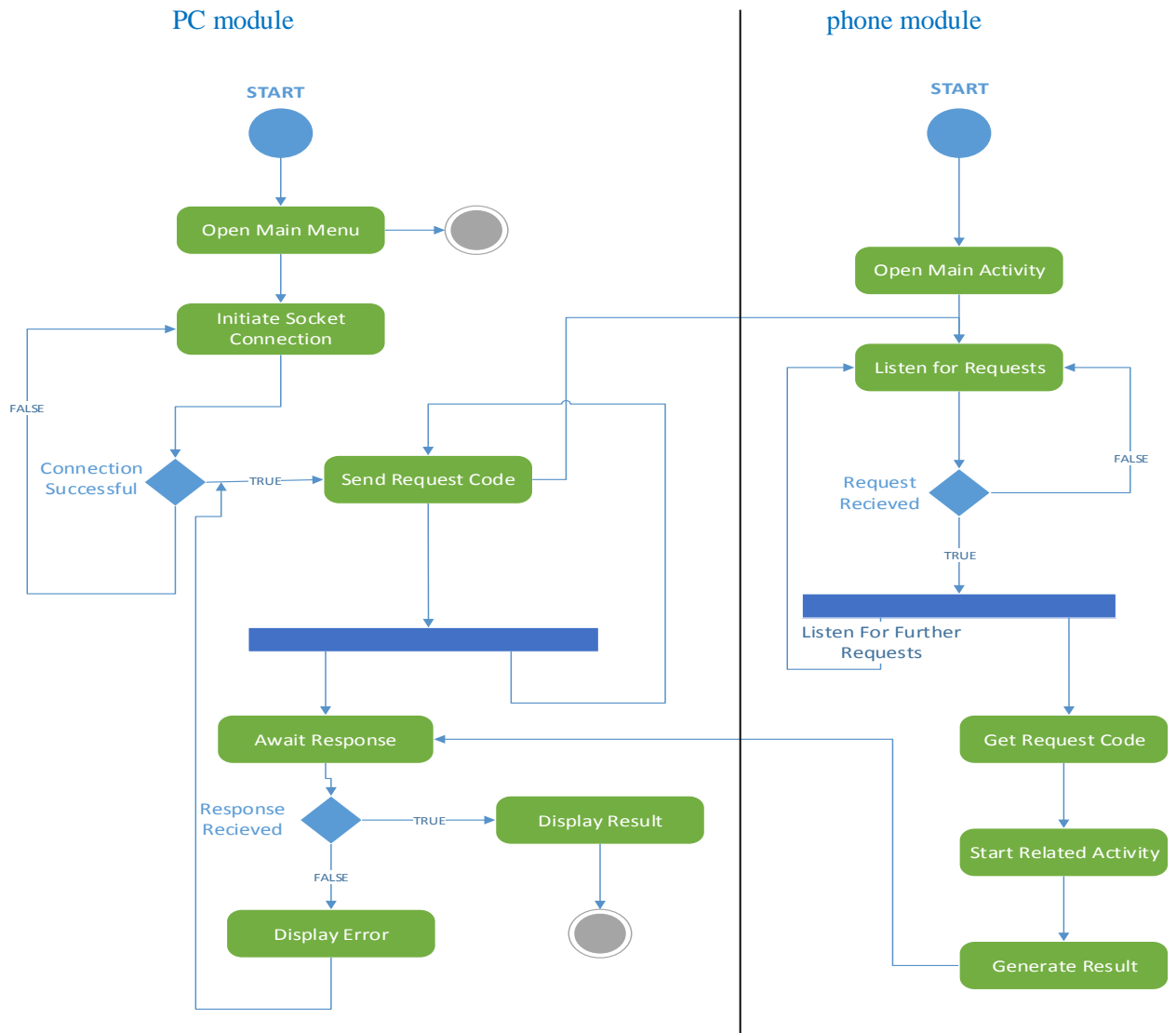


Figure 4-23 system Activity Diagram

4.6.1.2 App Management Activity

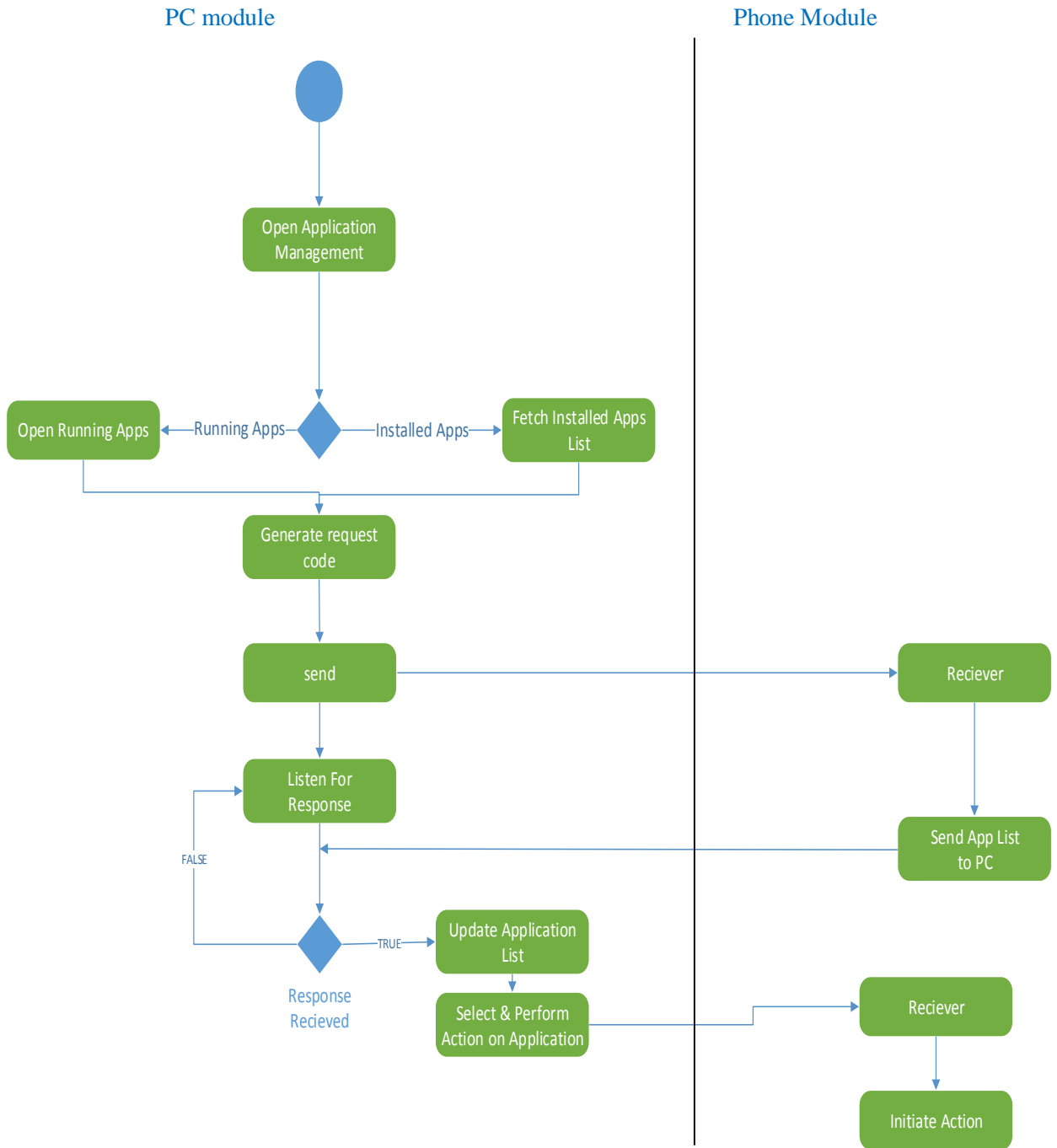


Figure 4-24: App Management Activity Diagram

4.6.1.3 Import Contacts Activity

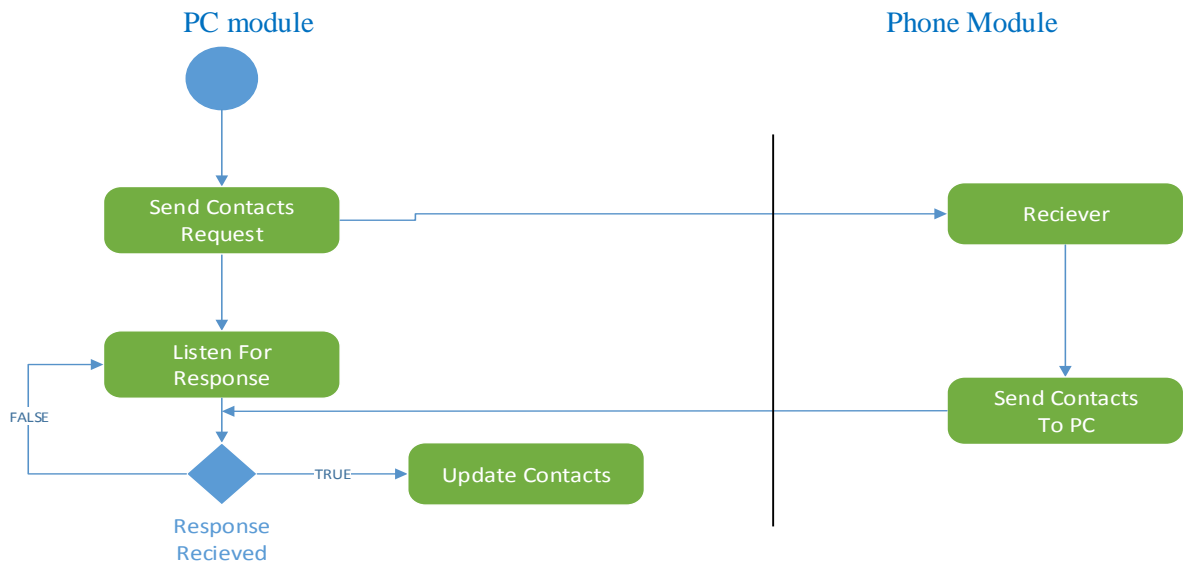


Figure 4-25 Import Contacts Activity Diagram

4.6.1.4 Call-Log Activity

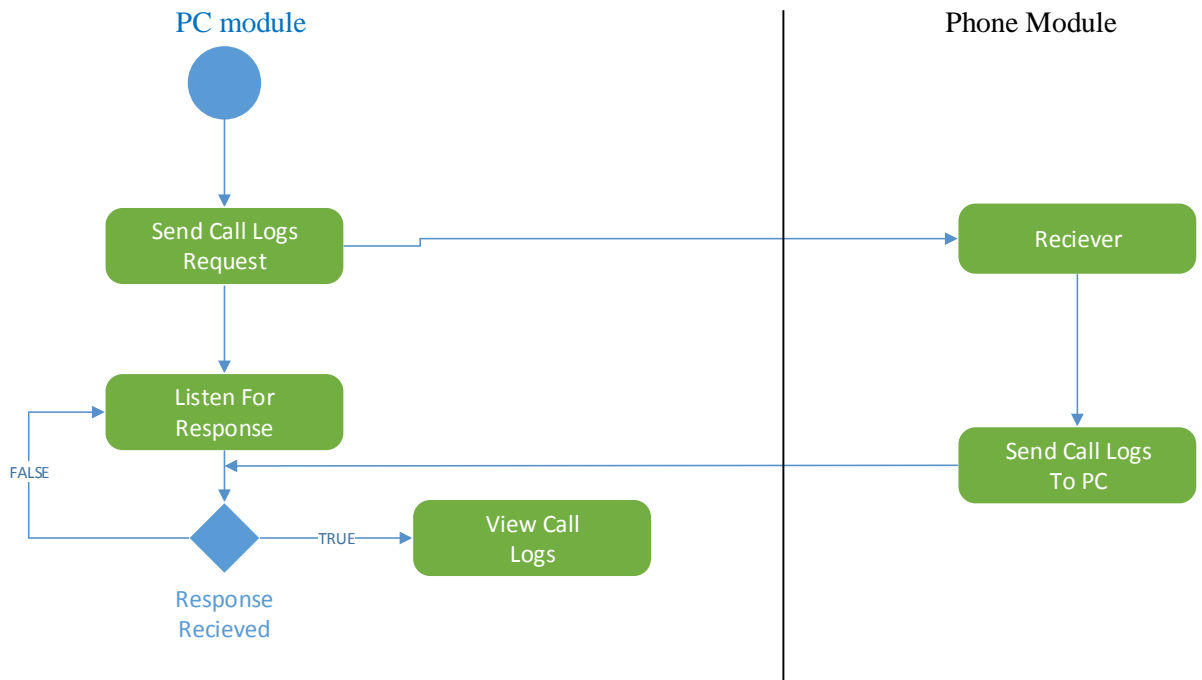


Figure 4-26 Call-Log Activity Diagram

4.6.1.5 Initiate Call Activity

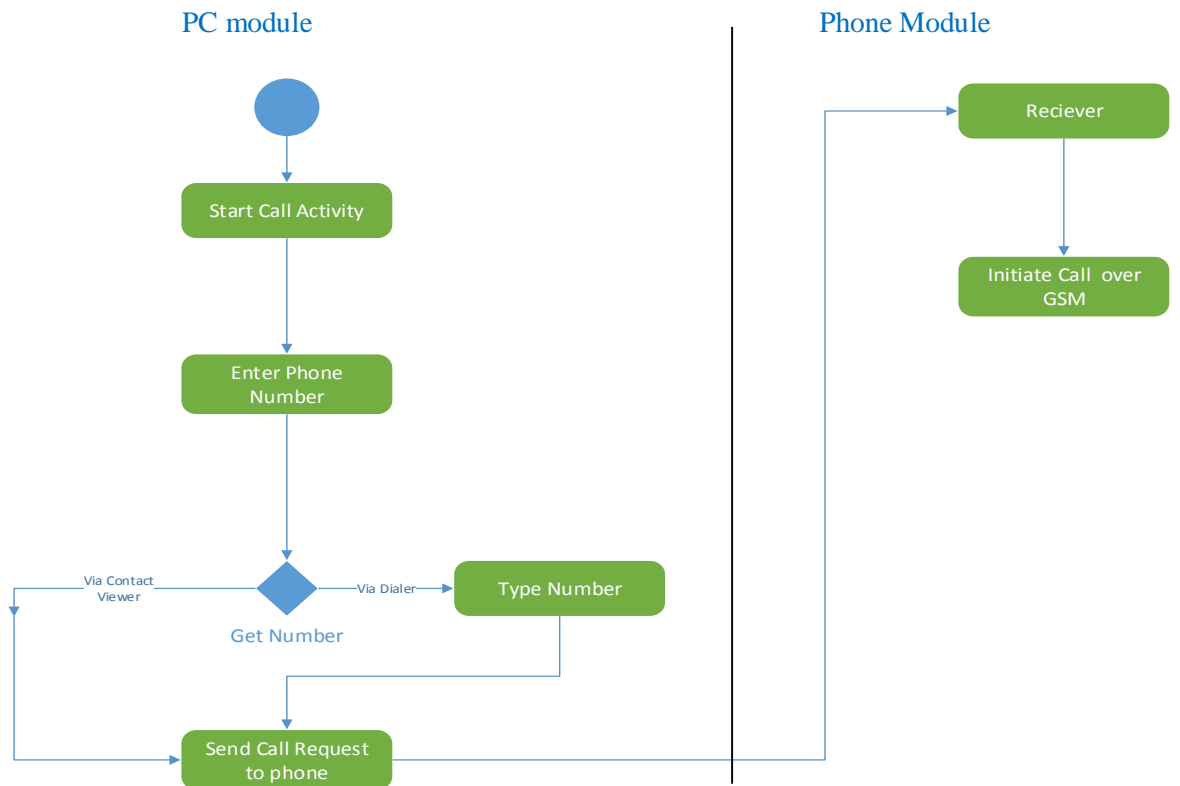


Figure 4-27 Initiate Call Activity Diagram

4.6.1.6 Receive SMS Activity

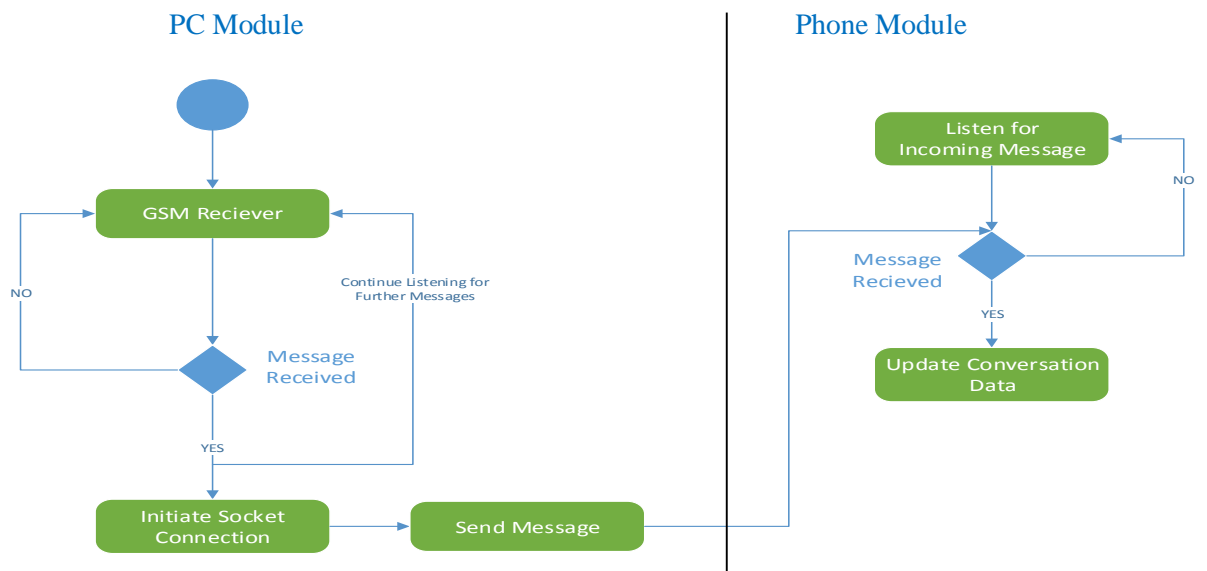


Figure 4-28 Receive SMS Activity Diagram

4.6.1.7 Data Backup Activity

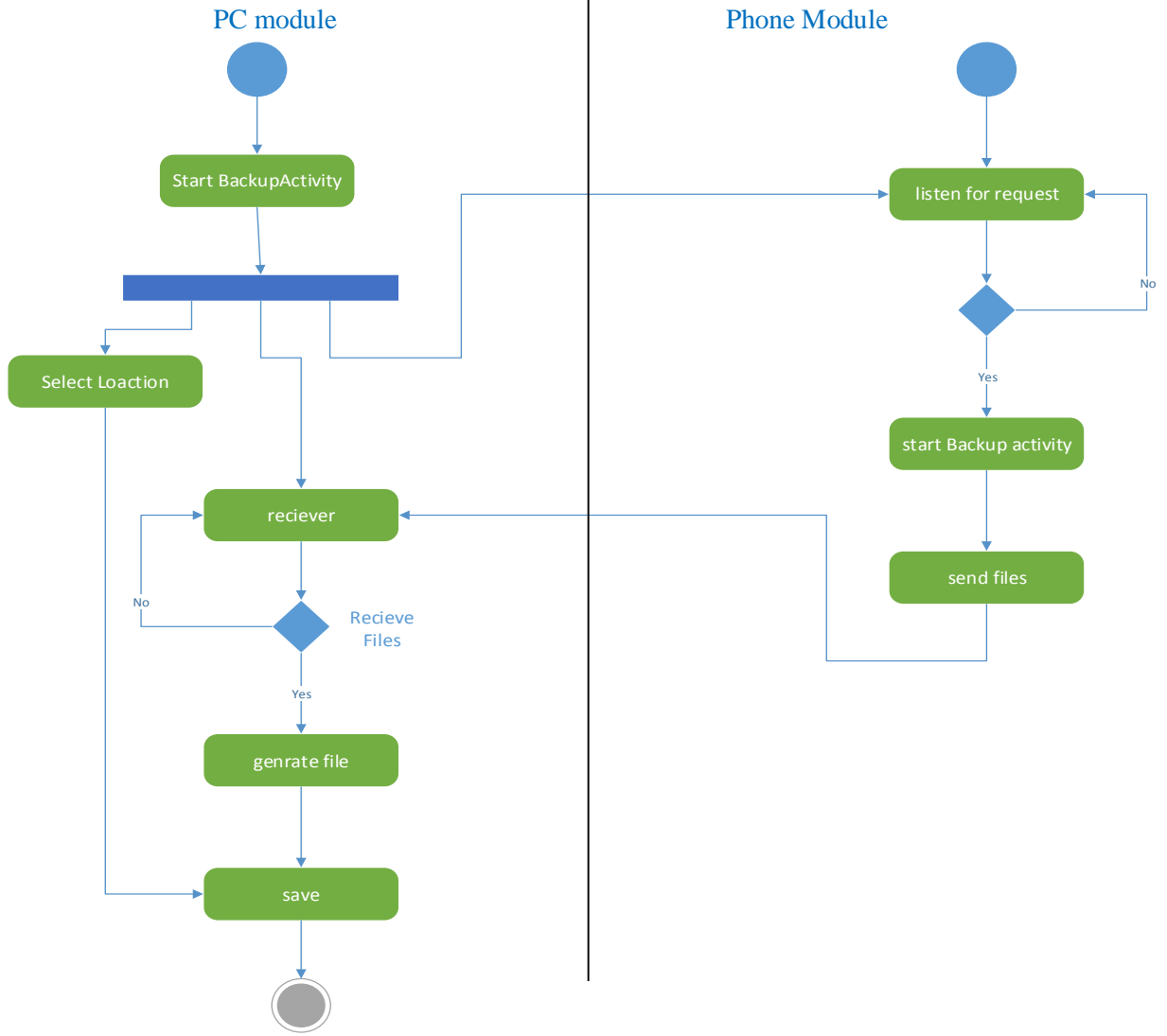


Figure 4-29 Data Backup Activity Diagram

4.6.1.8 File Exploration Activity

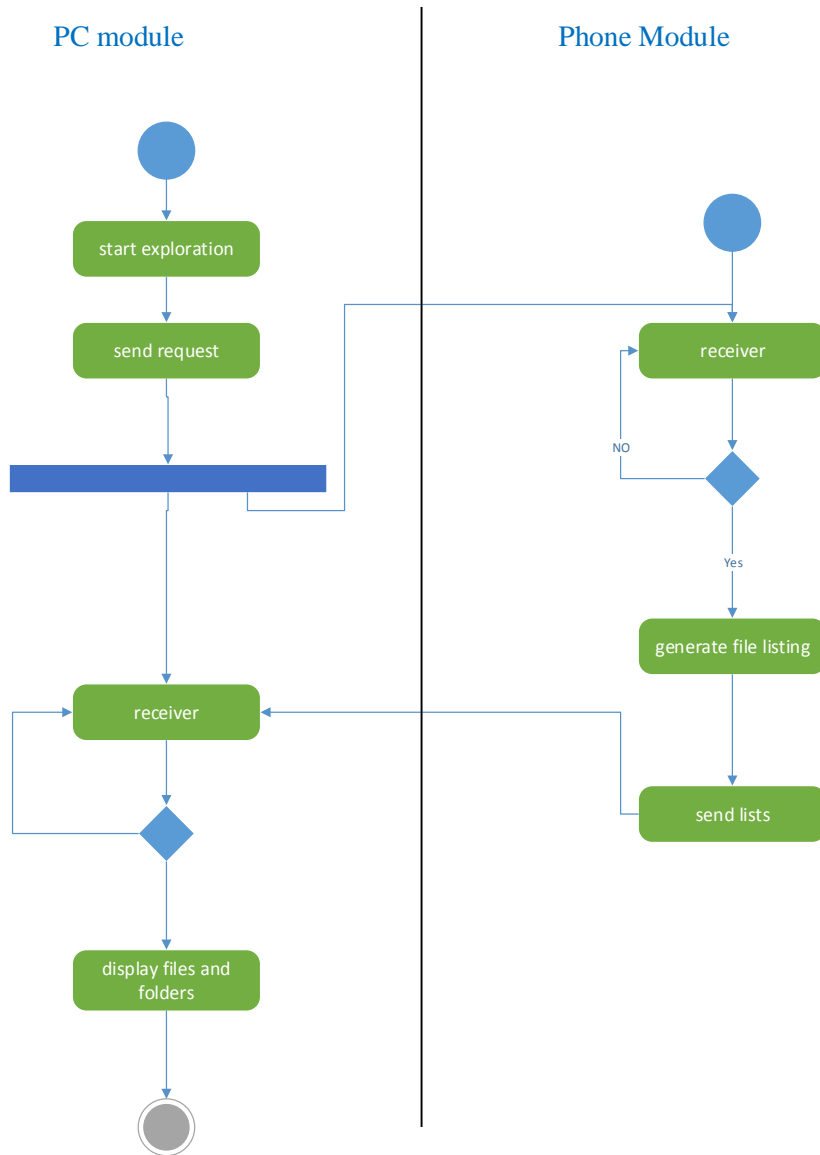


Figure 4-30 File Exploration Activity Diagram

4.6.1.9 Send SMS Activity

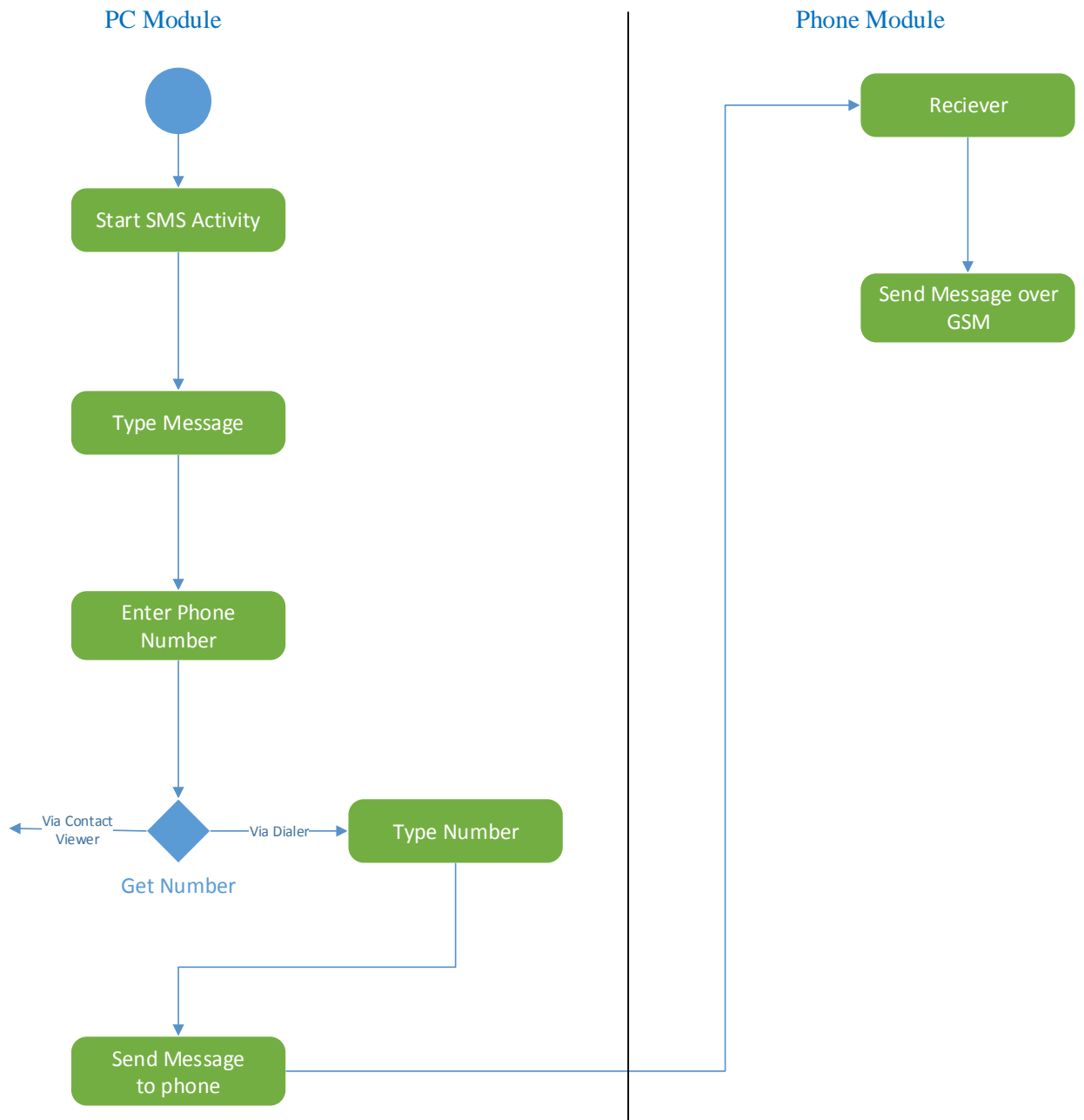


Figure 4-31 Send SMS Activity Diagram

4.6.1.10 Remote Camera Activity

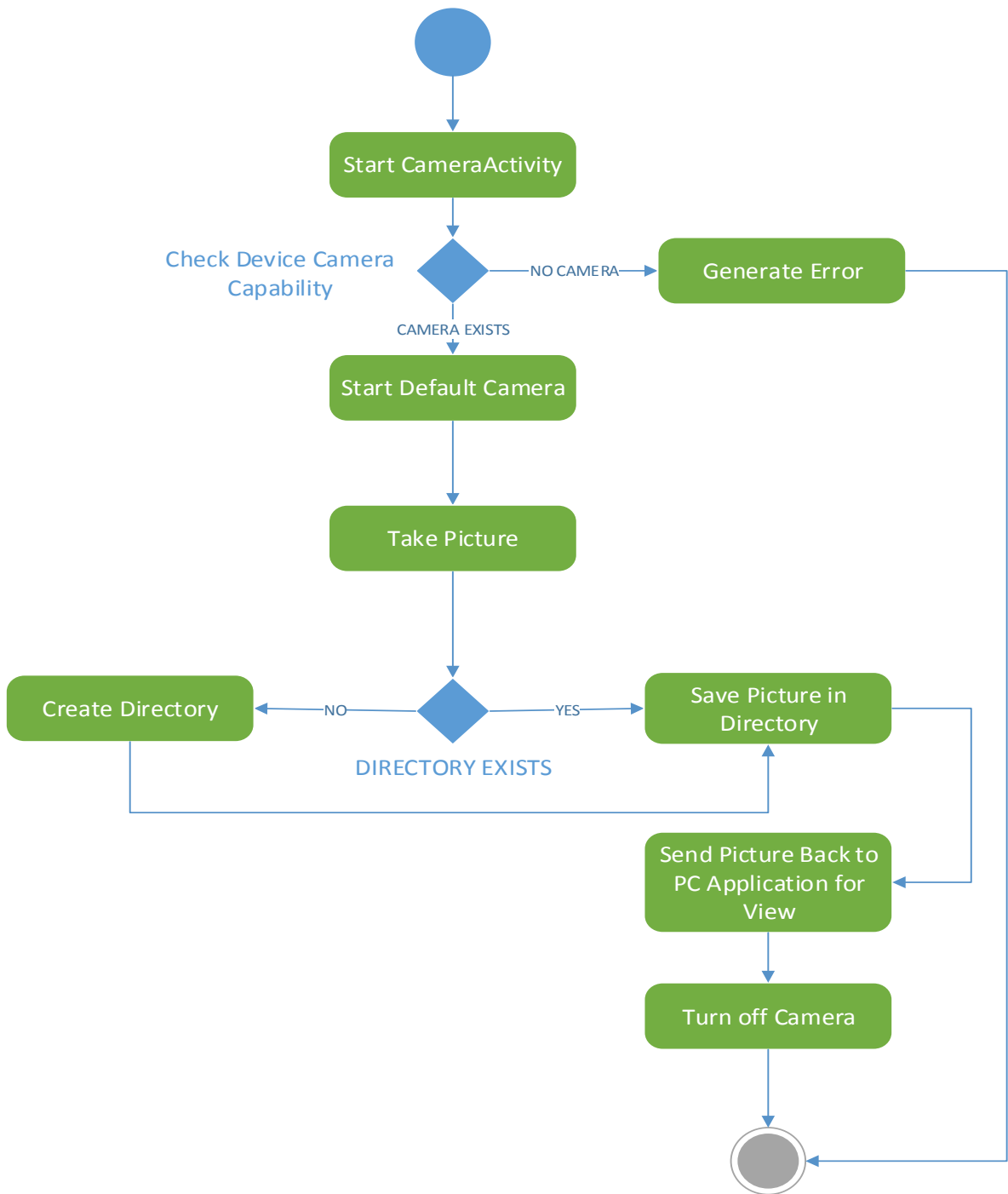


Figure 4-32: Remote Camera Activity Diagram

4.7 User Interface design

The user interface consists of multiple parts.

4.7.1 Main window



Figure 4-33: Main Window UI Design

The main window appears when the application starts. It includes the calling functions of all the other windows. It is main interface with which the user interacts. All the features of the application will be accessed through this window. This window will only appear after the connection has been established between the two devices.

4.7.2 Call window

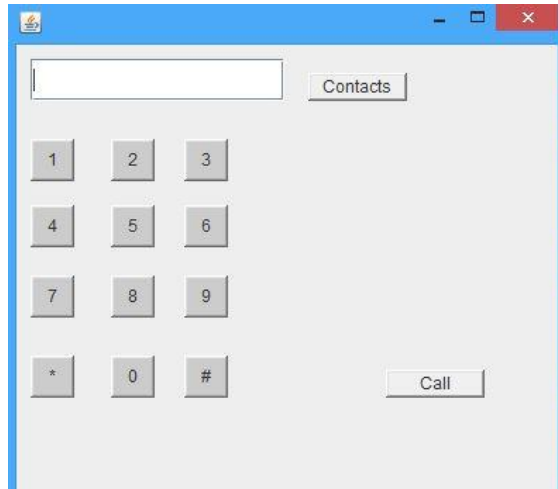


Figure 4-34: Call WindowUI Design

The call window is used for initiate call. The user enters the number or selects a contact and presses call. The call will be initiated on phone. The functionality of initiate call will be accessed through this window.

4.7.3 SMS window

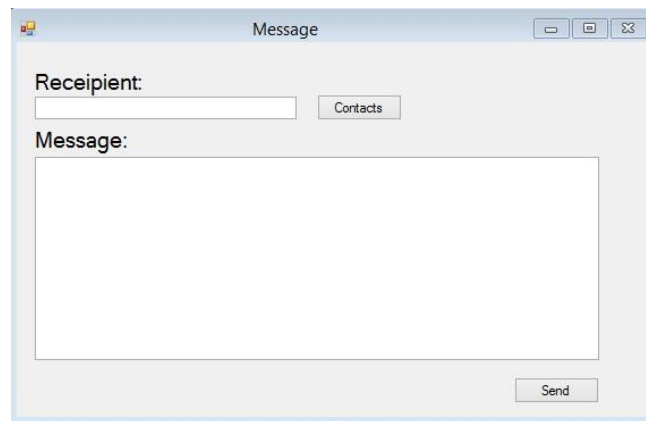


Figure 4-35: SMS Window UI Design

The SMS window will be used for sending and receiving SMS. User will be able to send an SMS by typing message and number.

4.7.4 Files window

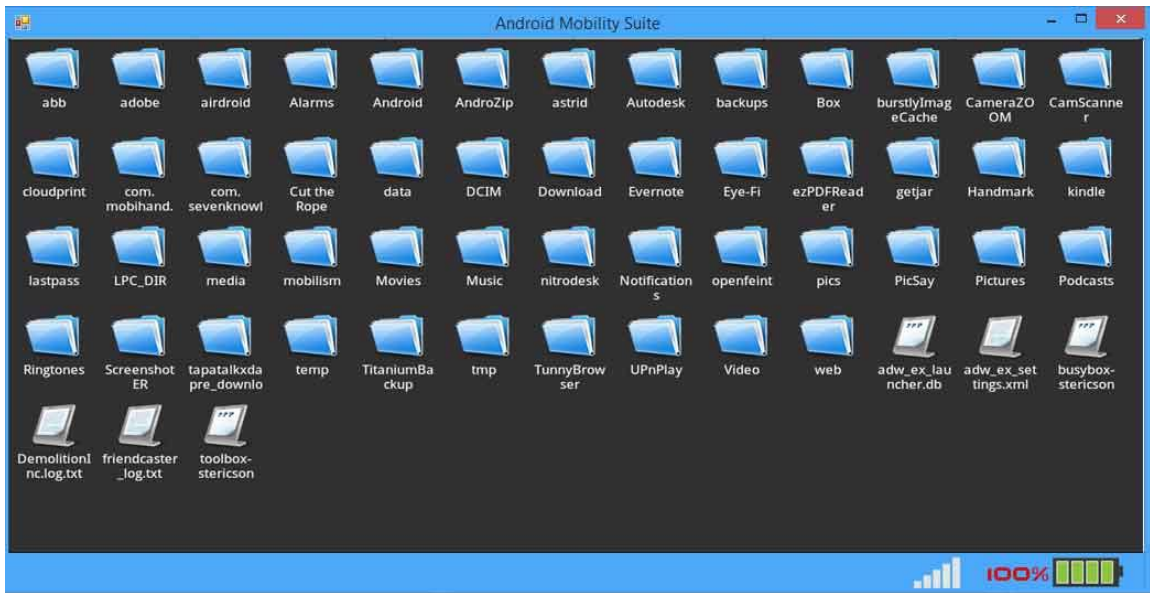


Figure 4-36: File Exploration UI Design

File exploration will be done through this window. The window will provide all the necessary controls to the user as well.

4.7.5 Contacts window

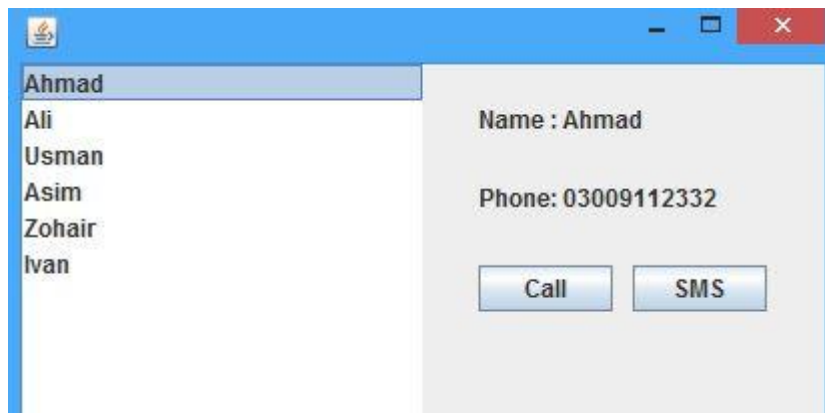


Figure 4-37: Contacts Window UI Design

The contacts window fetches all the contacts from the phone and displays them. It also gives certain controls to the user like call or SMS the selected contact.

4.7.6 Logs window

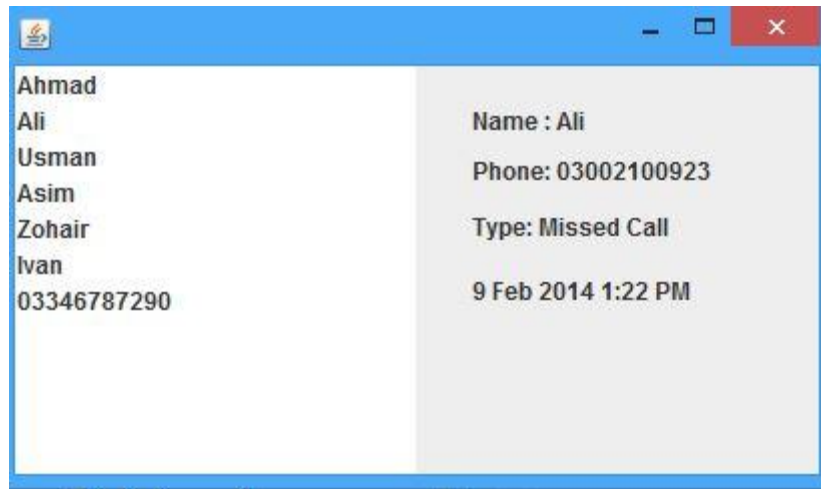


Figure 4-38: Logs Window UI Design

The logs window fetches the call logs from the android phone and displays it.

4.7.7 Apps window

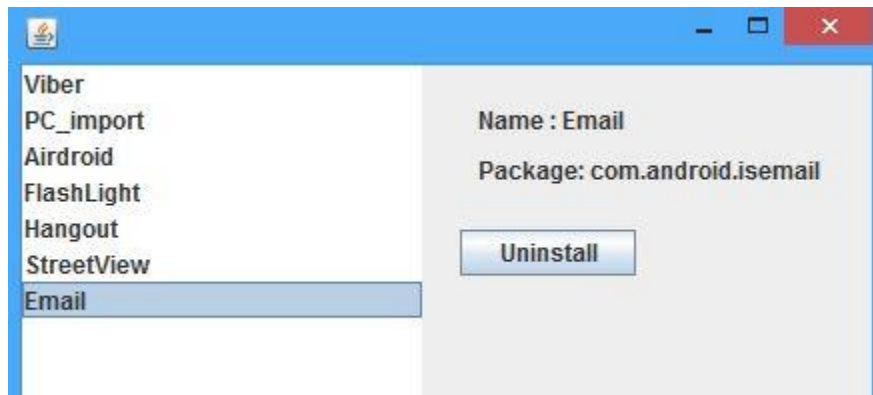


Figure 4-39: Apps Window UI Design

This window shows the running and installed apps on the android phone. It also gives certain controls like stop and uninstall application.

5 System Implementation

5.1 Tools and Technologies

5.1.1 Eclipse SDK

In computer programming, **Eclipse** is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop both android and desktop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages like Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Natural, Perl, PHP, Python, R, Ruby(including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

5.1.2 JavaFX Scene Builder

JavaFX Scene Builder is a visual layout tool that lets users quickly design JavaFX application user interfaces, without coding. Users can drag and drop UI components to a work area, modify their properties, apply style sheets, and the FXML code for the layout that they are creating is automatically generated in the background. The result is an FXML

file that can then be combined with a Java project by binding the UI to the application's logic.

5.2 Software Implementation

The system follows a client server based architecture so there are two apps running one on the phone and one on PC. Some global variables are needed for effective communication between the two devices. These include the IP addresses of both devices plus the ports that the sockets use for communication.

The system uses a set of keys to identify the type of requests. Each request has the key attached to it so the listener can identify the type of request and process the data accordingly. The set of keys is as follows:

| Key value | Request Type |
|-----------|------------------------------------|
| 0 | Authentication message |
| 1 | Initiate call on a number |
| 2 | Return all the contacts in phone |
| 3 | Return all call logs |
| 4 | Return installed applications |
| 5 | Return running applications |
| 6 | Uninstall a particular application |
| 7 | Send SMS |
| 8 | Return all the SMS in phone |
| 9 | Capture a image using camera |
| 10 | File exploration |

5.2.1 Authentication

The authentication involves a generation of passcode on mobile. The pass code consists of six character string. The string is generated using random function of Java language so that there is no repetition. A hash of the string is also generated. The user views the string on the mobile app. The user enters the string on PC. The PC app sends a request to mobile comprising of key, pass code, hash of passcode and its IP address. The mobile app matches the two hashes and if they are same authenticates the user.

5.2.2 Initiate Call

The user enters the phone number. The computer sends a request consisting of key and phone number. The phone number once received is converted into a specific format so that the call could be initiated.

```
number = "tel:" + number_received;
```

The call is then initiated.

5.2.3 SMS implementation

The user enters the message and recipient. The PC sends the request containing the recipient, body and the key of the request. The number and body are then separated. The SmsManager API is being used.

```
smsManager.sendMultipartMessage(Body,recipient);
```

The implementation converts long messages to multiple parts and sends them in multiple messages. It also caters for multiple recipients. In case of received SMS the app updates the database and sends the SMS to PC to be updated in PC application.

5.2.4 Application management

After receiving the request with the corresponding key. The app gets the running and installed apps of the phone. The app gets the installed packages using **getPackageManager().installedPackages()**. The running apps are get using **activityManager.getRunningAppProcesses()**.

The Lists containing apps are sent to PC where they are displayed.

5.2.5 Import Contacts Implementation

Upon receiving the request the app extracts all the contacts present in the phone memory. The **ContactsContract.contacts** is being used. A cursor is used to move through the database to get the required information of all the contacts. After getting all the contacts they are sent to the PC using the socket connection where they are displayed accordingly.

5.2.6 Remote Camera Implementation

The camera works on a number of presets. Upon receiving the request first a surfaceholder is initialized. After this the camera is initialized and all the parameters are set. On pressing the button for taking picture the **camera.takePicture()** takes the picture. Then **onPictureTaken()** method is called. The method converts the picture to bytes and sends them to the PC application where the bytes are reconverted into a picture. The picture is then displayed in an **imageView**.

5.2.7 Backup Implementation

After receiving the request. The mobile app returns the contacts and all the SMS form the mobile phone. The fetchContacts() method is used to get all the contacts whereas SMSretrieve() method is used to get all the SMS in the database. These two are sent to the PC application.

The contacts are stored in a CSV file and the SMS are stored in an XML file. Both are stored on the desired location on the hard disk as provided by the user.

5.2.8 Logs

After receiving the request with the corresponding key. The mobile app retrieves all the call logs using getCallDetails() method. Its uses a Cursor to loop through the database entries and gets the required values like number, type, duration and date etc.

6. Project Analysis and Evaluation

6.1 Testing

6.1.1 Testing Introduction

To ensure quality of the product, testing is conducted. Accuracy and efficiency of tasks performed by our system had to be tested to analyze the system and verify and validate it. Software testing techniques and results obtained are discussed in the coming sections.

6.1.2 Testing Levels

Separate modules were developed to provide different functionalities of the system. All of these modules were tested at different levels during development and after integration.

Different levels at which the system has been tested and results obtained are described in this section.

6.1.3 Unit Testing

Each module was developed and tested individually, each and every component was tested at unit level to ensure that they were functioning properly.

6.1.3.1 Authentication module

Pair wise

Use orthogonal array to design test cases:

| | Enter IP address | Enter security code |
|--------------|-------------------------|----------------------------|
| Row 1 | valid | empty |
| Row 2 | valid | Wrong |
| Row 3 | invalid | Empty |
| Row 4 | Empty | Empty |
| Row 5 | Invalid | Invalid |
| Row 6 | Valid | Valid |

Test cases:

<Valid IP address and empty security code, authentication unsuccessful >

<Valid IP address wrong security code entered, authentication unsuccessful>

<Invalid IP address and empty security code, authentication unsuccessful >

<Empty IP address and empty security code, authentication unsuccessful >

<Invalid IP address and invalid security code, authentication unsuccessful >

<Valid IP address and valid security code, authentication successful >

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|--|-----------------------------|-----------------------------|---------------------------|
| TC 1 | Valid IP address and empty security code | authentication unsuccessful | authentication unsuccessful | Pass |
| TC 2 | Valid IP address wrong security code entered | authentication unsuccessful | authentication unsuccessful | Pass |
| TC 3 | Invalid IP address and empty security code | authentication unsuccessful | authentication unsuccessful | Pass |
| TC 4 | Empty IP address and empty security code | authentication unsuccessful | authentication unsuccessful | Pass |
| TC 5 | I Invalid IP address and invalid security code | authentication unsuccessful | authentication unsuccessful | Pass |
| TC 6 | Valid IP address and valid security code | authentication successful | authentication successful | Pass |

6.1.3.2 Initiate call module

Cases:

Case 1: call made successfully

Case 2: call made successfully

Case3: call unsuccessful

Case 4: call unsuccessful

Case 5: call unsuccessful

Test cases:

<Valid phone number entered, Case 1 executed >

<Valid number form contact list selected, Case 2 executed>

<Invalid number entered, Case 3 executed >

< No number entered, Case 4 executed>

< Invalid number selected from contact list, Case 5 executed>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|---|------------------------|----------------------|---------------------------|
| TC 1 | Valid phone number entered | Case 1 executed | Case 1 executed | Pass |
| TC 2 | Valid number form contact list selected | Case 2 executed | Case 2 executed | Pass |
| TC 3 | Invalid number entered | Case 3 executed | Case 3 executed | Pass |
| TC 4 | No number entered | Case 4 executed | Case 4 executed | Pass |
| TC 5 | Invalid number selected from contact list | Case 5 executed | Case 5 executed | Pass |

6.1.3.3 Send SMS module

Test cases:

Pair wise

Use orthogonal array to design test cases:

| | Enter/select phone No | Enter message |
|--------------|------------------------------|----------------------|
| Row 1 | valid | empty |
| Row 2 | invalid | Not empty |
| Row 3 | invalid | empty |
| Row 4 | valid | Not empty |
| Row 5 | Empty | Empty |

Test cases:

<Valid phone number and empty message entered, want to send empty message? >

<Invalid number not empty message entered, message sending failed>

<Invalid number and empty message entered, message sending failed >

<Valid number and not empty message entered, message sent successfully>

< Empty message box empty phone no, enter recipient>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|--|-----------------------------|-----------------------------|---------------------------|
| TC 1 | Valid phone number and empty message entered | Want to send empty message? | Want to send empty message? | Pass |
| TC 2 | Invalid number not empty message entered | message sending failed | message sending failed | Pass |
| TC 3 | Valid number and not empty message entered | message sending failed | message sending failed | Pass |
| TC 5 | Empty message box empty phone no | enter recipient | enter recipient | Pass |

6.1.3.4 Receive SMS module

Cases:

Case 1: display of phone numbers with which conversations made

Case 2: display conversations of a particular contact number

Test cases:

<Receive SMS button clicked, Case 1 executed >

<Any number from list clicked, Case 2 executed>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|------------------------------|------------------------|----------------------|---------------------------|
| TC 1 | Receive SMS button clicked | Case 1 executed | Case 1 executed | Pass |
| TC 2 | Any number from list clicked | Case 2 executed | Case 2 executed | Pass |

6.1.3.5 Generate backup module

Cases:

Case 1: backup generated successfully

Case 2: low disk space than required

Case 3: cannot copy files here

Case 4: please select destination of backup

Test cases:

<Valid destination selected, Case 1 executed >

<Destination has low disk space, Case 2 executed>

<Restricted destination selected, Case 3 executed>

<Press ok without destination selection, Case 4 executed>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|--|------------------------|----------------------|---------------------------|
| TC 1 | Valid source and destination selected | Case 1 executed | Case 1 executed | Pass |
| TC 2 | Destination has low disk space | Case 2 executed | Case 2 executed | Pass |
| TC 3 | Restricted destination selected | Case 3 executed | Case 3 executed | Pass |
| TC 4 | Press ok without destination selection | Case 5 executed | Case 4 executed | Pass |

6.1.3.6 Call log module

Cases:

Case 1: list of contacts where call made received or missed calls

Case 2: detail of a particular contact time and duration displayed

Case 3: call made successfully

Test cases:

<Call log button clicked, Case 1 executed >

<Any contact from list selected, Case 2 executed>

<Call button on a particular contact is clicked, Case 3 executed>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|--|------------------------|----------------------|---------------------------|
| TC 1 | call log button clicked | Case 1 executed | Case 1 executed | Pass |
| TC 2 | any contact from list selected | Case 2 executed | Case 2 executed | Pass |
| TC 3 | call button on a particular contact is clicked | Case 3 executed | Case 3 executed | Pass |

6.1.3.7 Explore files module

Cases:

Case 1: list of folders displayed

Case 2: list of files displayed

Case 3: list of folders displayed

Case 4: cannot open file

Test cases:

<Explore files button clicked, Case 1 executed >

<any folder is clicked containing files from list displayed, Case 2 executed>

<any folder is clicked containing more folders from list displayed, Case 3 executed>

<any file is clicked from list displayed, Case 4 executed>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|--|------------------------|----------------------|---------------------------|
| TC 1 | explore files button clicked | Case 1 executed | Case 1 executed | Pass |
| TC 2 | any folder is clicked containing files from list displayed | Case 2 executed | Case 2 executed | Pass |

| | | | | |
|-------------|---|-----------------|-----------------|-------------|
| TC 3 | any folder is clicked containing more folders from list displayed | Case 3 executed | Case 3 executed | Pass |
| TC 4 | any file is clicked from list displayed | Case 4 executed | Case 4 executed | Pass |

6.1.3.8 Capture image

Cases:

Case 1: image view displayed on PC showing capture image and save image button

Case 2: image displayed on image view on PC

Test cases:

<Camera button clicked, Case 1 executed >

<Click capture image button, Case 2 executed>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|---------------------|----------------------------|------------------------|----------------------|---------------------------|
| TC 1 | Camera button clicked | Case 1 executed | Case 1 executed | Pass |
| TC 2 | Click capture image button | Case 2 executed | Case 2 executed | Pass |

6.1.3.9 Application management module

Cases:

Case 1: window showing buttons for running application and installed applications

Case 2: list of installed applications and a uninstall button

Case 3: want to uninstall the application message on phone

Case 4: application uninstalled successfully

Case 5: list of installed applications

Case 6: list of running applications displayed

Test cases:

<Application management button clicked, Case 1 executed >

<Click installed application button, Case 2 executed>

<Uninstall button clicked, Case 3 executed>

<Click ok on mobile phone, Case 4 executed>

<Click cancel on mobile phone, Case 5 executed>

<Click running button, Case 6 executed>

| Test case ID | Input | Expected output | Actual output | Status (pass/fail) |
|--------------|---------------------------------------|-----------------|-----------------|--------------------|
| TC 1 | Application management button clicked | Case 1 executed | Case 1 executed | Pass |
| TC 2 | Click installed application button | Case 2 executed | Case 2 executed | Pass |
| TC 3 | Uninstall button clicked | Case 3 executed | Case 3 executed | Pass |
| TC 4 | Click ok on mobile phone | Case 4 executed | Case 4 executed | Pass |
| TC 5 | Click cancel on mobile phone | Case 5 executed | Case 5 executed | Pass |
| TC 6 | Click running button | Case 6 executed | Case 6 executed | Pass |

6.1.4 Integration Testing

This Test Plan describes the integration tests that will be conducted on the implementation of android mobility suite.

The interfaces between the following modules will be tested:

1. Connection
2. Menu
3. Initiate call

4. Call log
5. Send SMS
6. Receive SMS
7. Data backup
8. File explorer
9. Remote Camera
10. Application management

It is assumed that unit testing already provided. Wi-Fi signals are available.

SYSTEM ARCHITECTURE:

The aim is to check if the modules after integrating with each other perform as they were required. The user of the system should be able to make successful connection after entering valid codes and should be able to perform functionalities like initiating call, view call logs, send message ,receive message, manage applications, explore files and generate backup .The system performance would be tested under valid and invalid inputs.

Functional Testing will focus on each and every use case that is included in the version currently being worked on. Testing will mainly consists of execution of test cases written to address the gap identified. It will focus on inputs, outputs and system changes due to the actions. User can initiate call from his PC

1. User can send and receive SMS using his PC
2. Data Backup
3. File Exploration and Data Management
4. Call logs management
5. Import contacts.

6. User can use camera of mobile phone from PC
7. Application management

For this purpose we are using the incremental approach to integrate the modules together.

Establish connection between the client and server:

First of all we will integrate two basic modules. Client application (PC module of the application) with server (mobile module of the application) to ensure that client is successfully making connection with server so that requests could be handled. We will first perform the test cases to validate their interaction. We will make sure that PC module is successfully connecting to mobile module.

Data transfer:

After this we can start integrating the modules related to transfer of data between mobile and PC.

First we will integrate connection with the main menu and then with the initiate call module and make sure that it provides the correct results. We will then integrate it with the call log module to see if the that when we view logs and make call through this module then initiate call is providing correct functionality input there. In case of any errors, after debugging and satisfaction we will incrementally add contacts module to check that if the module after integration with the already integrated modules works fine and doesn't disturb their performance as well and also names of contacts are displayed correctly instead of phone numbers in call log and initiate call can view list obtained from contacts and select number instead of entering new number. After this add send SMS module to ensure list of contacts provides input correctly to it for selecting a number and they are working perfect together

then add receive SMS module after that application management and then file explorer after the correct functionality of all these integrate data backup with them.

The advantage is that the integration testing process can start earlier, as soon as related modules have been successfully unit tested. An incremental approach makes it easier to find errors since the application environment introduces only one (or a few) modules at a time. Finally, this approach results in more overall testing, since the earlier modules get tested repeatedly as you add new modules.

6.1.4.1 Structure of Integration levels:

Integration test phases:

a. First phase of testing

In this phase tests will be designed to uncover functional errors in each module after it is integrated with the system. Errors associated with local or global data structures will be uncovered with such tests.

We will integrate the connection module with menu and then with call module and make sure that it provides the correct results. We will then integrate it with the call log module to see if the that when we view logs and make call through this module then initiate call is providing correct functionality input there. Previously designed unit tests for checking the functionality of initiating call will be performed again and also unit test cases for call log will be performed again

In case of any errors, after debugging and satisfaction we will incrementally add contacts module to check that if the module after integration with the already integrated modules works fine and doesn't disturb their performance as well and

also names of contacts are displayed correctly instead of phone numbers in call log and initiate call can view list obtained from contacts and select number instead of entering new number. After this add send SMS module to ensure list of contacts provides input correctly to it for selecting a number and they are working perfect together then add receive SMS module after that application management and then file explorer after the correct functionality of all these integrate data backup and camera with them.

b. Second phase of testing

Pair wise validity will be performed in which we will combine two modules together and check the functionality which is dependent on both modules. First call log and initiate will be tested pair wise. As call log is dependent on initiate call so check this dependent functionality. Then contacts and call log are tested pair wise as contacts provide contact name to each number so check whether instead of number now contact is providing name to the call log.

Then initiate call and contacts tested pair wise as initiate call uses contact list also with entering number. After this send SMS and contacts is tested pair wise as SMS use contacts list. Then receive SMS is paired with contacts as list of contacts is displayed instead of numbers after integration. Application management, file explorer, camera and backup are paired with connection and menu.

In end to end testing test cases will be performed which include testing of almost all modules like user after entering code establishes connection and chooses take send SMS form main menu.

c. Third phase of testing

In third phase endurance testing will be performed that how much stress our system can with stand i.e. how many applications of mobile can be controlled at a time by this application and for how many days or hours our system can keep working without breakdown

6.1.4.2 Modules to be integrated in each phase:

a. First phase

In first phase we need to check functional validity of all the modules. So, we will use all of our modules in this phase.

b. Second phase

In end to end tests. We will utilize all of our modules to perform the task starting from one of the module (e.g. initiate call) to the module that accepting information from other modules (e.g. call log)

c. Third phase

For checking the stress the system can withstand we will again require our whole integrated system in place. This phase validates that critical functions will meet production performance requirements during peak transaction volumes.

6.1.4.3 Building process and Schedule for each phase:

a. First phase

We started first phase on 30th April and checked for functionality of each module after integration and it took almost 4 hours so ended at the same day

b. Second phase

The second phase checks end to end validity. It started on 1st may 2014 after the 1st phase. It will be completed in 7 hours.

c. Third phase

The endurance testing is the last phase in integration testing. It started on 2nd may 2014 .System endurance will be checked by testing it for 4 continuous hours under different conditions and for valid and invalid inputs.

6.1.4.4 Environment to be set up and resources required in each phase:

Make sure that computer is running Windows with Wi-Fi signals available

- 64/32 bit windows OS
- Testers and developers set up manual and configuration guides
- Wi-Fi signals
- Android phone

6.1.4.5 Criteria for each Integration test phase n:

a. First phase:

i. Entry criteria:

Entry criteria for functional testing is that each module has been unit tested individually and all errors have been corrected. Interface integrity ensures that the two modules communicate with each other in the desired way.

ii. Exit criteria:

All modules are integrated incrementally. Exit criteria for the first phase is fully functionally tested modules integrated together. In the end we get a successfully *functionally* tested integrated system. The integrated system has passed all the integration tests and no defect is outstanding.

iii. Integration Technique to be used:

Incremental technique

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

b. Second phase:

i. Entry criteria:

Functionality of each module is tested after integrations and validated, is Entry criteria for End to end and pair wise.

ii. Exit criteria:

Fully integrated system with no more modules to integrate is Exit criteria .we get a, All the system integration tests for the phases have passed, all the defects found have been fixed and also documented.

iii. Integration Technique to be used:

Incremental technique

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

c. Third phase:

i. Entry criteria:

The entry criteria of this phase is that we have a validated system that works together from end to end. There are no major defects remaining while testing the end to end working of the integrated system.

ii. Exit criteria:

The exit criteria of this phase is that we have an integrated system to which when the load is applied can sustain the load and does not crash. For example opening multiple windows at a time for file exploring camera etc. Also system endurance is ensured during this phase so that in the end we have a system that can work for hours without crashing.

iii. Integration Technique to be used:

Incremental technique.

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

6.1.4.6 Test specification for each integration phase:

a. First phase:

This phase will include performing tests on integrated modules to check their interface integrity and functional validity. Same test cases as used in unit testing for modules functionality are tested again by replacing stubs with actual modules. These are given below:

i. Interface Integrity

| Test Case Id | Input | Initial conditions | Test procedure | Expected Result | Actual Result | Status |
|--------------|------------------------------|--|--|---|---|--------|
| TC1 | Wrong code | Application has executed and screen asking for code is displayed | When screen appeared for code input is provided through keyboard | Display error message | Error message | Pass |
| TC2 | Correct code | Application has executed and screen asking for code is displayed | When screen appeared for code input is provided through keyboard | Main menu screen shown | Main menu screen shown | Pass |
| TC3 | Initiate call button clicked | Main menu displayed | From the main menu initiate call button is pressed through mouse | Window showing keyboard, contacts button and textbox is shown | Window showing keyboard, contacts button and textbox is shown | Pass |
| TC4 | Enter number from the | Window showing keyboard, contacts | Enter number from the keyboard | Dialing call status is shown | Dialing call status is shown | Pass |

| | | | | | | |
|------|---|--|---|---|---|------|
| | keybo ard | button and textbox is shown | | | | |
| TC5 | Call log button clicked | Main menu displayed | From the main menu call log button is pressed through mouse | List of names displayed | List of names displayed | Pass |
| TC6 | Name from the list clicked | List of names displayed | click name from the list with mouse | Details of calls received dialed missed with time and duration displayed | Details of calls received dialed missed with time and duration displayed | Pass |
| TC7 | Contact s button clicked | Main menu displayed | From the main menu contacts button is pressed through mouse | List of names displayed | List of names displayed | Pass |
| TC8 | Name from the list clicked | List of names displayed | click name from the list with mouse | Number of that contact name is displayed | Number of that contact name is displayed | Pass |
| TC 9 | Send SMS button clicked | Main menu displayed | From the main menu send SMS button is pressed through mouse | Window showing text box for number and text box for message is displayed | Window showing text box for number and text box for message is displayed | Pass |
| TC10 | Enter message and number and press send button | Window showing text box for number and text box for message is displayed | Enter message and number and press send button | Message sent successfully | Message sent successfully | Pass |
| TC11 | Receiv e SMS button clicked | Main menu displayed | From the main menu receive SMS button is pressed through mouse | Window showing the names is displayed | Window showing the names is displayed | Pass |
| TC12 | Applicat ion manage ment button is clicked | Main menu displayed | From the main menu application management button is pressed through mouse | Window showing the installed and uninstalled buttons displayed | Window showing the installed and uninstalled buttons displayed | Pass |
| TC13 | File explorer s button | Main menu displayed | From the main menu files explorer button is | Window showing the files and | Window showing the files and folders list is displayed | Pass |

| | | | | | | |
|-------|-------------------------------|---------------------|--|---------------------------------|---------------------------------|------|
| | is clicked | | pressed through mouse | folders list is displayed | | |
| TC 14 | Data backup button is clicked | Main menu displayed | From the main menu data backup button is pressed through mouse | Window of data backup is opened | Window of data backup is opened | Pass |
| TC 15 | Camera button is clicked | Main menu displayed | From the main menu data camera is pressed through mouse | Window of camera is opened | Window of camera is opened | Pass |

b. Second phase:

This will include designing test cases to check end to end validity. In our system this can be checked by designing test cases that will start from getting user's credentials from the login screen till the result is retrieved.

For Pair Wise:

| Modules integrated | Test Case Id | Input | Initial conditions | Test procedure | Expected Result | Actual Result | Status |
|----------------------------|--------------|---|---|--|-------------------------------|-------------------------------|--------|
| Connection and menu | TC1 | Correct code | System is in running condition with code input screen displayed | Enter the correct code from keyboard | Main menu is displayed | Main menu is displayed | Pass |
| initiate call and call log | TC2 | Make call button clicked | User is in call log window | Select call from the menu of call log by using mouse | List of contacts is displayed | Call initiated displayed | Pass |
| Contacts initiate call log | TC3 | Button of contacts is clicked form initiate call window | User is in initiate call window | Click button of contacts form initiate call window | List of contacts is displayed | List of contacts is displayed | Pass |

| | | | | | | | |
|----------------------------|-----|---|-------------------------------|--|---|---|------|
| Contacts initiate call log | TC4 | Extraction of contact list pressed | List of contacts is displayed | Select a contact from list and click call button | List of contacts is displayed | List of contacts is displayed | Pass |
| Contacts and call log | TC5 | Call log button clicked | Main menu displayed | Click call log button from main menu with mouse | Name of the contacts instead of number displayed | Name of the contacts instead of number displayed | Pass |
| Contacts send SMS | TC6 | Click contacts button in send sms window | Send SMS window displayed | Click contacts button in send sms | List of names of contacts is displayed | List of names of contacts is displayed | Pass |
| Contacts send SMS | TC6 | Click contact name from the list displayed in send SMS window | List of contacts displayed | Click contact name from the list in send sms and press send button | Status of message sent is shown | Status of message sent is shown | Pass |
| Contacts and receive SMS | TC7 | Receive SMS button clicked | Main menu displayed | Click receive SMS button | Contact names with which conversations made are displayed | Contact names with which conversations made are displayed | Pass |

For the End to End:

For end to end testing we will perform a test which will include almost all the modules so that it can be checked that they all are working as required together.

We will start from user input code, then main menu will be displayed and after that all the features are checked for all functionalities from main menu.

c. Third phase:

This will include designing test cases to check system endurance:

Due to high availability, load test carries great importance. The metrics are as follows:

| Metrics | Minimum | Good | Excellent |
|---|---------|------|-----------|
| Maximum Number of Concurrent Operations Supported | 1 | 5 | 8 |

Endurance Test cases:

| Test Case Id | Input | Initial condition | Test Procedure | Expected Result | Actual Result | Status |
|--------------|--------|--------------------|---|---|---|--------|
| TC1 | 4 days | System is executed | System functions are checked for 4 days without termination | System works properly, no unexpected errors with condition that host machine is running normally. | System works properly, no unexpected errors with condition that host machine is running normally. | Pass |
| TC2 | 1 week | System is executed | System functions are checked for 1 week without termination | System works properly, no unexpected errors with condition that host machine is running normally. | System works properly, no unexpected errors with condition that host machine is running normally. | Pass |

Actual test results for each integration test phase:

The Actual results are already shown in the test cases specifications for each integration phase above.

6.1.5 System Testing

System testing was performed at the end of development and integration of all the components. The system testing enabled us to detect bugs and defects in our code as well as assembly of the system and improve the system by removing a maximum number of them. The current library of test cases helps us to keep development teams on track and

also verify that system conforms to requirements which are described in the requirement document.

6.1.5.1 Basic Tests

a. Boot tests:

| Test Case Id | Input | Initial conditions | Test procedure | Expected Result | Actual Result | Status |
|--------------|--|---|--|---|--------------------------------|--------|
| TC1 | System launched by clicking button on system file that was saved in memory | Integrated and Working Version is Available | The exe file for the built system is clicked as it is the available boot option provided | The system boots up its image from memory | System gets its software image | Pass |

6.1.5.2 Functionality Tests

a. Communication System Tests:

The System requires that both devices be connected with a same Wireless Access Point.

b. Module tests:

Same as done in integration testing.

c. GUI tests:

| Test case id | Input | Initial condition | Test procedure | Expected output | Actual output | Status |
|--------------|--|--|--|--|----------------------------------|--------|
| TC1 | Users checks the system's GUI | System has executed and screen is opened | When screen appeared. testing team checks whether the font is readable and text is interpretable | Team should find the GUI user friendly | Team found the GUI user friendly | Pass |
| TC2 | Users calculate the system navigate time | System has executed and screen is opened | User navigates through whole GUI | The navigation time should not be more than 3 to 4 seconds | The navigation time is 3 seconds | Pass |

d. Security Tests

i. Availability and Integrity tests:

| Test Case Id | Input | Initial conditions | Test procedure | Expected Result | Actual Result | Status |
|--------------|--------------------------|--|--|---|---|--------|
| TC1 | Valid code input of user | Integrated and Working Version is Applicable | Valid code for the user will be provided to the system | Information validated user will be able to view main menu | Information validated and user will be able to view main menu | Pass |

ii. Confidentiality tests:

| Test Case Id | Input | Initial condition | Test Procedure | Expected Result | Actual Result | Status |
|--------------|--------------|--------------------|----------------------|--|--|--------|
| TC1 | Invalid code | System is executed | Invalid code entered | Error message of invalid code displayed and Access not allowed | Error message of invalid code displayed and Access not allowed | Pass |

6.1.5.3 Scalability Tests

Not Applicable

6.1.5.4 Performance Tests

a. INTEROPERABILITY AND COMPATIBILITY TESTS:

| Test case id | Input | Test procedure | Expected output | Actual output | Status |
|--------------|--------------|---|--|----------------------------------|--------|
| TC1 | Windows test | Execute the software on windows 7/8 and check every functionality that the system is compatible with windows platform | System should work on that environment | System works on that environment | Pass |

6.1.5.5 Robustness Tests

a. Boundary value tests:

Not applicable.

b. Power cycling tests:

| Test Case Id | Input | Initial condition | Test Procedure | Expected Result | Actual Result | Status |
|--------------|--------------|--------------------|--|--|--|--------|
| TC1 | Power glitch | System is executed | System checked that after power glitch any error occurred on not | Connection will break and need to restart the system | Connection will break and need to restart the system | pass |

c. High-Availability Tests:

Not applicable

6.1.5.6 Documentation Tests

| Test Case Id | Input | Initial condition | Test Procedure | Expected Result | Actual Result | Status |
|--------------|---------------------------------------|------------------------|--|---|-----------------------------------|--------|
| TC1 | User manual is viewed by testing team | System is not executed | Team reads the manual and execute the system side wise | System should be successfully executed and used by testing team | Team successfully used the system | pass |

6.1.6 Acceptance Testing

This is acceptance test plan for Android Mobility Suite. By testing we want to achieve that the system matches the customer requirements and is acceptable by them. This is the first

version of the software, it is built to provide ease in bringing usability of features of android smartphone on your personal computer. By this test plan we will test the software for the quality attributes according to the customer acceptance criteria.

Date of approval: 5nd May 2013.

6.1.6.1 Functional Correctness and Completeness

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows xp/7/8.

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|-----------------------|--|--|
| TC 1 | Connection In Feature | Is to test that the authentication feature's functionality is according to what user requirement is and do the work it is programmed to do | Run the software and provide the correct code ,if the menu appears, it will show that the test is successful |
| TC 2 | Initiate call feature | Is to test that the initiate call feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of initiate call window displayed showing keyboard and contact button to enter number .enter number by both the ways to check it initiates proper call it will show that test is successful |
| TC 3 | Call log feature | Is to test that the call log feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of call log. Window displayed showing name of contacts click a contact details are displayed. press call button if initiates call then call log feature is working good it will show that test is successful |
| TC 4 | Contacts feature | Is to test that the contacts feature works as it is | Establish Connection then menu displayed click button of contacts |

| | | | |
|------|--------------------------------|--|--|
| | | supposed to do according to requirements | Window displayed showing name of contacts click a contact and press call button call initiated then test is successful |
| TC 5 | Send SMS feature | Is to test that the send SMS feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of send SMS. Window displayed showing test box for entering number and text box for message then enter number through contacts check sending message by both the ways. press send if message sent then send SMS feature is working good it will show that test is successful |
| TC6 | Receive SMS feature | I Is to test that the receive SMS feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of receive SMS. Window displayed showing contact names click on contact conversations are displayed is working good it will show that test is successful |
| TC 7 | Application management feature | Is to test that the application management feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of application management. Window displayed showing button for installed and running click on installed and press uninstall, then click on running and view running applications all the functionalities are according to requirements it is working good it will show that test is successful |
| TC8 | File explorer feature | Is to test that the file explorer feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of file explorer. Window displayed showing files and folder explore through folders if this shows successful test |
| TC9 | Data backup feature | Is to test that the data backup feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of data backup. Window displayed for entering destination location if backup created successfully then this shows successful test |
| TC10 | Remote camera feature | Is to test that the remote camera feature works as it is supposed to do according to requirements | Establish Connection then menu displayed click button of camera. Window displayed showing the image view and capture image button and save image button |

| | | | |
|--|--|--|---|
| | | | .capture image then save it if image saved successfully then this shows successful test |
|--|--|--|---|

6.1.6.2 Accuracy

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows xp/7/8.

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|-----------------------|---------------------------------------|--|
| TC 1 | Call log feature | Is to test the accuracy of the result | Establish Connection then menu displayed click button of call log. Window displayed showing name of contacts click a contact details are displayed. press call button if initiates call then call log feature is working good it will show that test is successful |
| TC 2 | Contacts feature | Is to test the accuracy of the result | Establish Connection then menu displayed click button of contacts Window displayed showing name of contacts if result is as it was expected then test is successful |
| TC 3 | File explorer feature | Is to test the accuracy of the result | Establish Connection then menu displayed click button of file explorer. Window displayed showing files and folder explore through folders if this shows as expected then successful test |
| | Data backup feature | Is to test the accuracy of the result | Establish Connection then menu displayed click button of data backup. Window displayed |

| | | | |
|--|------------------------|---------------------------------------|--|
| | | | showing the source and destination locations if backup created as it was expected then test successful |
| | Application management | Is to test the accuracy of the result | Establish Connection then menu displayed click button of application management. Window displayed showing button for installed and running view both running and installed applications if result is as it was expected then test successful |

6.1.6.3 Data Integrity

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows xp/7/8.

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|---------------|---|--|
| TC 1 | Contacts list | To test the data preservation i.e. contacts remain unchanged when retrieved by another operation like send SMS or initiate call | Establish Connection then menu displayed click button of contacts. Window displayed showing the contacts and now retrieve the contact list both by send sms feature and initiate call feature if both the times list are same then test successful |

6.1.6.4 Backup and Recoverability

Operational Environment:

The operational environment required to test the software is the operating system and database server, it requires. As it is not hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8. Backup call records and application is generated and is stored in a separate 1tb hard disk

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|-------------|---|--|
| TC 1 | Data backup | To test that if system fails or shuts down the data backup is save. | Establish Connection then menu displayed click button of generate backup fill source and destination then backup is generated now shut down the system abruptly Restart again and check if the backup still exists if it exists then test successful. |

6.1.6.5 Usability

Operational Environment:

The operational environment required to test the software is the operating system. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|-----------------------------|--|---|
| TC 1 | User Interface: Main screen | The objective is to check whether the main screen provides enough options that are easy to use so that the user can navigate through the system. | Login this will take user to the main screen. Various options will be present through which the user can select the desired option. User can click initiate call, call log, application management, file explorer, data backup, send & receive SMS, and remote camera |

| | | | |
|------|---|--|--|
| TC 2 | User Interface: Initiate call | Determine whether the user can initiate call with relative ease. | Establish Connection then menu displayed click button of initiate call window displayed showing keyboard and contact button to enter number .enter number by both the ways to check it initiates proper call it will show that test is successful |
| TC 3 | User Interface: Call log | | Establish Connection then menu displayed click button of call log. Window displayed showing name of contacts click a contact details are displayed. press call button if initiates call then call log feature is working good it will show that test is successful |
| TC4 | User Interface: contacts | Determine whether the user can access contacts with relative ease. | Establish Connection then menu displayed click button of contacts Window displayed showing name of contacts click a contact and press call button call initiated then test is successful |
| TC5 | User Interface: Send SMS | Determine whether the user can send message with relative ease. | Establish Connection then menu displayed click button of send SMS. Window displayed showing test box for entering number and text box for message then enter number through contacts check sending message by both the ways. press send if message sent then send SMS feature is working good it will show that test is successful |
| TC7 | User Interface: Receive SMS | Determine whether the user can receive message with relative ease. | Establish Connection then menu displayed click button of receive SMS. Window displayed showing contact names click on contact conversations are displayed is working good it will show that test is successful |
| TC8 | User Interface: Application management | Determine whether the user manage applications with relative ease. | Establish Connection then menu displayed click button of application management. Window displayed showing button for installed and running click on installed and press uninstall, then click on running and view running applications all the functionalities are according to requirements it is working good it will show that test is successful |
| TC9 | User Interface: File explorer | Determine whether the user explore files with relative ease. | Establish Connection then menu displayed click button of file explorer. Window displayed showing files and folder explore through folders if this shows successful test |
| TC10 | User interface: Remote camera | Determine whether the user uses remote camera with relative ease. | Establish Connection then menu displayed click button of camera. Window displayed showing the image view and capture image button and save image button .capture image then save it if image saved successfully then this shows successful test |

| | | | |
|------|--------------------------------|---|--|
| TC11 | User interface: Data backup | Determine whether the generate backup with relative ease. | Establish Connection then menu displayed click button of data backup. Window displayed showing the image view and capture image button and save image button .capture image then save it if image saved successfully then this shows successful test |
|------|--------------------------------|---|--|

6.1.6.6 Performance

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows XP/7/8.

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|-----------------------|---|--|
| TC 1 | Display response time | To test the systems displaying response time according to the customer criteria | Establish connection then main menu appears click on any of features if response time is according to defined then test successful |
| TC 2 | Opening response time | To test the systems opening time according to the customer criteria | Open the software and check the response time until code screen shows up. |

6.1.6.7 Robustness

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|-------|-----------|-----------|
|--------------|-------|-----------|-----------|

| | | | |
|------|-------------------|---|---|
| TC 1 | Code input screen | Determine how the system reacts when erroneous inputs are given. | Login invalid input generates message of invalid code and asks to re-enter |
| TC 2 | Initiate call | Determine how the system reacts when erroneous inputs are given. | Enter contact number. If number is wrong error message displays that number is not correct call failed |
| TC 3 | Send sms | Determine how the system reacts when erroneous inputs are given. | Enter contact number. If number is wrong error message displays that number is not correct message sending failed the |
| TC 5 | Data backup | Determine what system reacts when wrong source and destinations are given | Enter destination and source if they are wrong error message displayed. |

6.1.6.8 Confidentiality

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|----------------|---|---|
| TC 1 | authentication | Determine the protection of data from PC to connecting with phone | Start the system. It will ask the user to enter authentication code generated on mobile into text box. if it is wrong then system does not allow the user to control that phone |

6.1.6.9 Compatibility and interoperability

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

| Test Case ID | Title | Objective | Procedure |
|--------------|--------------|--|--|
| TC 1 | Windows test | To test whether or not system work on the platform keep android kit kat and change windows | Execute the software on windows 7/8 and check every functionality that the system is compatible with windows platform system works on all of these so test successful |
| TC 2 | Android test | To test whether or not system work on the platform keep windows 8 and change android | Android OS version 2.3 ,3.0, 4.0, 4.1, 4.4 system works on all of these so test successful |

6.1.6.10 Load Testing

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

The software is not a server base software so maximum no. of users that can operate it at a time is one and all the test cases described will also check for the load testing as it will be running on maximum load.

6.1.6.11 Schedule:

The system is medium and no. of test cases are acceptable so it will take one to two weeks for the acceptance testing to complete.

6.1.6.12 Human Resource:

The following testers will carry out acceptance testing for the android mobility suite

1. Isma Batool
2. Muhammad Usman
3. Muneeb Ahmed

6.2 Summary

Testing not only maintains the software and system quality but also improves over all usability and stability of the project. At different stages of development suitable testing techniques were used to ensure product worked accurately and efficiently. Almost all the errors detected during testing were removed.

6.3 Results and Analysis

6.3.1 Results and Analysis Introduction

The Android Mobility Suite has been developed to suite needs of users who spend a lot of time on the computer. It has been developed to be used on any computer even if it does not belong to you as no data is saved on client machine. The limitation of the AMS is that it has not been tested on different devices. There are thousands of type of devices available in the market and android development requires that you test your products to ensure compatibility with maximum devices.

6.3.2 Results

The major purpose of this product was enhancing usability and user experience. Visually the desktop & android application has been made user friendly with features which are easy to learn as well as easy to remember.

The product has been tested to work on Android OS version 4.0 to up to 4.0. Although version 2.3 was specified as minimum level in development, our product has not been tested on that version due to non-availability of device with that OS and as that OS version has been superseded by newer versions.

6.3.3 Analysis

Architecturally speaking the system lacks cohesion of components. This way improvements and add-on of new features can be done without any hassle.

7 Conclusion and Future Work

The purpose of this project was to learn various aspects of android and java. Implementing socket programming in order to create an interface between windows and android. With socket programming and java we controlled particular features of android phone with PC .Messaging, calls, contacts retrieval, remote camera, application management, file exploration and backup generation done successfully. Detailed knowledge of android programming and desktop programming using JavaFX was achieved.

Due to limitation of time we were able to provide only these features while some features were left. In future work we can add a lot of features to the existing system like music control, detailed file exploration and management, encrypted transmission, live streaming of camera and many other features.

Appendix A: Glossary

Activity diagram: An analysis model that shows a dynamic view of a system by depicting the flow from one activity to another. Similar to a flowchart.

Assumption: A statement that is believed to be true in the absence of proof or definitive knowledge.

Availability: Present and ready for use; at hand; accessible.

MB: megabytes

Class: A description of a set of objects having common properties and behaviors, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

Class diagram: An analysis model that shows a set of system or problem domain classes and their relationships.

API (application programming interface): [specifies](#) how some [software components](#) should interact with each other. In addition to accessing [databases](#) or [computer hardware](#), such as [hard disk drives](#) or [video cards](#), an API can be used to ease the work of programming [graphical user interface](#) components.

Wi-Fi: Wi-Fi is the name of a popular [wireless](#) networking technology that uses radio waves to provide wireless high-speed [Internet](#) and [network](#) connections

Wi-Fi Router: used to provide access to the [Internet](#).

OS-Operating System

Data flow diagram: An analysis model that depicts the processes, data collections, and flows among them that characterize the behavior of a business process or of a software system.

Dependency: A reliance that a project has on an external factor, event, or group outside its control.

External interface requirement: A description of an interface between a software system and a user, another software system, or a hardware device.

Feature: A set of logically related functional requirements that provides a capability to the user and enables the satisfaction of a business objective.

Flowchart: An analysis model that shows the processing steps and decision points in the logic of a process or of a program.

Functional requirement: A statement of a piece of required functionality or a behavior that a system will exhibit under specific conditions.

Hardware: A computer and the associated physical equipment directly involved in the performance of data-processing or communications functions.

Hardware Interface: The logical and physical characteristics of each interface between the software product and the hardware components of the system.

Implementation: execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

Interface: A point where two systems, subjects, organizations, etc., meet and interact.

Nonfunctional requirement: A description of a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behavior.

Operating Environment: The circumstances surrounding and potentially affecting something that is operating.

Operating System: A collection of software that manages computer hardware resources and provides common services for computer programs.

Perspective: The way in which objects appear to the eye.

Post condition: A condition that describes the state of a system after a use case is successfully completed.

Precondition: A Condition that must be satisfied before a use case may begin.

Procedure: A written description of a course of action to be taken to perform a given activity, describing how the activity is to be accomplished.

Process: A sequence of activities performed for a given purpose. A *process description* is a documented definition of those activities.

Quality attribute: A kind of nonfunctional requirement that describes a quality or property of a system. Examples include usability, portability, maintainability, integrity, efficiency, reliability, and robustness. Quality attribute requirements describe the extent to which a software product demonstrates desired characteristics, not what the product does.

References: List of any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

Response: A reaction, as that of an organism or a mechanism, to a specific stimulus.

Requirement: A statement of a customer need or objective, or of a condition or capability that a product must possess to satisfy such a need or objective. A property that a product must have to provide value to a stakeholder.

Requirements specification: See software requirement specification and specification, requirement.

Scope: The portion of the ultimate product vision that the current project will address. The scope draws the boundary between what's in and what's out for the project.

Software requirements specification: A collection of the functional and nonfunctional requirements for a software product.

Specification, requirements: The process of documenting a system's requirements in a structured, shareable, and manageable form. Also, the product from this process.

Stimulus: Something causing or regarded as causing a response.

Supplementary Information: Something added to complete the information.

System requirement: A top-level requirement for a product that contains multiple subsystems, which could be all-software or software and hardware.

Usability: Fit for use; convenient to use.

Use case: A description of an interaction between an actor and a system that results in an outcome that provides value to the actor.

Use case diagram: An analysis model that identifies the actors who can interact with a system to accomplish valuable goals and the various use cases that each actor will perform.

User: A customer who will interact with a system either directly or indirectly (for example, using outputs from the system but not generating those outputs personally). Also called *end user*.

User class: A group of users for a system who have similar characteristics and requirements for the system.

User Interface: *the logical characteristics of each interface between the software product and the users.*

User requirement: User goals or tasks that users must be able to perform with a system, or statements of the user's expectations of system quality.

Validation: The process of evaluating a work product to determine whether it satisfies customer requirements.

Verification: The process of evaluating a work product to determine whether it satisfies the specifications and conditions imposed on it at the beginning of the development phase during which it was created.

Vision: A long-term strategic concept of the ultimate purpose and form of a new system.

Vision and scope document: A document that presents the business requirements for a new system, including a product vision statement and a project scope description.

SMS: short message service

Encryption: hiding text by using a key so that eavesdropper hacker could not read it

PC: personal computer

Call log: display of missed, received and dialed calls

Appendix B: Bibliography

- [1]. IT103 Research Paper; The Android Operating System; “Benefits of the Android Operating System” October 2011 <http://jyangit103.blogspot.com/2011/10/benefits-of-android-operating-system.html>
- [2]. Android Developers Website “Managing The Activity Lifecycle” <http://developer.android.com/training/basics/activity-lifecycle/index.html>
- [3]. The Server Side Enterprise Java Community; “What’s the big IDE? Comparing Eclipse and NetBeans” <http://www.theserverside.com/feature/Whats-the-Big-IDE-Comparing-Eclipse-vs-NetBeans>
- [4]. Eastern Michigan University; “Development Techniques for Android Platform and Mobile Device Application” January 2012 <http://commons.emich.edu/cgi/viewcontent.cgi?article=1762&context=theses>
- [5]. Android Developers Website ; “Application Fundamentals” <http://developer.android.com/guide/components/fundamentals.html>
- [6]. Javaworld Developers Website ; “Socket Programming in Java: A Tutorial ” ; Dec 1996 <http://www.javaworld.com/jw-12-1996/jw-12-sockets.html?page=1>
- [7]. AndroidHive ; “Android Working with Android API” ; Sept. 2013 <http://www.androidhive.info/2013/09/android-working-with-camera-api/>
- [8]. Vogella Tutorials ; “Android Animations” <http://www.vogella.com/articles/AndroidAnimation/article.html>
- [9]. Vogella Tutorials; “Android Logging ” <http://www.vogella.com/articles/AndroidLogging/article.html>
- [10]. Vogella Tutorials; “Android Custom Views” <http://www.vogella.com/articles/AndroidCustomViews/article.html>

- [11]. Vogella Tutorials ; “Android Drag and Drop”
<http://www.vogella.com/articles/AndroidDragAndDrop/article.html>
- [12].Vogella Tutorials ; “Android ListViews ”
<http://www.vogella.com/articles/AndroidListView/article.html>
- [13].Tutorials Point Blog ; “Java Multithreading”
http://www.tutorialspoint.com/java/java_multithreading.htm

Appendix C: User Manual



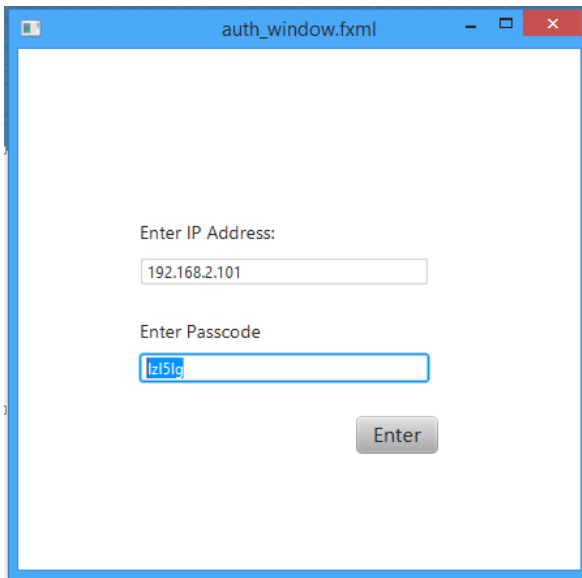
This document will provide all the necessary information for running the system effectively.

1. Authentication

After running both the android and PC app, the first step is authentication. It is necessary because without it connection will not be established and you will not be able to perform any tasks. After running the android app, it shows an IP address and a passcode



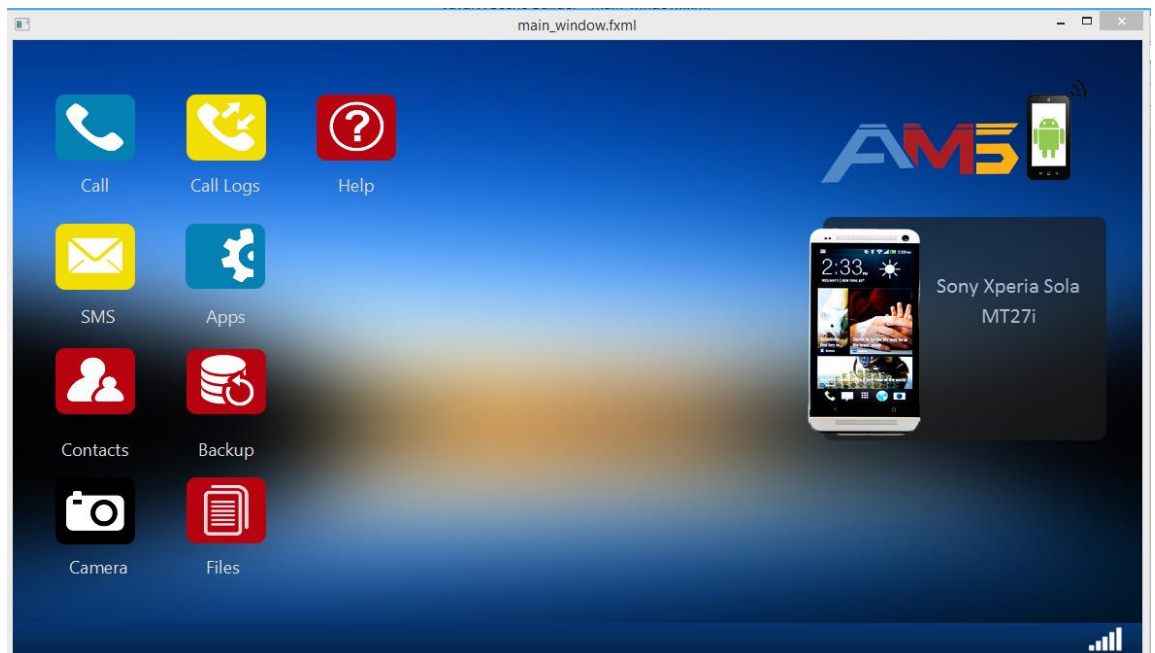
You need to enter the IP address and the passcode on the PC application.



Press enter. After pressing enter if the authentication is successful the main window will appear.

2. Main Window

The main window allows the user to perform all the functionality of the system.

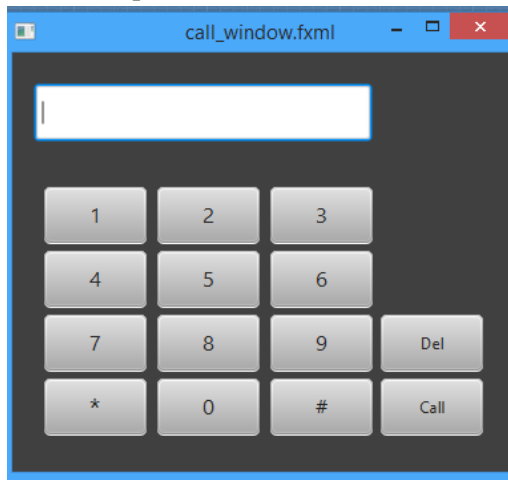


All the features are provided here.

On the left are all the buttons. Each button constitutes a functionality that is being provided. On the right is the phone model name and on the bottom left are signal strength, battery and Wi-Fi signal strength.

3. Call

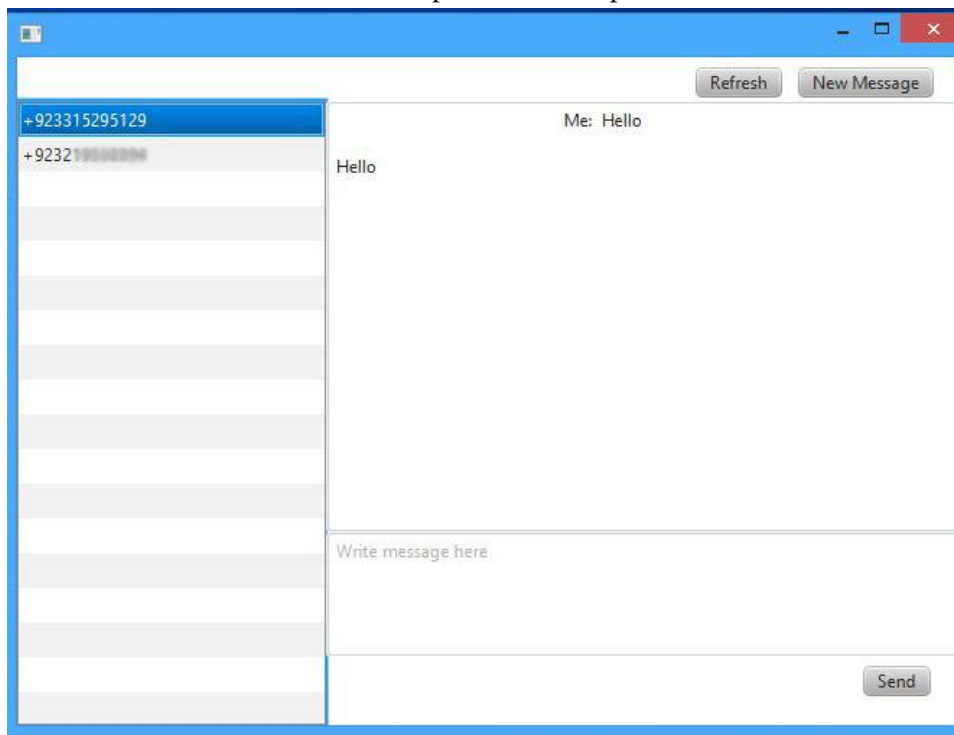
In order to initiate a call from PC press the call button and a new window will appear.



Enter the number and press the call button. A call will initiate on the mobile phone. You now need to switch to the phone in order to take the call.

4. SMS

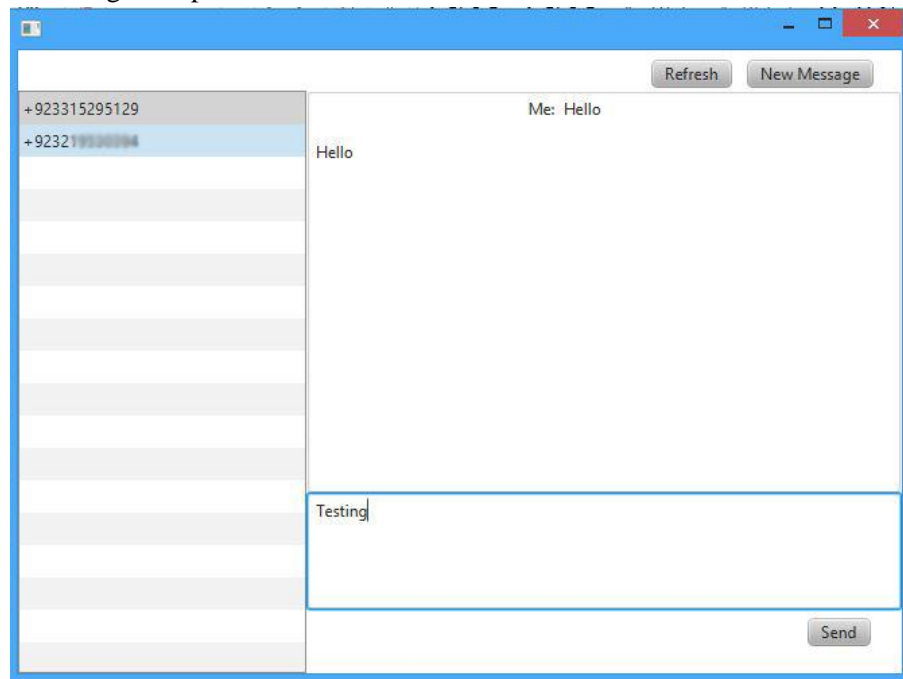
You can send and receive SMS. AMS provides a complete conversation view of SMS.



On the left is the list of all conversations. If you click on an item in the list it will show all the messages on the conversation view in the right.

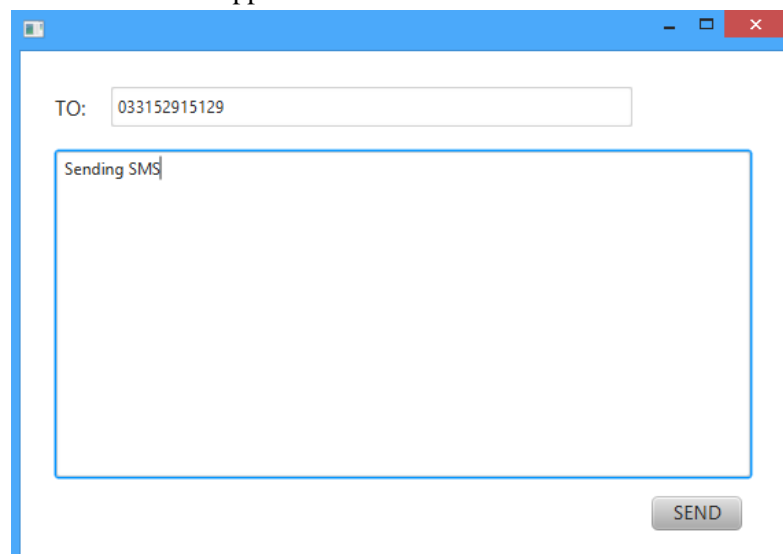
Send SMS

If you want to send an SMS to a person in the conversation list e.g. the one selected. Just type the message and press Send.



The SMS will be sent.

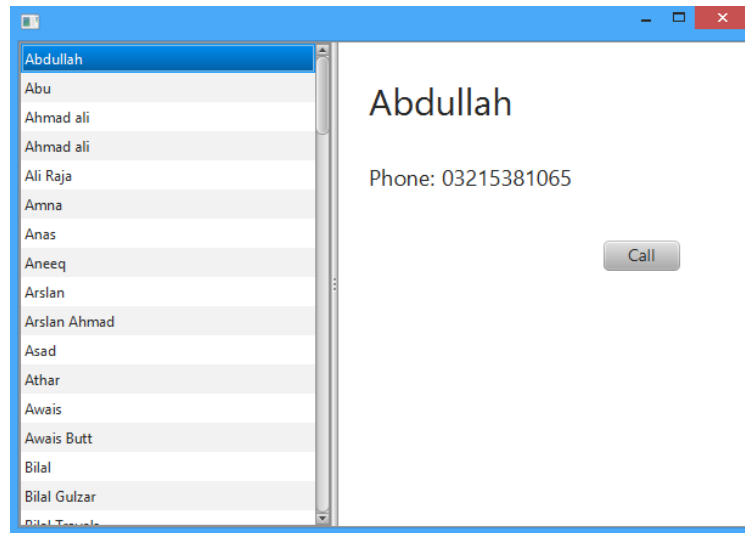
If you want to send SMS to a new number. Press the New Message button on the top right and a new window will appear.



Fill the required fields and press Send button. The SMS will be sent

5. Viewing Contacts

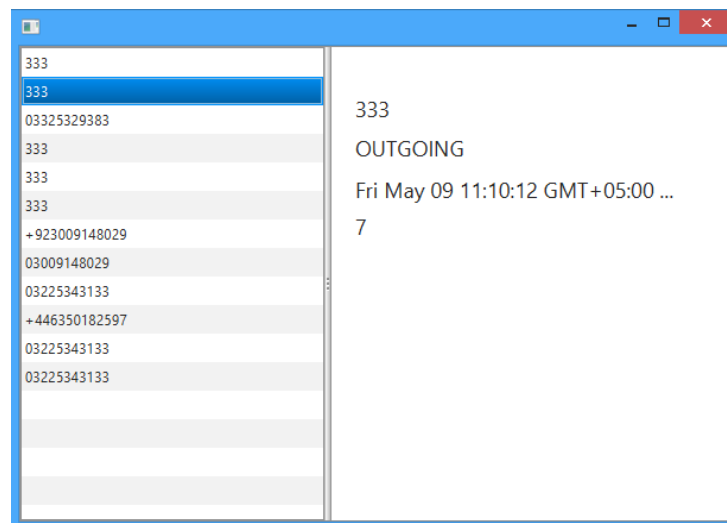
Press the Contacts button on the main window. A new window will appear



Select the required contact by clicking on the name. Once clicked you will get the information about the contact.

6. Viewing Call Logs

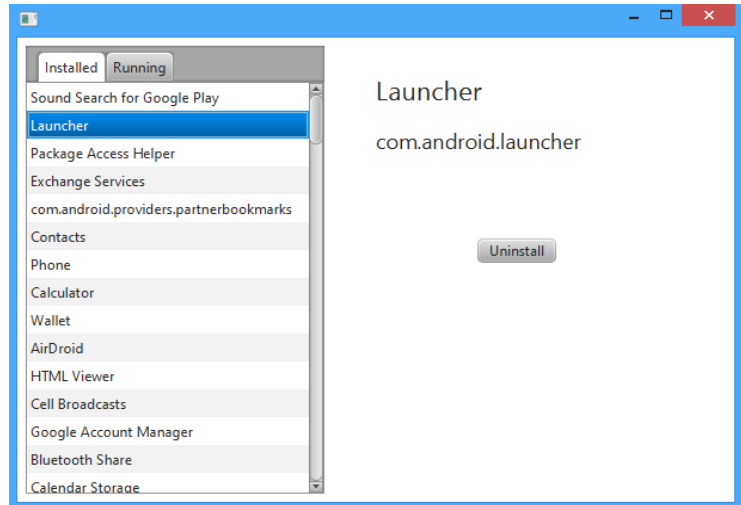
Press the Call Logs button on the main window. A new window will appear.



The window will show all the call logs. Clicking on an item in the list will provide the necessary details regarding the call.

7. Viewing Applications

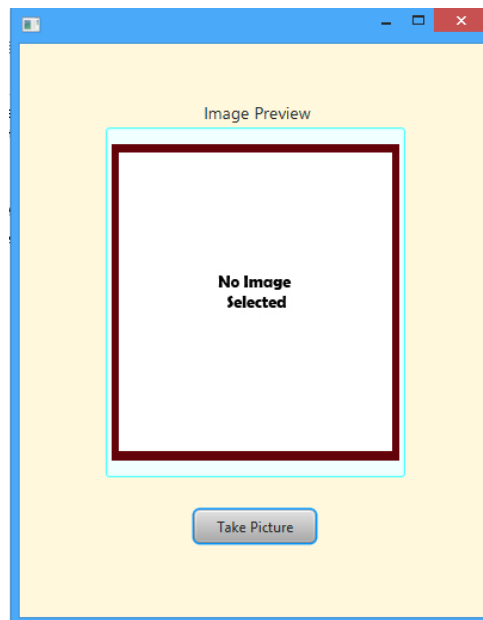
Click on the Apps button in the main window. A new window will appear



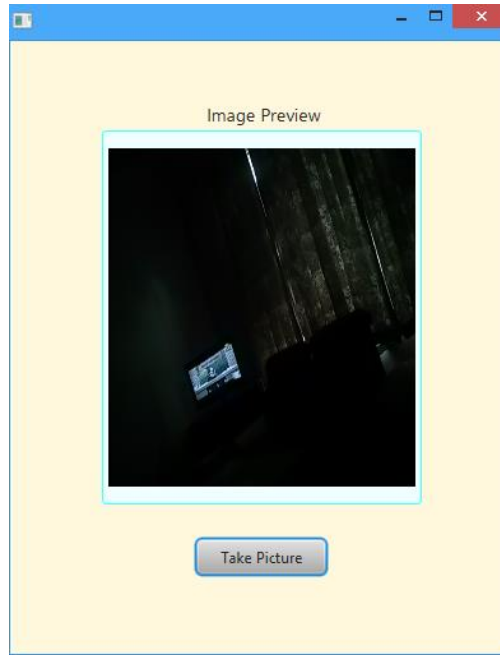
It shows the installed and running apps. Click on the tab to view the corresponding apps. Clicking on the name will provide the details of the application. You can uninstall an installed application by clicking the Uninstall button.

8. Remote Camera

Click on the Camera button in the main window. A new Window will appear.

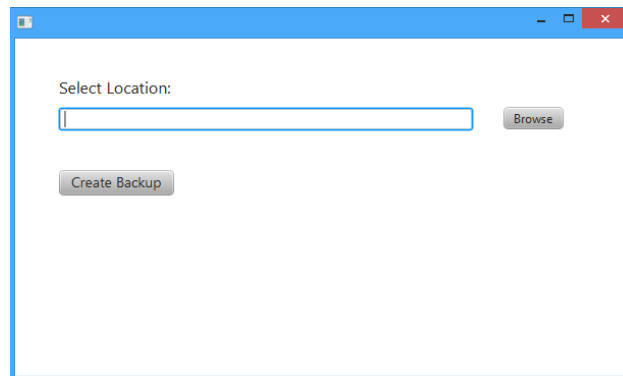


Click the Take Picture button. The mobile camera will take a picture and display it here in the preview.

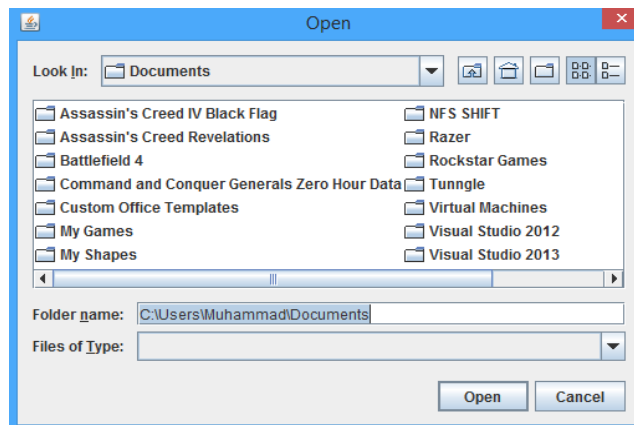


9. Backup

Press the Backup button on main window. A new window will appear



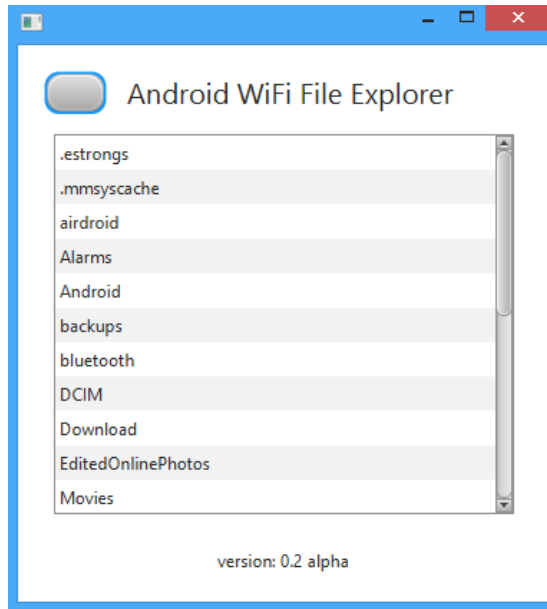
Select the location where you want to create backup using Browse button.



Press create Backup and backup will be created at the specified location. Backup will be of contacts and SMS.

10. File Exploration

Click on the file button. A new window will appear



Double click on an item to go inside that folder. If you want to come back. Press the back button on top left.