College of IT & Computer Engineering

Android Remote PC Controller

Prepared By:

Al-Montaser Noman Al-Mohtaseb.

Project Supervisor:

Dr. Radwan Tahboub.

قدم هذا البحث لإنـهاء متطلبات التخرج لتخصص علم الحاسوب.

2014/2015

Acknowledgement

This graduation project couldn’t be done without many efforts, and many people’s support, so I want to thank all of my teachers who taught me in Palestine Polytechnic University in the computer science major in this period that changed a lot of things in my academic and professional life. I would to thank especially the project supervisor Dr. Radwan Tahboub for his supervision efforts on this work, and the graduation project’s coordinator Ms. Ezdehar jawabreh for her efforts.

I wish that this work could be good enough to prove what I leaned in my university, and I will do my best to reach the people who will be benefited from it as a real life project, to make their life’s more comfortable and easier while using this technology.

Abstract

The aim of this project is to build a software for the smart phones that can control the PCs, and keeps a lot of efforts and resources in using them. And this will be done by providing full access to the main resources of the PCs; the files, and the processes with full management capability, and by enabling the users to use their smart phones as a mouse or keyboard, and to control their computer features and the computer power modes.

The targeted devices in this project is the smart phones that has the android OS, and the PCs that running the Windows OS. This project mainly developed using java programming language in the android devices software part, and C# programming language in the PC software part using object oriented paradigm.

ملخص

يهدف هذا المشروع لبناء تطبيق للأجهزة الذكية يتيح للمستخدم التحكم بجهاز الكمبيوتر عن طريف جهازه الذكي بحيث يتيح له امكانيات كاملة لتنظيم ملفاته و البرامج و المهام التي تعمل على الجهاز و استخدام التطبيق كوسيلة ادخال للوحة التحكم او الفأرة, و امكانية رفع الصوت او الإضائة و التحكم في اطفاء او قفل او إعادة تشغيل الجهاز من خلال خيارات الطاقة.

التطبيق يستهدف الأجهزة التي تعمل على نظام اندرويد و الأجهزة الشخصية بنظام ويندوز, تم تطوير هذا التطبيق بإستخدام لغة البرمجة "سي شارب" على الويندوز و لغة البرمجة "جافا" على الأندرويد بإستخدام البرمجة كائنية التوجه.

Table of Contents

Chapter 1: Introduction.

Chapter 2: Software Project Management Plan.

Chapter 3: Software Requirements Specifications.

Chapter 4: Design.

Chapter 5: Implementation.

Chapter 6: Testing.

List of Tables

Chapter 1

Chapter 3

Chapter 3

Chapter 1   Introduction

1.1 Overview.

1.2 Project Idea Description.

1.3 Motivation.

1.4 Project Scope.

1.5 Description of later parts.

1.1 Overview.

This chapter is the graduation project document, and will talk about the project idea and description, the motivations, the scope of the project and brief description for later parts.

1.2 Project Idea and Description.

The purpose of this project is to build a software that enables it's user to control his PC or Laptop remotely from his smart phone, and to access its resources. The software also will enable the user to use the smart phone instead of a lot of input devices. The application will provide the capability to access the remote files to manage them easily. So the user can create new files, or can open, delete or rename existing ones, and transferring the files if necessary.

The user can manage the processes, so stopping a process or running an application will be possible without touching the computer or being physically exist in the same place. And that will be provided by the application that any user will be pleased to have in his smart phone. Another feature will be provided is using the smart phone as an input device; for example, the user can use the phone as a keyboard or as a mouse. So the smart phone will be treated as mouse, or remote keyboard in comfortable and interesting way.

In addition the user can control his PC features; increasing the brightness or decreasing the sound volume. The power options will be provided too, so the user can logout, shutdown, sleep or restart his computer.

This application can be used by one user at a time with encrypted connection.

1.3 Motivations.

The motivations to build this project are:

a. The application will save many efforts and resources, so you can control your PC even if your mouse or keyboard has been broken.
b. The user can easily access his computer by the smart phone, and won’t be tied to the existing input technology.
c. Transferring the files using TCP connection will be provided which is faster than the Bluetooth or the USB, so you don’t need to use the cable or the Bluetooth to transfer the files anymore.

1.4 Project Scope.

1.4.1 Project Requirements.

1. Allowing the user to use his smart phone as a mouse, so when he moves his figure on the touch screen the mouse will move on the computer. The tabs will be treated as clicks, and the long presses as a left button mouse click, the mouse either will has a virtual wheel to control the mouse wheel to scroll down or up.

2. Allowing the user to use his keyboard as a remote keyboard in a clear way, so when the user uses his smart phone touch screen as a keyboard, he will see what is written on it until he press “Enter”, and the input will be transferred to the PC in real time.

3. Allowing the user to control and transfer his files, so the user can explore, rename, delete, or open a file. The user can also transfer his files from his PC to the smart phone and vice versa.

4. Allowing the user to control his processes, so the user can see the running processes and terminate any of them, or seeing the processes information.

5. Controlling the PC features. The user can increase the sound volume, or decrease it, and can increase the brightness volume or decrease it.

6. PC power options will be provided, so the user will be able to shutdown, restart, sleep, or restart his PC remotely.

7. The application on the PC will be listening for the incoming connections, and will provide an id as a unique key to allow the user to connect from his smart phone.

8. The application in the PC will manage the files and the processes by the remote commands.

1.4.2 Project Context.

This project aims to build a smart phone application that enables the user to control his PC remotely, and will be built as two parts, client side part (the smart phone), and the server side part (the PC), and the project will be done within two months.

The Input, process and output will be as the following:

1. The input from the user will be the commands from the files manager, or the processes manager, and from the touch screen (the mouse moves or keyboard keystrokes).

2. The process in the application will be by encapsulating the commands in appropriate way and sending them to the PC.

3. The output will be the event that will be executed in real-time.

1.4.3 Project Boundaries.

The boundaries for this project:

This application as mentioned is created to control the PC and to manage its resources in a local network so it will make your job easier in controlling your PC, so whether you are a student, dentist or lecturer or whatever you are, you won’t refuse to use this application, because it provides many features and capabilities that any one will need in any age and in any place, and it will save a lot of time and efforts in interesting way. This will give you a chance to use the computer in a comfortable way.

1.4.4. Project Deliverables.

The deliverables of this project is all phases of the software development process, starting with the proposal, and delivering the introduction that has the idea and the requirements, to the planning, analysis, design, and the construction of the code, and finally testing its components.

Here are the deliverables:

1. Project Proposal.
2. Project Introduction.
3. Planning Management Plan.
4. System Analysis Document.
5. Design Documents.
6. Implementation.
7. Test documentation.

3. Project Constraints and Assumptions.

This project will be implemented as 2 parts:

a. An android application that the user will use, and will enable him to connect to his PC, and to use the application advantages, and will be made by Java and using the android ADT (Android Developer Tools).

b. The server part that will access all of the PC and system resources and will control it, and will be made by C# programming language.

4. Schedule for milestones.

Milestone	Date
Project Proposal	6-Sep-2014
Introduction	10-Sep-2014
Literature	15-Sep-2014
Project Management Plan	22-Sep-2014
Software Requirement Specifications	27-Sep-2014
Software Design Description	30-Sep-2014
Implementation	5-Dec-2014
Software Testing Documentation	7-Dec-2014

Table (1.1) schedule for milestones.

1.5 Description for Later Parts.

Planning and analysis

The planning and the analysis phase will contain the task set for the project, and the risk analysis with its mitigations and solutions, the project software and hardware resources, and the human resources efforts, the activity network, and finally the time table.

Design

Designing the UML diagrams for the system, UML class diagrams, activity diagrams, sequence diagrams and the object relational diagrams, and the user interfaces design.

Implementation

The system will be implemented and divided into two parts:

1. The android application; the client side that will be existed on the smart phone.
2. The PC application; the server side that will handle the smart phone events.

Testing:

This phase contains the unit testing documentation and the integration testing for the software parts.

Chapter 2 Software Project Management Plan (SPMP)

2.1 General Overview

2.2 Project Management Plan

2.3 Task Set

2.4 Risks (using Risk Information Sheet (RIS))

2.5 Project Resources and Total Estimated Cost

2.6 Activity Network

2.7 Timetable (Gantt chart)

2.1 Overview.

This chapter is the project management plan part of the introduction to graduation project, and will propose the project management plan, the task set, risks and the project resources and estimated cost, and will propose the project methodologies and components, and lastly the conclusion.

2.2 Project Management plan.

The following sub-sections will propose the task set, risks analysis and project resources.

2.2.1 Task Set.

In this section the task set will be defined, for all of the project phases, from the requirements collection and analysis, through the literature review, planning, analysis, construction and finally testing.

1. Requirements collection and analysis.

The requirements connection and analysis is will be as the following:

1.1 Identifying the requirements.

Identifying the needs of the project, the project idea and description, the project requirements and the technical issues.

1.2 Defining the project output and scope.

Defining the application input, processing and output and the project context, defining the project boundaries and requirements.

1.3 Defining the functionalities and behaviors of the system.

Defining the functionalities for the applications and the importance of the project.

2. Preliminary concept planning.

2.1 Defining the task set and refining each task.

In this section the task set are defined to show in clear way what is the project phases and the tasks that needs to be done.

2.2 Drawing the activity network for the task set.

The activity network shows how the task set will be applied sequentially using the activity network diagram.

2.3 Estimating the total cost for the hardware, the software and the human resource efforts.

2.4 Constructing the time table (Gantt chart) for the main tasks.

2.5. Evaluating the risks of the project.

In this section we evaluate the potential risks using the risk information sheet, and we find the solutions and the mitigations for it.

3 Modeling.

The modeling for this project will contain two phases, the analysis and the design:

3.1 Analysis.

3.1.1 Defining the actors and the users in the system, and their rules.

In this section the actors will be defined with their rules, so it will be clear how the user will use this application.

3.1.2 Preparing Requirement Description.

In this section the requirements will be shown as scenarios and use cases.

3.1.3 Preparing CRC Modeling.

The CRC shows the potential classes that will be used in the development process, with its potential operations and attributes.

4.1.4 Preparing Classes analysis.

a. Responsibilities.

b. Collaborations.

6. Designing Class Hierarchies and Relationships.

3.2 Design.

3.21 Designing Objects-Relational Model.

3.2.2 Designing State Behavioral Modeling.

3.2.3 Subsystem Design.

3.2.4 Class and Object Design.

3.2.5 Interface (Message) Design.

3.2.6 Designing Object Interfacing.

3.2.7 User Interface Design.

4. Construction.

4.1. Constructing the android application using java programming language based on the design that will be in object oriented paradigm.

4.2. Constructing PC application using C# language based on the design that will be in object oriented paradigm.

5. Testing.

5.1 Testing the classes (Unit Testing).

2.2 Integration Testing.

3.2.2 Risks.

Some of the risks may be faced in the future during the application use or during the project deployment process, so here are some risks with its mitigations.

1. One of the main risks that we may face in the project is the speed of transferring the commands in the network, since this application sends a lot of information to insure the quality of the application control, so the application will implemented in real time and continuously and will be in very efficient way to transfer the data.

Risk Information Sheet			
Risk ID:01	Date: 15-Sep-2014	Probability: 90%	Impact : high
Description : The application may hang and the commands may arrive late so this may affect the whole performance of the application.			
Refinement: The commands needs to be executed at run-time and in a reliable way, so losing the information will be annoying to the user.			
Mitigation: Making the application as a real time application and producing efficient way to send the information.			

Table (3.1) Risk Information Sheet 1.

5 Another risk may appear is the fail in the application especially that the application uses the network, and the system resources, and the APIs in windows may fail any time because it deals with system input traffic, so the solution is to test it very well and handling all of the possible exceptions mentioned in the APIs documentations.

Risk Information Sheet			
Risk ID:02	Date: 15-Sep-2014	Probability: 80%	Impact : high
Description : The application may fail to execute a command or task, because the OS operations always expected to fail and they have to be handled in a general form.			
Refinement: The application may fail to execute a command, like accessing a file without an appropriate permission.			
Mitigation: Testing the components and the units and identifying the fail scenarios and handling them.			

Table (3.2) Risk Information Sheet 2.

3. The last risk may appear in the development process in the feature, so if we decided to develop this application for different screen sizes, for instance for a tablet, the user interfaces will differ in the size, so its mitigation is to create the user interfaces dynamic and relative, and lay outing them very well.

Risk Information Sheet			
Risk ID:03	Date: 15-Sep-2014	Probability: 80%	Impact : high
Description : Deployment problem in the screen sizes for the android smart phones.			
Refinement: The application will appear properly for the deployed screen size, but may differ in different screens.			
Mitigation: Making the application interfaces responsive and easy to be edited for feature sizes.			

Table (3.3) Risk Information Sheet 3.

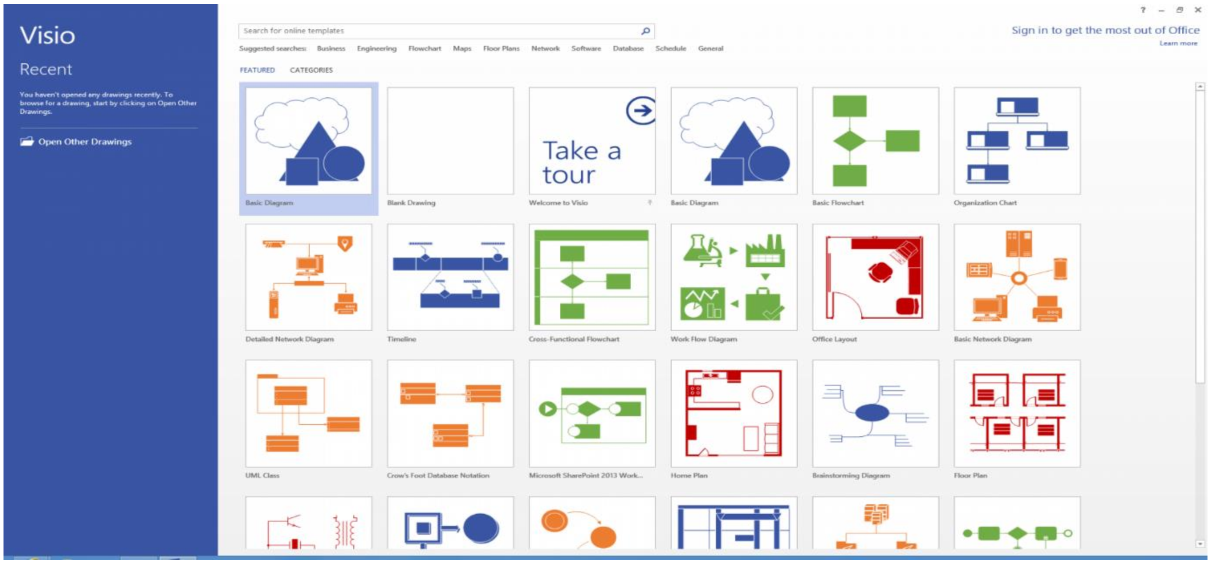
2.2.3 Project resources and total estimated cost.

1. Software Resources.

The software resources that will be used in the development process to develop the applications and to design the components, are the following:

A. Microsoft Visio 2013.

This application will be used to design the UML diagrams for the applications (The smart phone applications, and the computer application), so most of the diagrams (UML Class Diagrams, Activity Diagrams, Sequence Diagrams, Object Relational Diagrams, and the Class Hierarchal Relationships) will be designed by Microsoft Visio 2013.

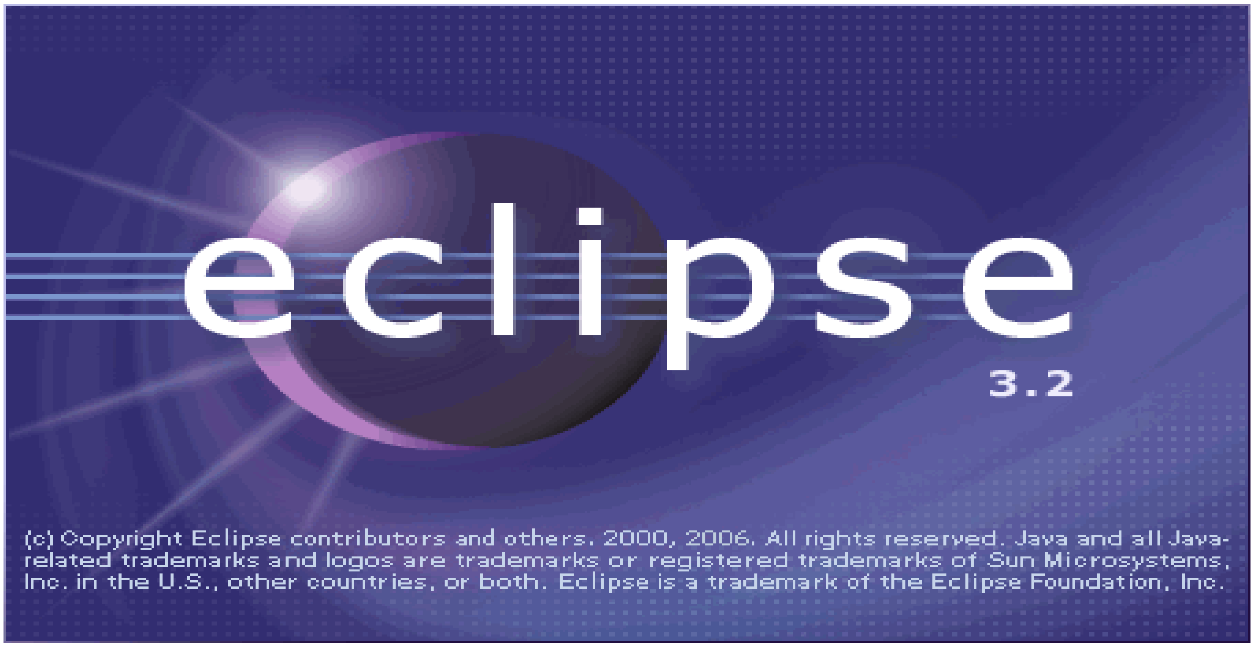


You haven't opened any drawings recently. To browse for a drawing, start by clicking on Open Other Drawings.

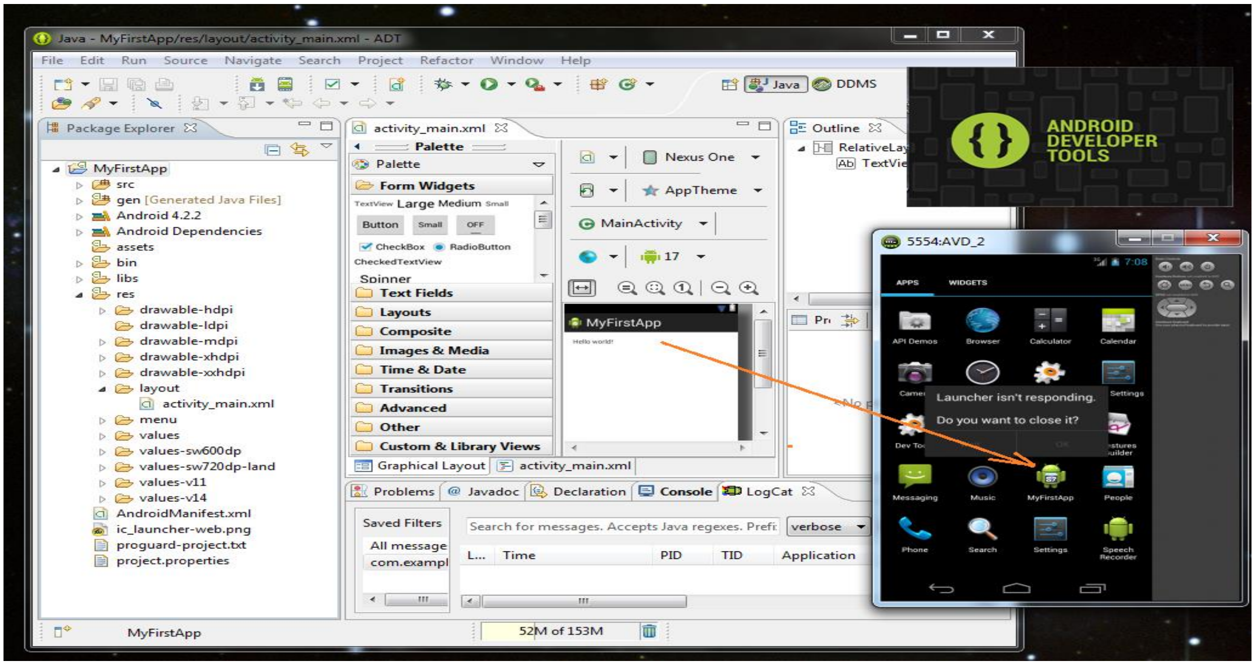
The software is free for the students.

B. Eclipse IDE.

This IDE is an open source and known application to develop applications in different languages such as (Java, PHP, C/C++, Python), and can be integrated to develop android applications after installing the ADT (Android Development Tools) from android developers website.



(c) Copyright Eclipse contributors and others. 2000, 2006. All rights reserved. Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both. Eclipse is a trademark of the Eclipse Foundation, Inc.



Launcher isn't responding.

Do you want to close it?

This IDE is totally open source and free.

C. Microsoft Visual Studio 2013.

Microsoft visual studio is an IDE and Compiler, for different languages, especially the languages that provided from Microsoft, like C#,F# and others, This IDE will used to develop the application on the PC in C# language.

This software is free for the students.

2. Hardware resources.

The hardware resources that will be used in this project during the development process is:

A. Personal Computer.

Personal Computer to prepare the documentation and the design for the project, and to develop the project software components, that will be tested on it, the components that will be developed for the smart phones.

The total cost for using the PC during the semester nearly 200$.

B. Smart Phone with Android OS.

Smart phone with android OS will be used during the development and the testing process to simulate and test the application and insure that it works properly.

The total cost of the device (Galaxy S Duos) 300$.

The estimated cost for the software, hardware and the human resources in the following table.

Resource	Cost
Human Resource	1000$*4Months= 4000$
Hardware Personal Computer Android Device	300$ + 200$ =500$
Software MS Visio Free MS Visual Studio Eclipse	Free
Total	4500$

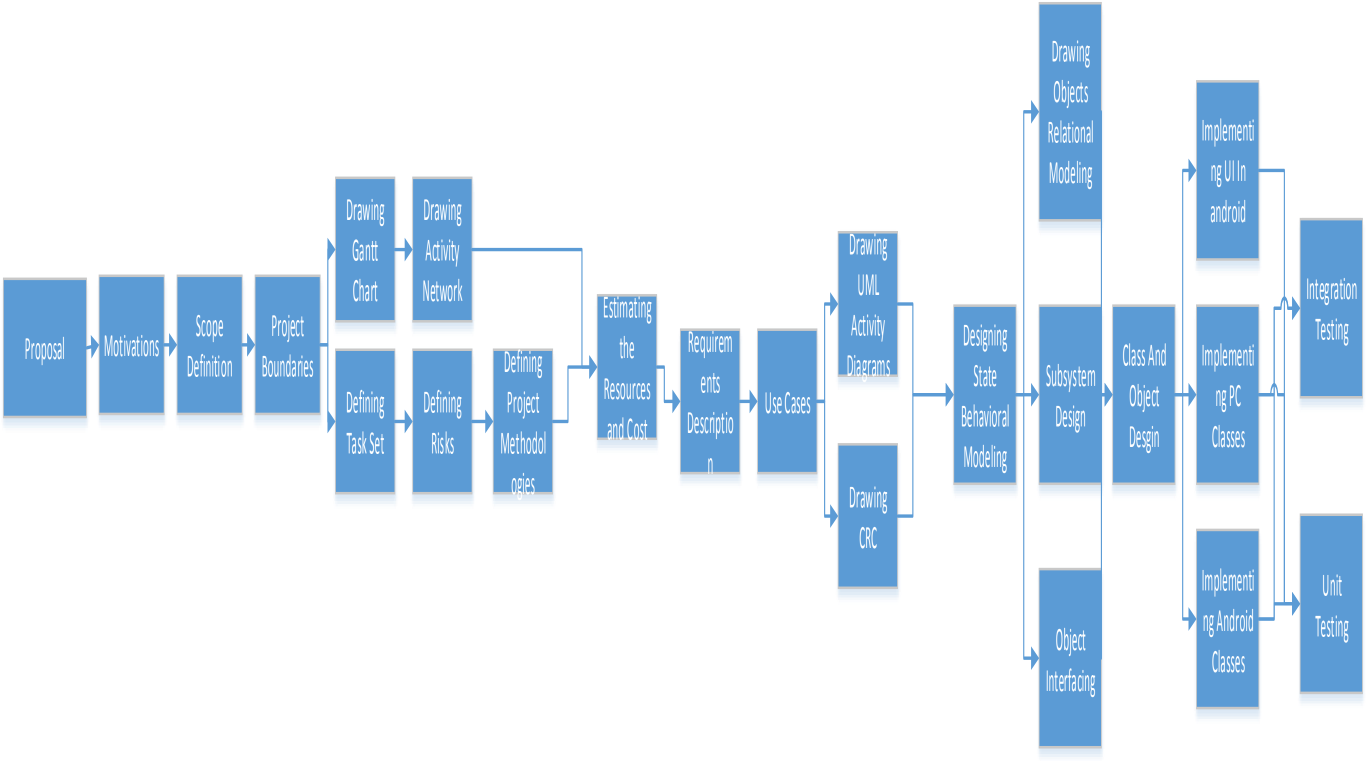
Table (3.4) Project resources estimated cost.

6 Feasibility study.

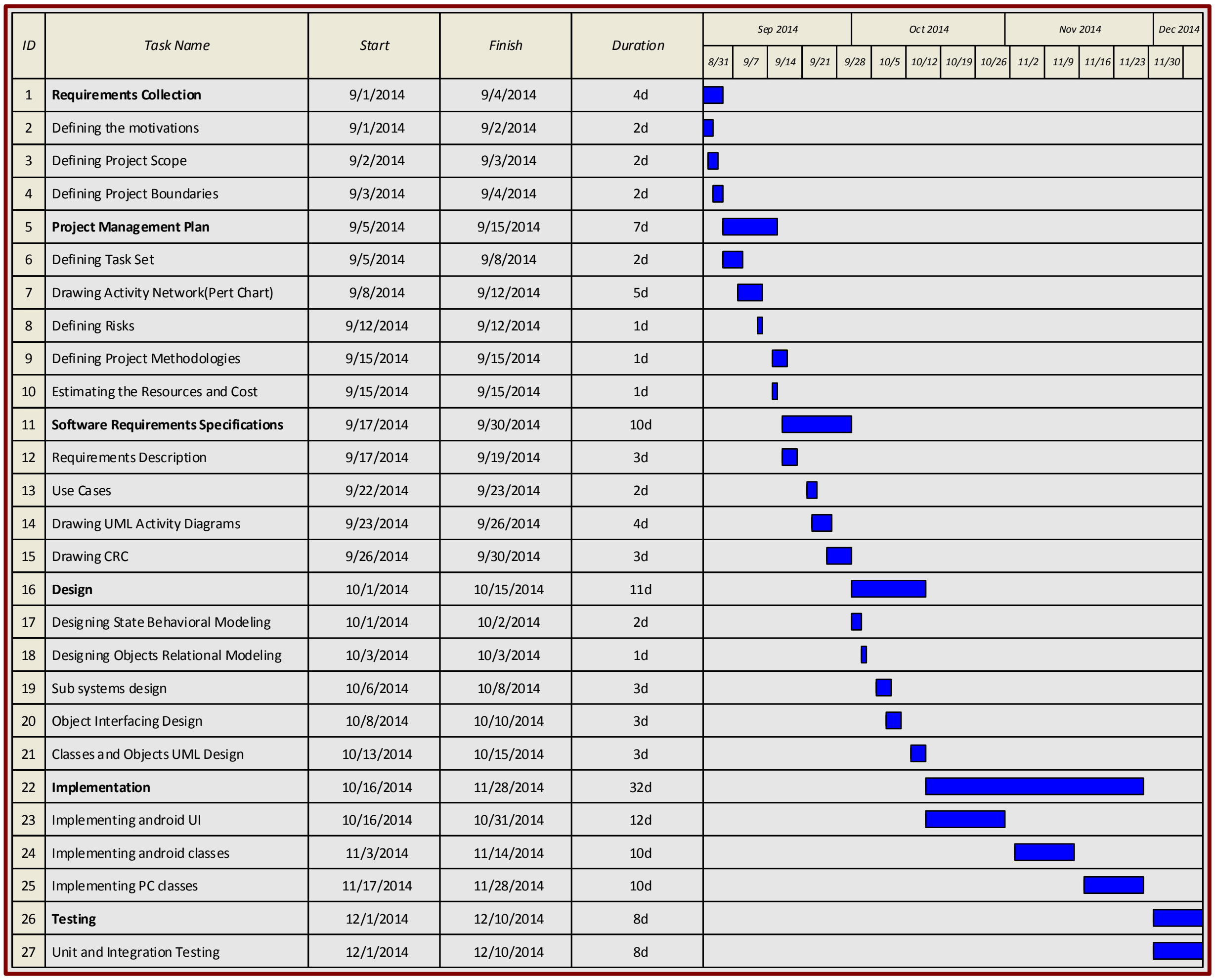
For this application each copy of it can be sold by nearly 2$, if we spent 1000$ to advertise it during 6 months, it may achieve 3000 copies to be downloaded.

This means the project total income will be 6000$ - 1000$ = 5000$.

2.2.4 Activity Network.



2.2.5 Gantt Chart.



ID	Task Name	Start	Finish	Duration
1	**Requirements Collection**	9/1/2014	9/4/2014	4d
2	Defining the motivations	9/1/2014	9/2/2014	2d
3	Defining Project Scope	9/2/2014	9/3/2014	2d
4	Defining Project Boundaries	9/3/2014	9/4/2014	2d
5	**Project Management Plan**	9/5/2014	9/15/2014	7d
6	Defining Task Set	9/5/2014	9/8/2014	2d
7	Drawing Activity Network(Pert Chart)	9/8/2014	9/12/2014	5d
8	Defining Risks	9/12/2014	9/12/2014	1d
9	Defining Project Methodologies	9/15/2014	9/15/2014	1d
10	Estimating the Resources and Cost	9/15/2014	9/15/2014	1d
11	**Software Requirements Specifications**	9/17/2014	9/30/2014	10d
12	Requirements Description	9/17/2014	9/19/2014	3d
13	Use Cases	9/22/2014	9/23/2014	2d
14	Drawing UML Activity Diagrams	9/23/2014	9/26/2014	4d
15	Drawing CRC	9/26/2014	9/30/2014	3d
16	**Design**	10/1/2014	10/15/2014	11d
17	Designing State Behavioral Modeling	10/1/2014	10/2/2014	2d
18	Designing Objects Relational Modeling	10/3/2014	10/3/2014	1d
19	Sub systems design	10/6/2014	10/8/2014	3d
20	Object Interfacing Design	10/8/2014	10/10/2014	3d
21	Classes and Objects UML Design	10/13/2014	10/15/2014	3d
22	**Implementation**	10/16/2014	11/28/2014	32d
23	Implementing android UI	10/16/2014	10/31/2014	12d
24	Implementing android classes	11/3/2014	11/14/2014	10d
25	Implementing PC classes	11/17/2014	11/28/2014	10d
26	**Testing**	12/1/2014	12/10/2014	8d
27	Unit and Integration Testing	12/1/2014	12/10/2014	8d

2.3 Project Methodology.

3.3.1 Options.

The smart phone application will be implemented for android and using Java programming language and using Object Oriented.

The PC application will be implemented using C# using Object Oriented.

3.3.2 Analysis of Options.

The most appropriate programming language for the android os is Java, and the Object Oriented paradigm will be used to hide the complexity of the application and to make it open for future enhancements.

The PC application will be implemented using c#, because it has the capabilities to access the system resources efficiently. The paradigm that will be used is Object Oriented to try to hide the complexity of dealing with the system resources.

2.4 Project Components.

Software Components.

The Software will consist as mentioned 2 Components, an application for android the client and the server the PC application.

Chapter 3 Software Requirement Specification (SRS)

3.1 General Overview

3.2 Requirement Description

3.3 CRC (Class Responsibilities Collaborator) Modeling

4.3. N Class-n

4.3. N.1 Responsibilities

4.3. N.2 Collaborations

3.4 Class Hierarchies and Relationships

3.1 Overview.

This chapter is the software requirements specification part of the introduction to graduation project, and will propose the actors and use case diagrams and the use cases descriptions, CRC and class hierarchies and relationship, and in the last the UML activity diagram.

3.2 Requirements Description.

A. Actors.

The actors in the system is:

1. The user: is any person who uses this application.
2. The application: the server side that handle the smart phone requests and listens for his incoming connections.

B. Use Cases Description.

In the smart phone and the PC application.

1. Connect use case and scenario shows how the user connects to his PC from the smart phone to start controlling it.

Use Case:	Connect to PC.
Primary Actor:	User.
Goal in Context:	To enable the user to connect to his PC.
Precondition:	The application is connected to the network.
Trigger:	The user decides to connect to his PC to start control it and access its resources.
Scenario:	1. The user clicks on the connect button after insert the IP. 2. If the connection permitted the application will show the control panel. 5. If the connection denied the application will show an error message to help the user to connect correctly.
Exception:	1. The application is not connected to the network.
Priority:	Essential.

Table (4.1) 1. Connect Use Case.

2. The use case of using the smart phone as a keyboard, when the user decides to use his smart phone as an input device to type something on the computer, without using the traditional keyboard.

Use Case:	Use remote keyboard.
Primary Actor:	User.
Goal in Context:	To enable the user to use the keyboard on his smart phone screen.
Precondition:	The application is connected to a PC.
Trigger:	The user decides to use his smart phone as a keyboard.
Scenario:	1. The user clicks on the keyboard button. 2. The application will show a full keyboard. 3. The user touches any button (keystroke), the application will send the keystroke to the PC to be pressed. 4. The application will show the effects on the screen.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.2) Use Remote Keyboard Use Case.

3. Use remote mouse use case, shows how the scenario will be when the use decides to use his smart phone as a remote mouse to control his PC without touching he traditional mouse.

Use Case:	Use remote mouse.
Primary Actor:	User.
Goal in Context:	To enable the user to use the mouse on his smart phone screen.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the network.
Trigger:	The user decides to use his smart phone as a mouse.
Scenario:	1. The user clicks on the mouse button. 2. The application will show a mouse on the touch screen (an area to move your figure on, with two buttons and wheel). 3. The user moves his finger on the touch screen. 4. The application will calculate the movement value and sends it to the PC and executed.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.3) Use Remote Mouse Use Case.

4. This use case shows how the user can use the mouse buttons using his touch screen on the smart phone.

Use Case:	Tab right or left buttons in the mouse.
Primary Actor:	User.
Goal in Context:	To enable the user to click the right or left buttons on the PC mouse.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the network.
Trigger:	The user decides to click right or left button in the mouse.
Scenario:	1. The user tabs on the mouse button. 2. The application will show a mouse on the touch screen (an area to move your figure on, with two buttons and wheel). 3. The user tabs on the right or left mouse button. 4. The application will send the command for the PC to be executed.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.4) Use Remote Mouse Buttons Use Case.

5. This use case shows how the user can use the smart phone touch screen as a wheel, for example to scroll the mouse wheel on his PC.

Use Case:	Scroll down or up the wheel in the smartphone mouse.
Primary Actor:	User.
Goal in Context:	To enable the user to use the mouse wheel on his smart phone screen.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the network.
Trigger:	The user decides to use his smart phone mouse wheel.
Scenario:	1. The user clicks on the mouse button. 2. The application will show a mouse on the touch screen (an area to move your figure on, with two buttons and wheel). 3. The user moves his figure on the touch screen. 4. The application will send the value to the PC to be executed.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.5) Use Remote Mouse Wheel Use Case.

6. Exploring files use case, showing how the user can explore his files and access them.

Use Case:	Explore files.
Primary Actor:	User.
Goal in Context:	To enable the user to explore the files on his PC.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the network.
Trigger:	The user decides to explore the files on his PC.
Scenario:	1. The user clicks on the files button. 2. The user selects the file path. 2. The application will send a request to receive the current file location contents. 3. The application receives the files and folders and shows them.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.6) Exploring Files Use Case.

7. Delete file use case showing how the user can delete his file, and what is the scenario for doing that.

Use Case:	Delete File.
Primary Actor:	User.
Goal in Context:	To enable the user to delete a file on his PC.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the network.
Trigger:	The user decides to use his smart phone as a mouse.
Scenario:	1. The user clicks on files. 2. The user explores the files. 3. The user selects delete file option. 4. The application asks the user to confirm the operation. 5. The application sends the delete file command to be executed on the PC.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.7) Delete File Use Case.

8. Open file use case, showing how the user can open a file, while exploring the existing files.

Use Case:	Open File.
Primary Actor:	User.
Goal in Context:	To enable the user to open a file.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to open a file whatever it is.
Scenario:	1. The user clicks on the files button. 2. After the user explores the files the user selects Open. 3. The application then will send a command to run that file or to open it.
Exception:	1. The application is not connected to the internet. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.8) Open File Use Case.

9. Rename file use case, shows how the user rename a file.

Use Case:	Rename File.
Primary Actor:	User.
Goal in Context:	To enable the user to rename a file.
Precondition:	1. The application is connected to a PC.
Trigger:	The user decides to open a file whatever it is.
Scenario:	1. The user clicks on the files button. 2. After the user explores the files the user selects rename to a specific file. 3. The application then will show an input box to enter the name of the file. 4. The application sends a command with the file name to rename that file. 5. The file name updates in the list.
Exception:	1. The application is not connected to a PC.

Table (4.9) Rename File Use Case.

10. Explore processes use case, shows how this feature implemented in a simplified way.

Use Case:	Explore Processes.
Primary Actor:	User.
Goal in Context:	To enable the user explore his processes.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to explore his processes.
Scenario:	1. The user clicks on task manager. 2. The application will send a command to receive the processes information. 3. After the application receives the information will show it, to enable the user to manage them.
Exception:	1. The application is not connected to a PC.
Priority:	Essential.

Table (4.10) Explore Processes Use Case.

11. Close process use case shows how the user can close a process, while he is exploring the processes.

Use Case:	Close Process
Primary Actor:	User.
Goal in Context:	To enable the user to use the task manager to close a running process.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to use his smart phone as a mouse.
Scenario:	1. The user clicks on the task manager. 2. The application will show the processes. 3. The user then selects the process and chooses close. 4. The application will send a command to close that file.
Exception:	1. The application is not connected to the internet. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.11) Close Process Use Case.

12. View process information shows what happens when the user decides to see a running process information.

Use Case:	View process information.
Primary Actor:	User.
Goal in Context:	To enable the user to see a process information.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to use his smart phone as a mouse.
Scenario:	1. The user clicks on the task manager button. 2. The application will the current processes. 3. The user then selects a process to see its information. 4. The application will send a command to receive the process information, and will show it.
Exception:	1. The application is not connected to the internet. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.12) View Process Information Use Case.

13. Increasing the sound volume use case, showing how increasing the sound in the PC occurs.

Use Case:	Increase Sound Volume.
Primary Actor:	User.
Goal in Context:	To enable the user to increase the sound volume.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to decrease the sound volume in his computer.
Scenario:	1. The user clicks on the sound volume button. 2. A scroll bar volume will be shown. 3. The user changes its value to up. 4. The application sends a command to increase the volume.
Exception:	1. The application is not connected to the internet. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.13) Increase Sound Volume Use Case.

14. Decreasing the sound volume use case, showing how decreasing the sound in the PC occurs.

Use Case:	Decrease Sound Volume.
Primary Actor:	User.
Goal in Context:	To enable the user to decrease the sound volume.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to use his smart phone as a mouse.
Scenario:	1. The user clicks on the sound volume button. 2. A scroll bar volume will be shown. 3. The user changes its value to down. 4. The application sends a command to increase the volume.
Exception:	1. The application is not connected to the internet. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.14) Decrease Sound Volume Use Case.

15. Increase the brightness use case.

Use Case:	Increase brightness.
Primary Actor:	User.
Goal in Context:	To enable the user to increase the PC screen brightness.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to use his smart phone as a mouse.
Scenario:	1. The user clicks on the brightness button. 2. A scroll bar volume for brightness will be shown. 3. The user changes its value to up. 4. The application sends a command to increase the volume.
Exception:	1. The application is not connected to the internet. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.15) Increase Brightness Use Case.

16. Decrease the brightness use case.

Use Case:	Decrease brightness.
Primary Actor:	User.
Goal in Context:	To enable the user to decrease his PC screen brightness.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the internet.
Trigger:	The user decides to decrease his PC screen brightness.
Scenario:	1. The user clicks on the brightness button. 2. A scroll bar volume will be shown. 3. The user changes its value to down. 4. The application sends a command to decrease the volume.
Exception:	1. The application is not connected to the internet. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.16) Decrease Brightness Use Case.

17. Shut down use case, shows when the user decides to shut down his PC remotely.

Use Case:	Shut Down
Primary Actor:	User.
Goal in Context:	To enable the user to shut down his PC.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the PC.
Trigger:	The user decides to shut down his PC.
Scenario:	1. The user clicks on the power options. 2. Power options menu will appear. 3. The user then clicks on shut down. 4. The application sends the command to the PC, and the PC shut down.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.17) Shut Down PC Use Case.

18. Log Out use case, shows when the user decides to log out from his PC remotely.

Use Case:	Log Out
Primary Actor:	User.
Goal in Context:	To enable the user to log out from his PC.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the PC.
Trigger:	The user decides to log out from his PC.
Scenario:	1. The user clicks on the power options. 2. Power options menu will appear. 3. The user then clicks on Log Out. 4. The application sends the command to the PC, and the PC Logs Out.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.18) Log Out PC Use Case.

19. Restart use case, shows when the user decides to restart from his PC remotely.

.

Use Case:	Restart
Primary Actor:	User.
Goal in Context:	To enable the user to restart his PC.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the PC.
Trigger:	The user decides to shut down his PC.
Scenario:	1. The user clicks on the power options. 2. Power options menu will appear. 3. The user then clicks on restart. 4. The application sends the command to the PC, and the PC restart.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

Table (4.19) Restart PC Use Case.

20. Sleep use case, shows when the user decides to sleep from his PC remotely.

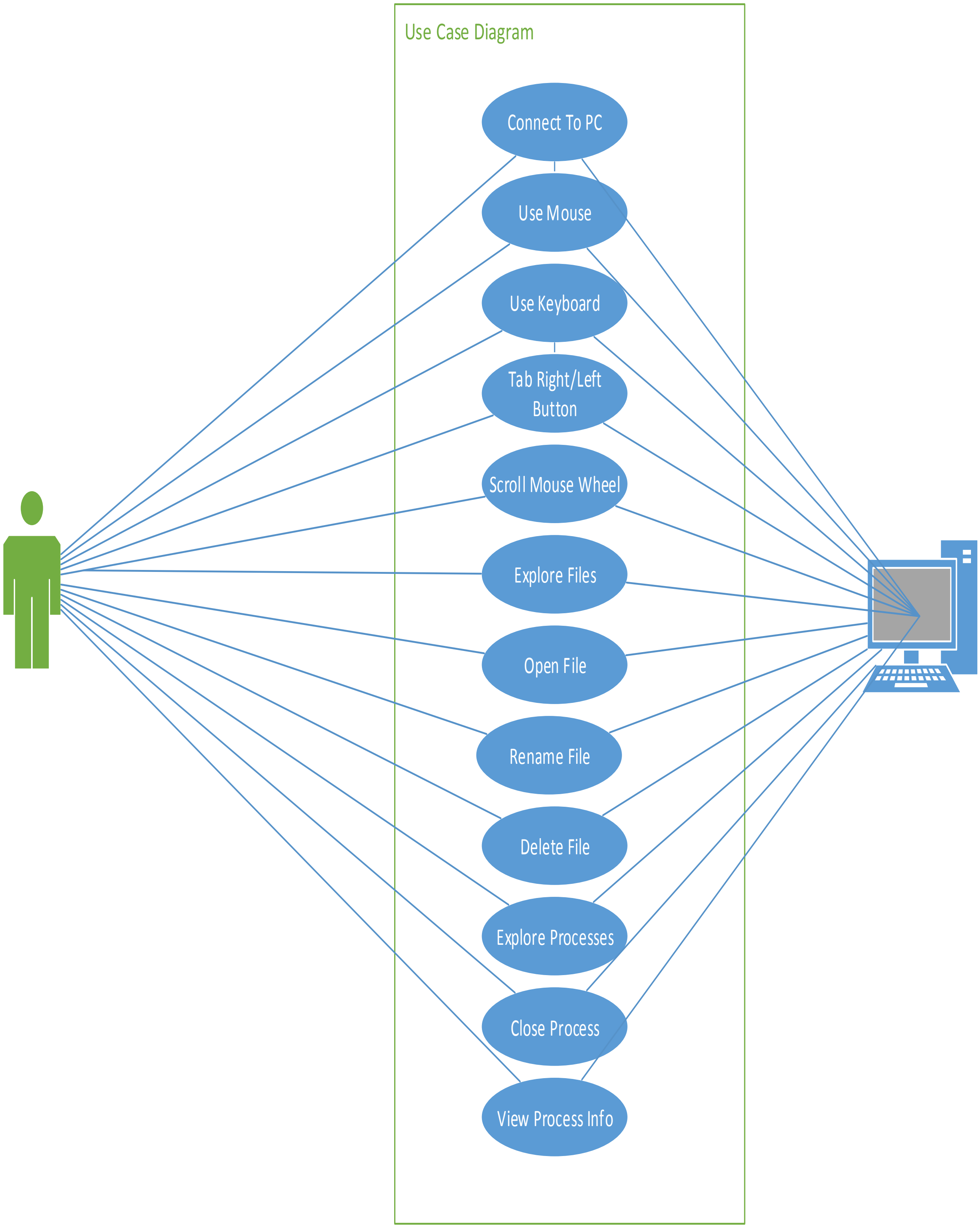
Use Case:	Sleep
Primary Actor:	User.
Goal in Context:	To enable the user to turn his PC into sleep mode.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the PC.
Trigger:	The user decides to sleep his PC.
Scenario:	1. The user clicks on the power options. 2. Power options menu will appear. 3. The user then clicks on sleep. 4. The application sends the command to the PC, and the PC turns on sleep mode.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

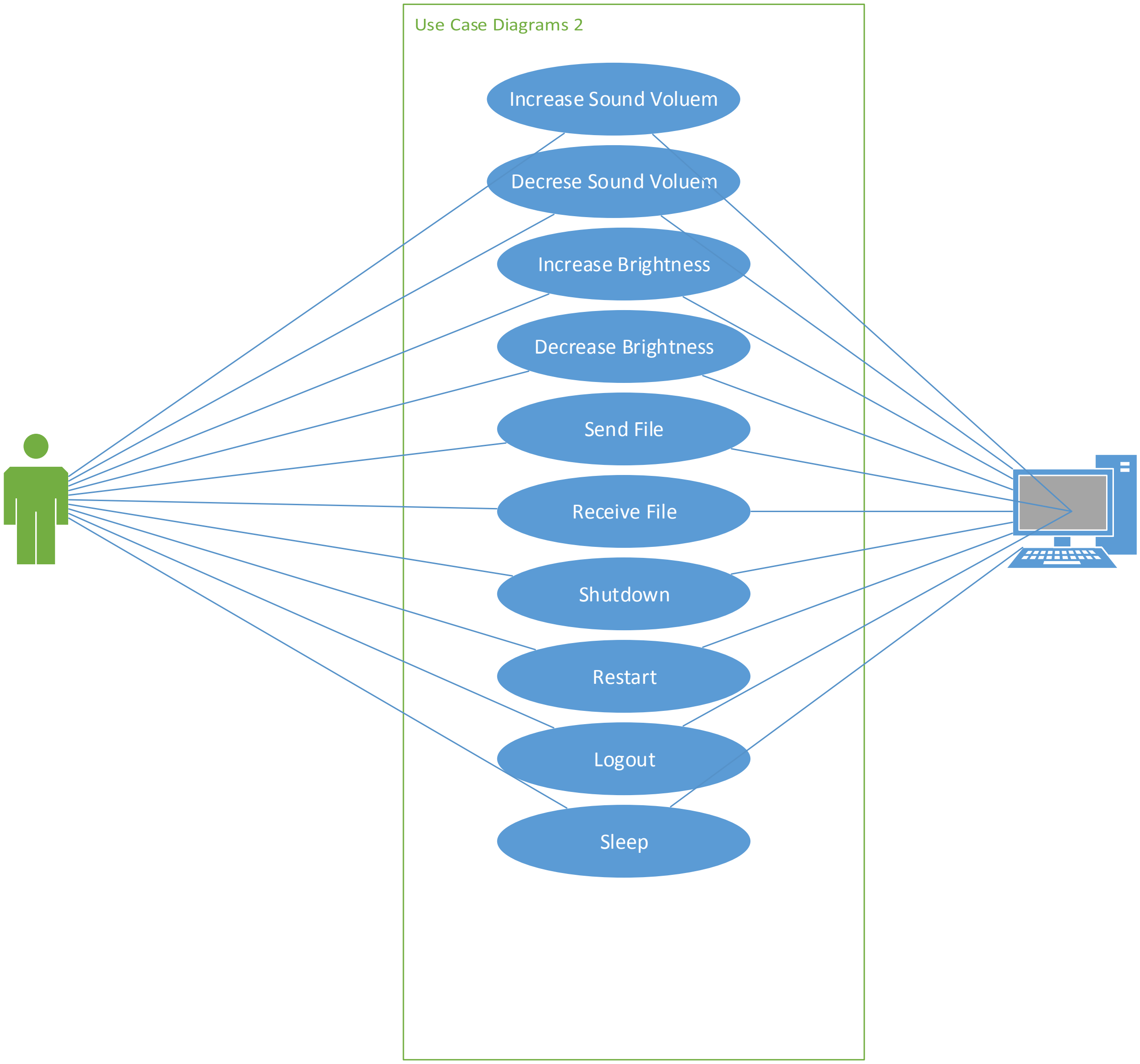
20. Download file use case, shows when the user decides to get a file from his PC.

Use Case:	Download File
Primary Actor:	User.
Goal in Context:	To enable the user to send a file to his PC.
Precondition:	1. The application is connected to a PC. 2. The application is connected to the PC.
Trigger:	The user decides to sleep his PC.
Scenario:	1. The user clicks on file explorer. 2. The user selects a file and selects download. 3. The application sends a command to the PC application to download the file. 4. The PC application starts to send the file and the smart phone to receive it.
Exception:	1. The application is not connected to the network. 2. The application is not connected to a PC.
Priority:	Essential.

B. Use Cases Diagrams.

The following is the general Use Case Diagram but divided into two parts in regard to its size.




Use Case Diagrams 2
Increase Sound Voluem
Decrese Sound Voluem
Increase Brightness
Decrease Brightness
Send File
Receive File
Shutdown
Restart
Logout
Sleep

3.4 CRC Cards.

CRC’s for android application:

1. Keyboard Activity CRC.

This class is used to handle the keyboard keystrokes that the user press.

Keyboard Activity.	
Handles Keystrokes.	PCConnector.
Converts Keystrokes to integral values.	
Shows output in text box.	

2. Main Activity CRC.

The main activity class contains the tabs for the applications.

Main Activity.	
Connects to PC.	PCConnector

3. Mouse Activity.

This activity is responsible to handle the user figure motions and sending them to the user.

Mouse Activity.	
Handles user touch and calculates the next position to send.	PCConnector.
Handles the buttons keystrokes.	
Handles the mouse wheel move.	

7 Task Manager Activity.

This activity is responsible to view the processes and sending the commands to manage them.

Task Manager Activity.	
Shows the processes info.	
Handles user commands.	PCConnector
Request Processes	PCConnector

8 File Manger Activity.

This activity is used to view the files and to handle the user commands to be implemented on the PC.

File manager activity.	
Shows files and directories.	
Handles user commands.	PCConnector.

9 PC Connecter Class.

This class is used to connect the PC to the smart phone and transferring the data encrypted.

PCConnecter.	
Connects to PC.	
Sends command.	
Receives info.	
Defines PC Address	

CRCs for PC Application:

1. Keyboard Handler Class.

This class performs the keyboard keystrokes that is received from the client side.

Keyboard Handler.	
Handles Keystrokes.	AndroidConnector.
Executes Keystrokes.	
Handles Special Keystrokes.	AndroidConnector

2. Mouse Handler Class.

This class is used to receive the cursor location and move it, and to perform the received clicks.

Mouse Handler.	
Handles New Mouse Positions.	AndroidConnector.
Handles Mouse Clicks.	Android Connector.
Moves the Cursor.	
Executes Mouse Clicks.	
Executes Mouse Wheel Movement.	
Handles Wheel Scroll	Android Connector.

3. File Manager Class.

This class performs the operations received to manage the files i.e. (delete, explore, rename,move, and open).

File Manager.	
Explore Path.	
Gets file list for path.	
Handles User Commands	AndroidConnector
Move Current Location.	
Delete File.	
Rename File.	
Open File.	

4. Processes Manger Class.

This class is used to get the processes and to manage them.

Processes Manager.	
Gets Processes Info.	
Handles User Commands.	AndroidConnector.
Close a process.	
Gets Running User Application.	

5. System Properties Handler Class.

The System properties class is used to control the system features such as the sound volume and the brightness volume.

System Properties Handler.	
Increases Screen Brightness.	
Decreases Screen Brightness.	
Increases Sound Volume.	
Decreases Sound Volume.	
Handles User Commands.	Android Connector.

7. Main Controller Class.

This class interprets the commands received and passing them to the current class.

Main Controller.	
Connects the controllers and handlers with the AndroidConnector.	AndroidConnector.

8. Smart Phone Connector Class.

The smart phone class is used to listen on a specific port for the smart phone connection and then transferring the data.

Smart Phone Connecter.	
Listens for New Connections.	
Accepts Connections.	
Sends Info.	
Receives Commands.	

8. Power Manger Class.

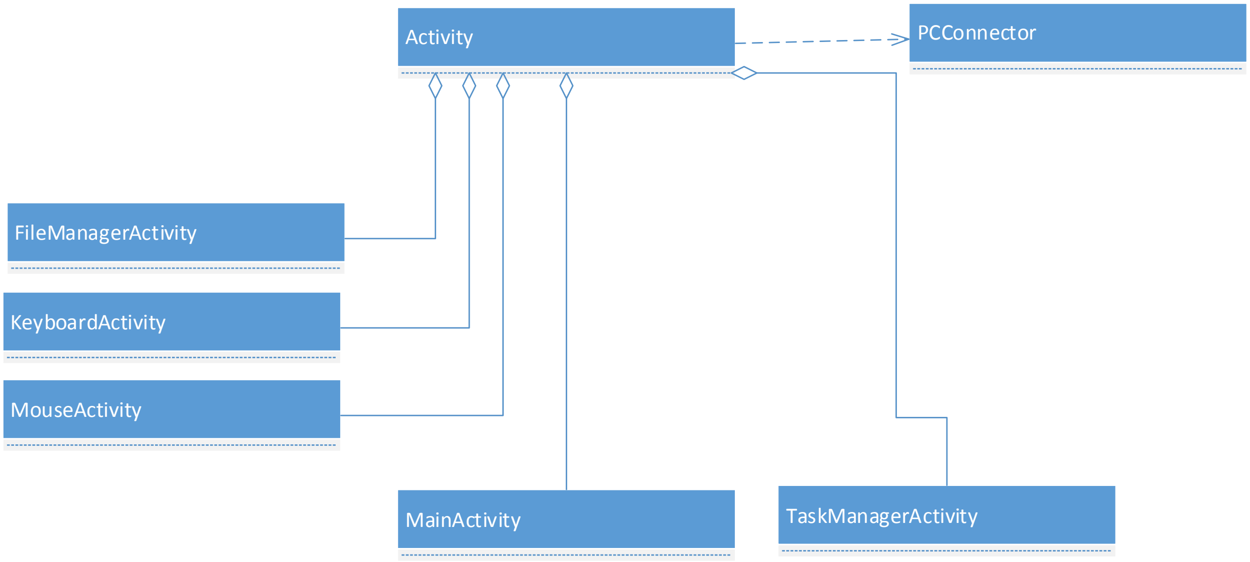
The power manager class is used to control the power options (shutdown, sleep, logout, and restart).

Power Manager.	
Handles user commands.	Android Connector.
Shut down the PC.	
Sleep the PC	
Log Out from the PC.	
Restart The PC.	

3.4 Class Hierarchies and Relationships.

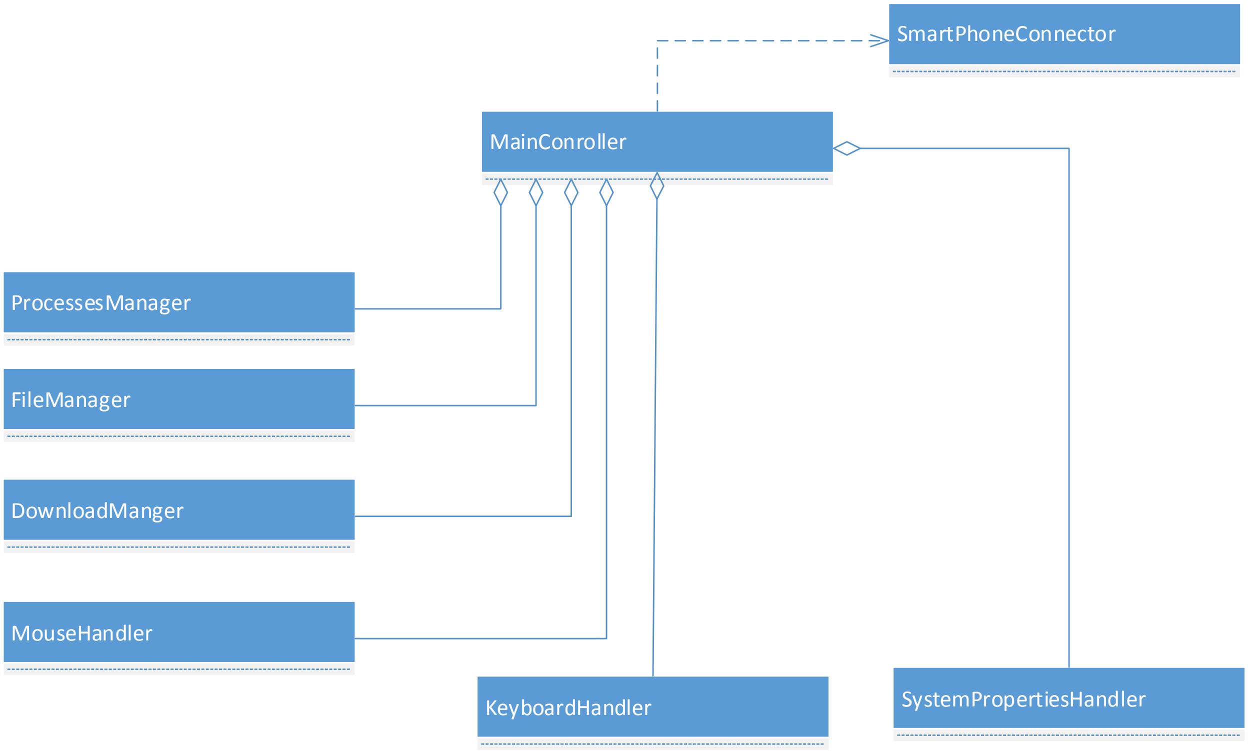
The following is the class hierarchies and relationships for the client side application on the smart phone.

Class Hierarchies and Relationships for Android:



Class Hierarchies and Relationships for PC:

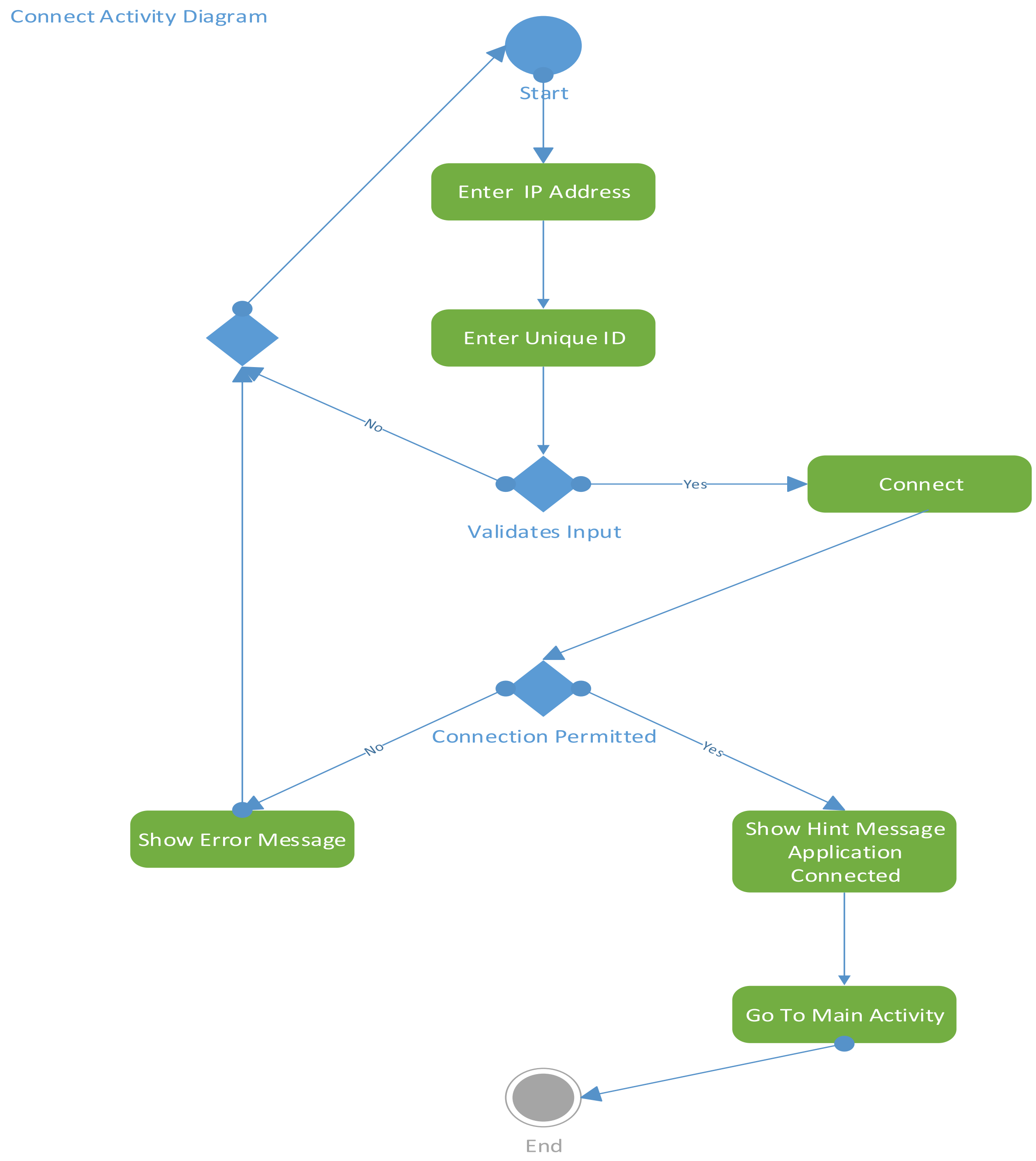
The following is the class hierarchies and relationships for the server side application on the PC, The main controller has a dependency on the SmartPhoneConnector, and the ProcessesManager, FileManager, DownloadManager, MouseHandler, KeyboardHandler and the System Properties Handler are aggregated in MainController.



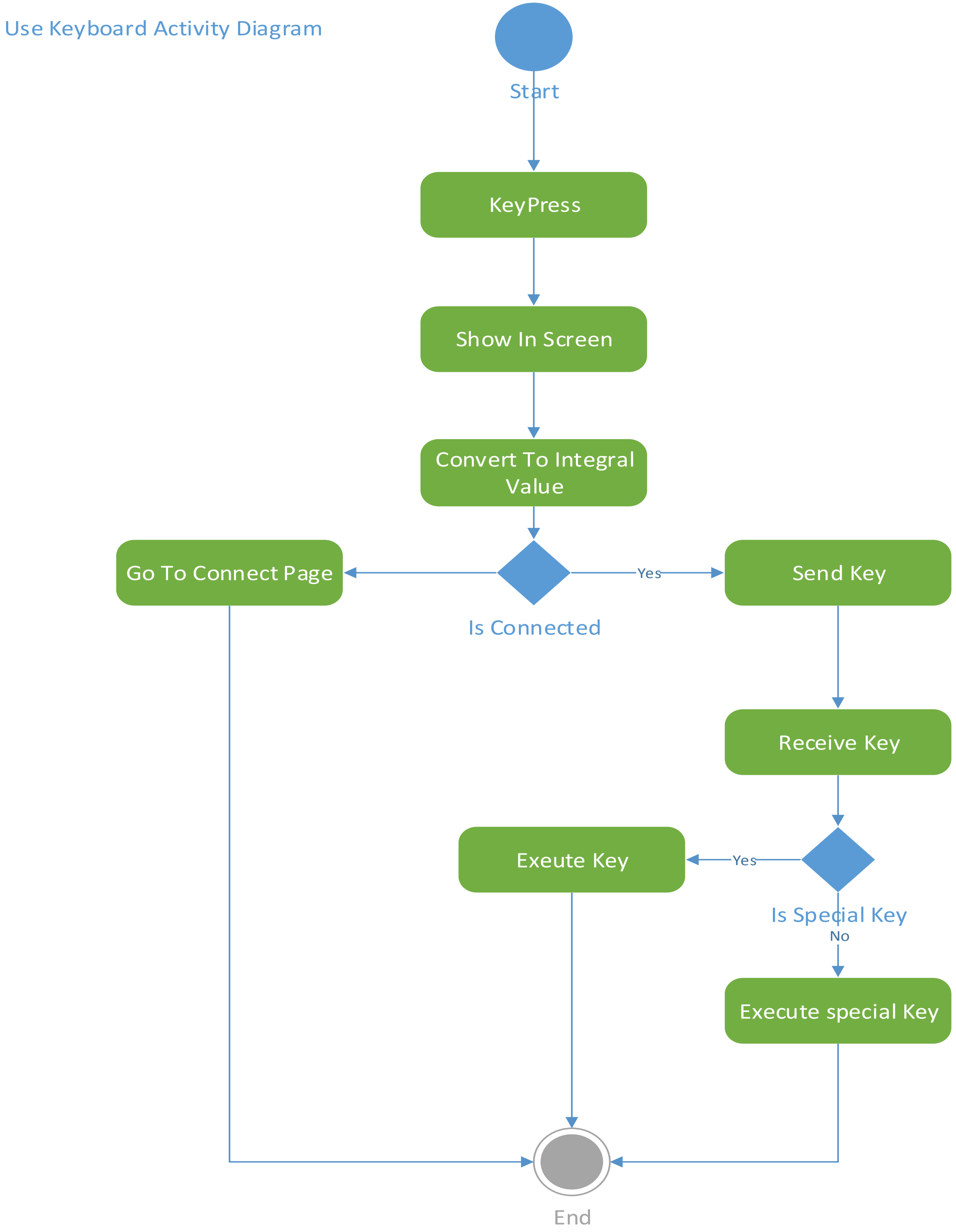
3.5 UML Activity Diagrams.

In this section a UML activity diagram for each use case will be shown.

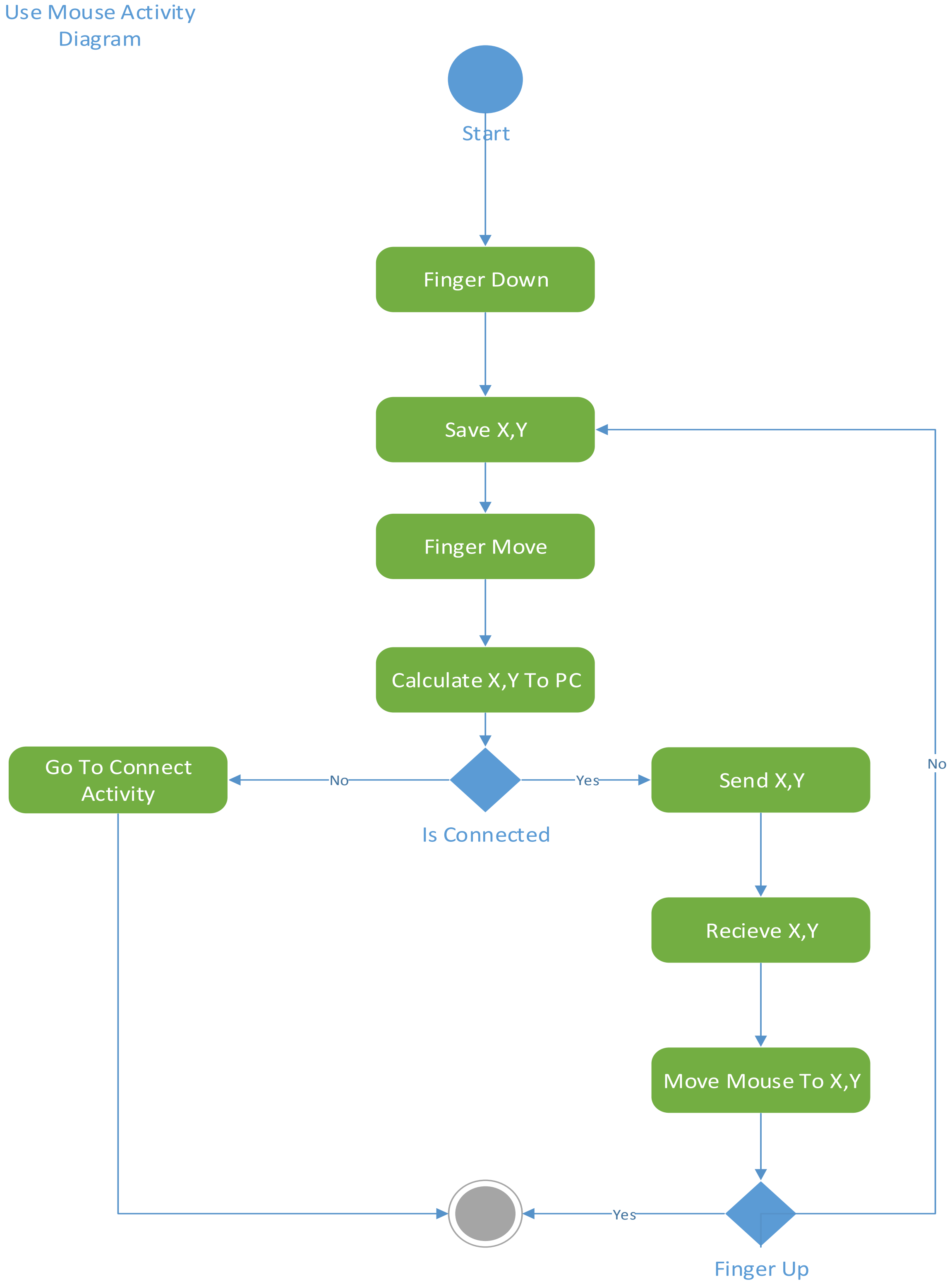
1. Connect to PC Activity diagram.



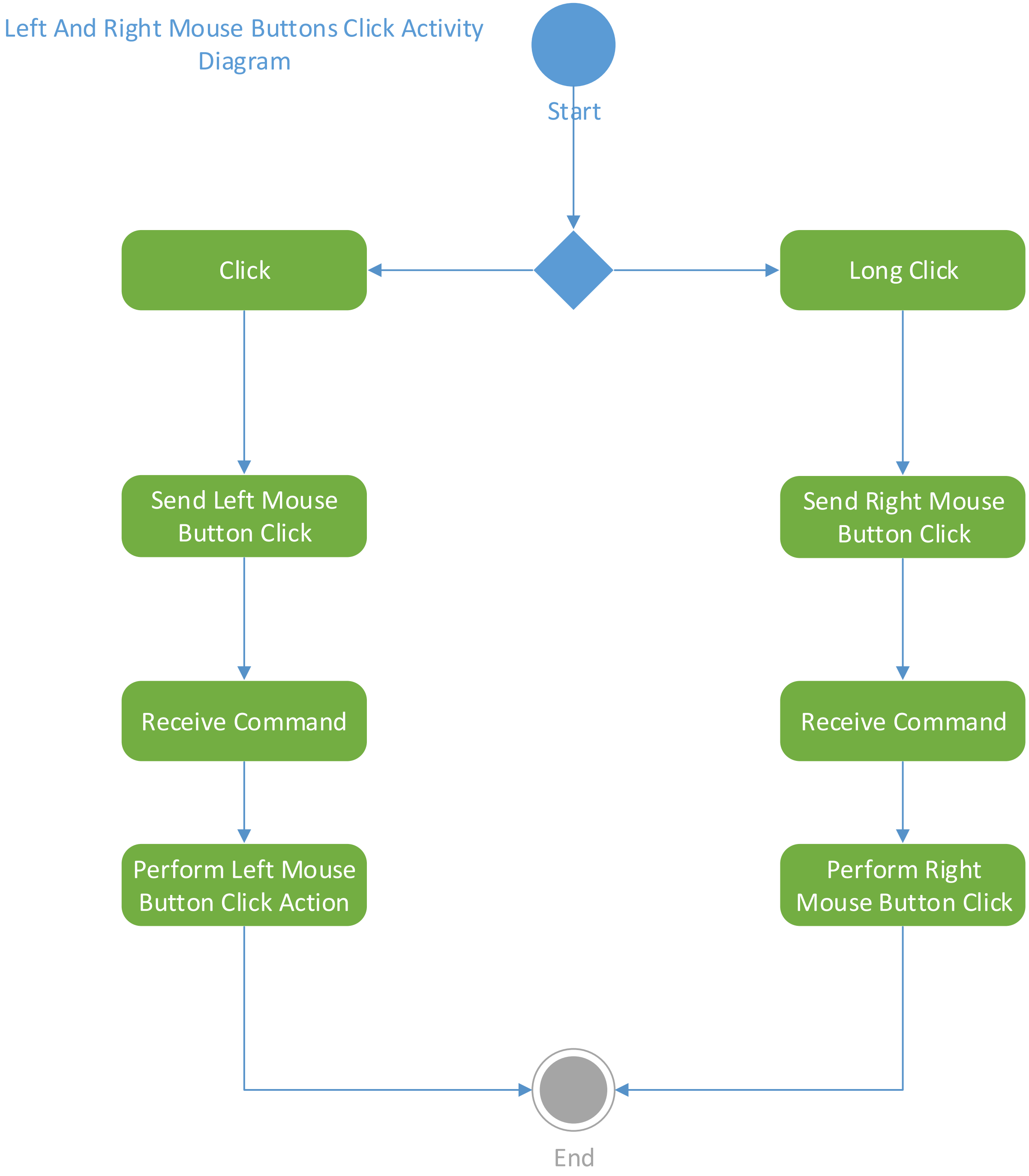
2. Use Keyboard Activity diagram.



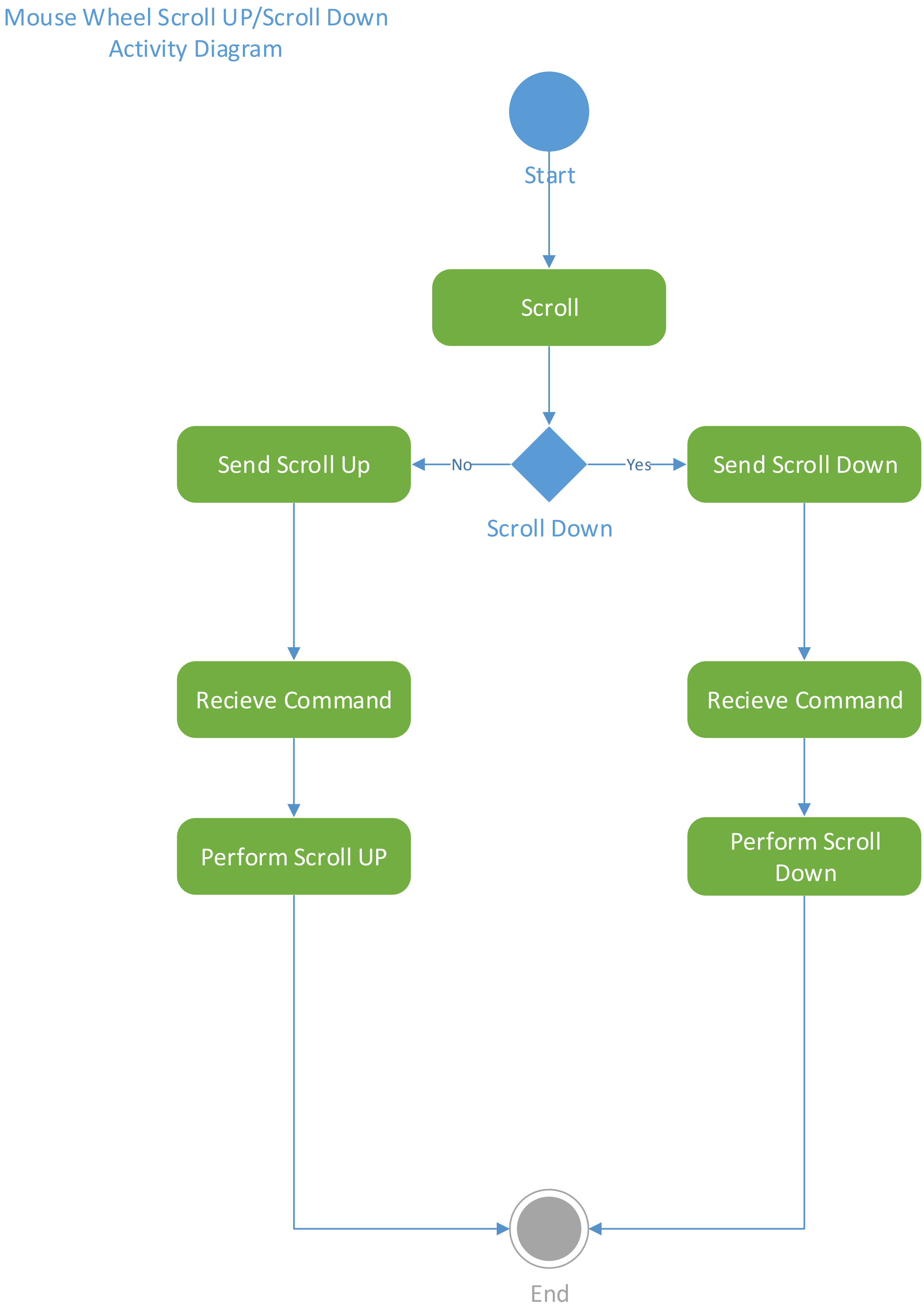
3. Use Mouse Activity Diagram.



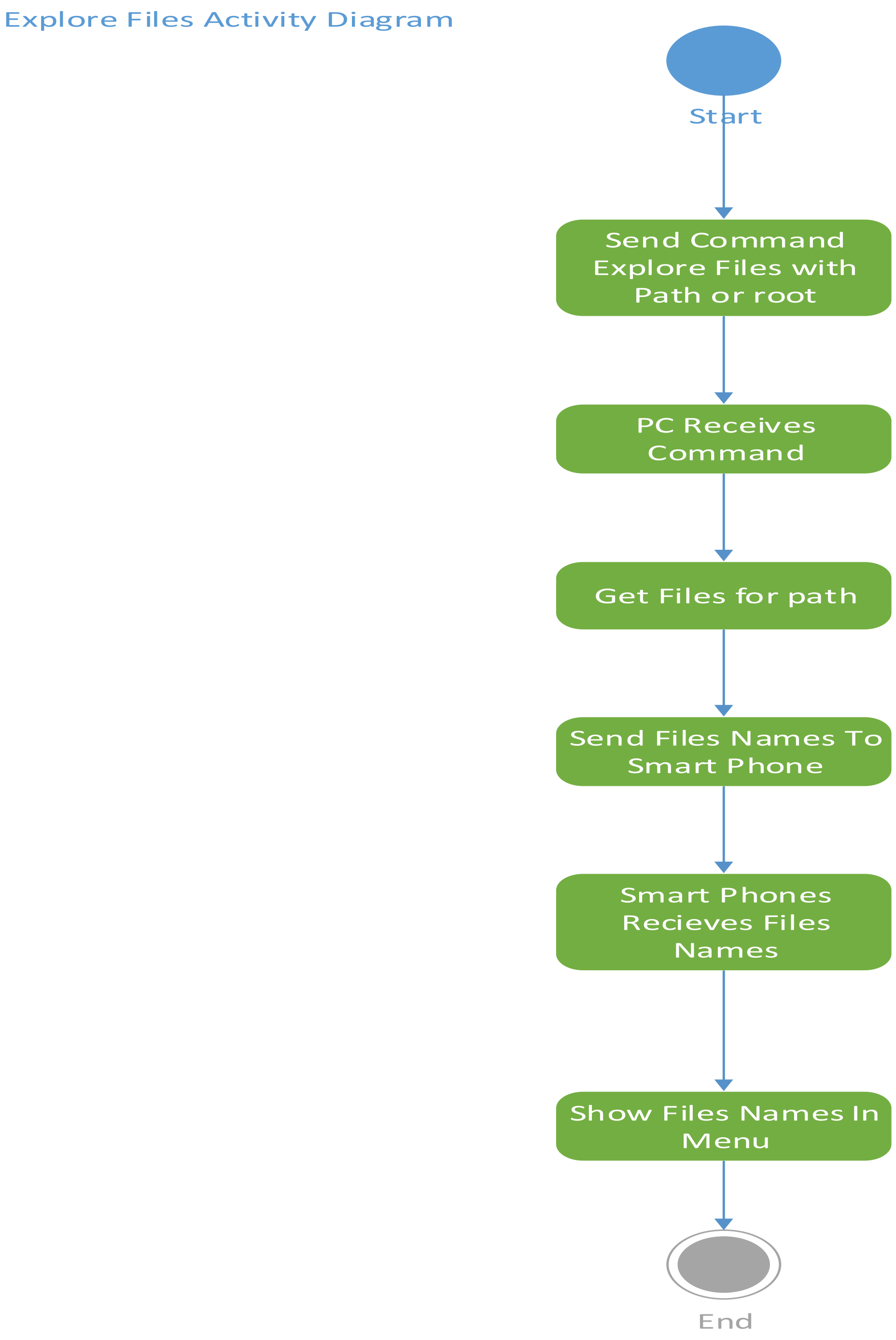
4. Left and right mouse buttons click activity diagram.


Left And Right Mouse Buttons Click Activity Diagram
Start
Click
Long Click
Send Left Mouse Button Click
Send Right Mouse Button Click
Receive Command
Receive Command
Perform Left Mouse Button Click Action
Perform Right Mouse Button Click
End

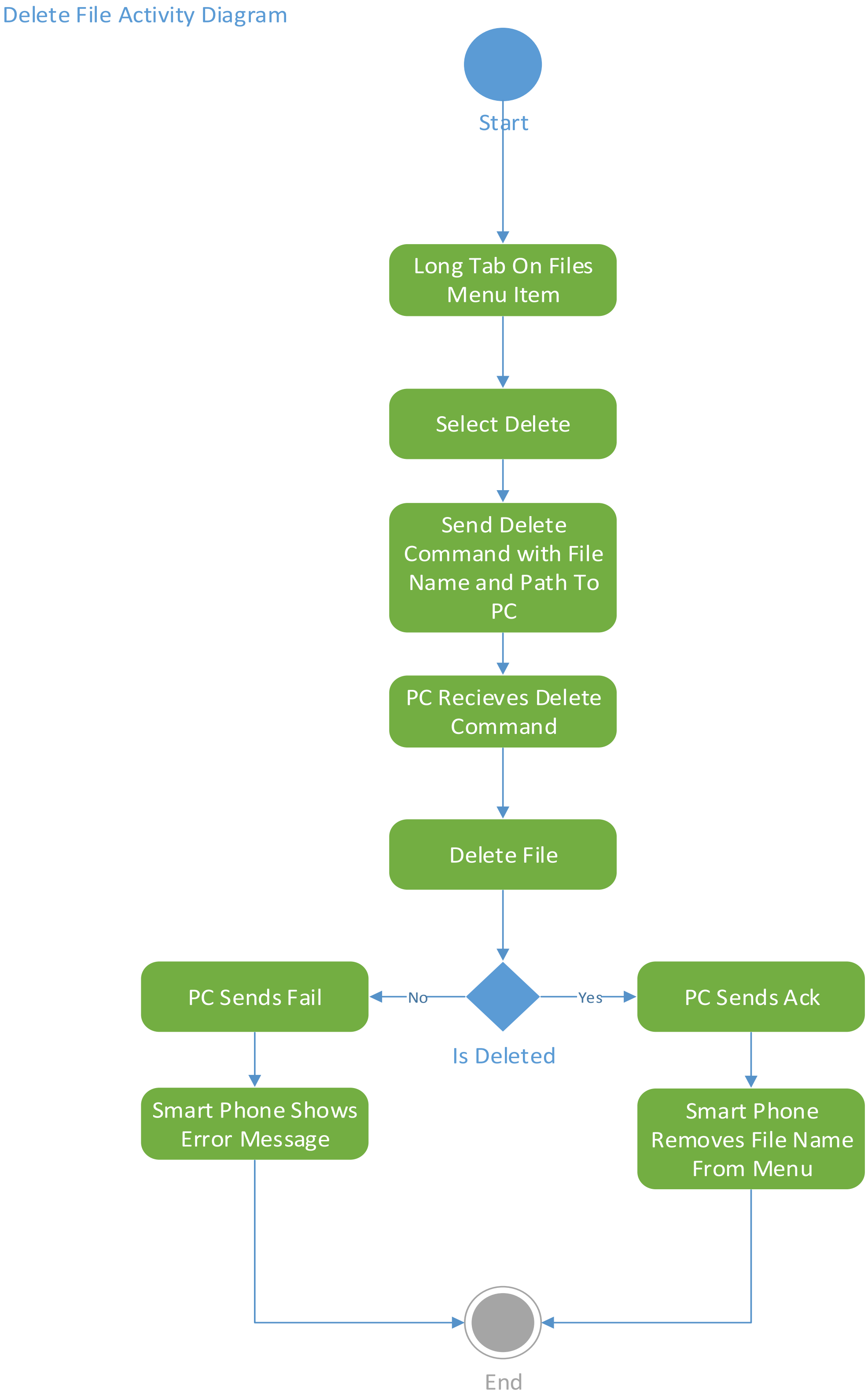
5. Mouse Wheel Scroll UP/Down Activity diagram.



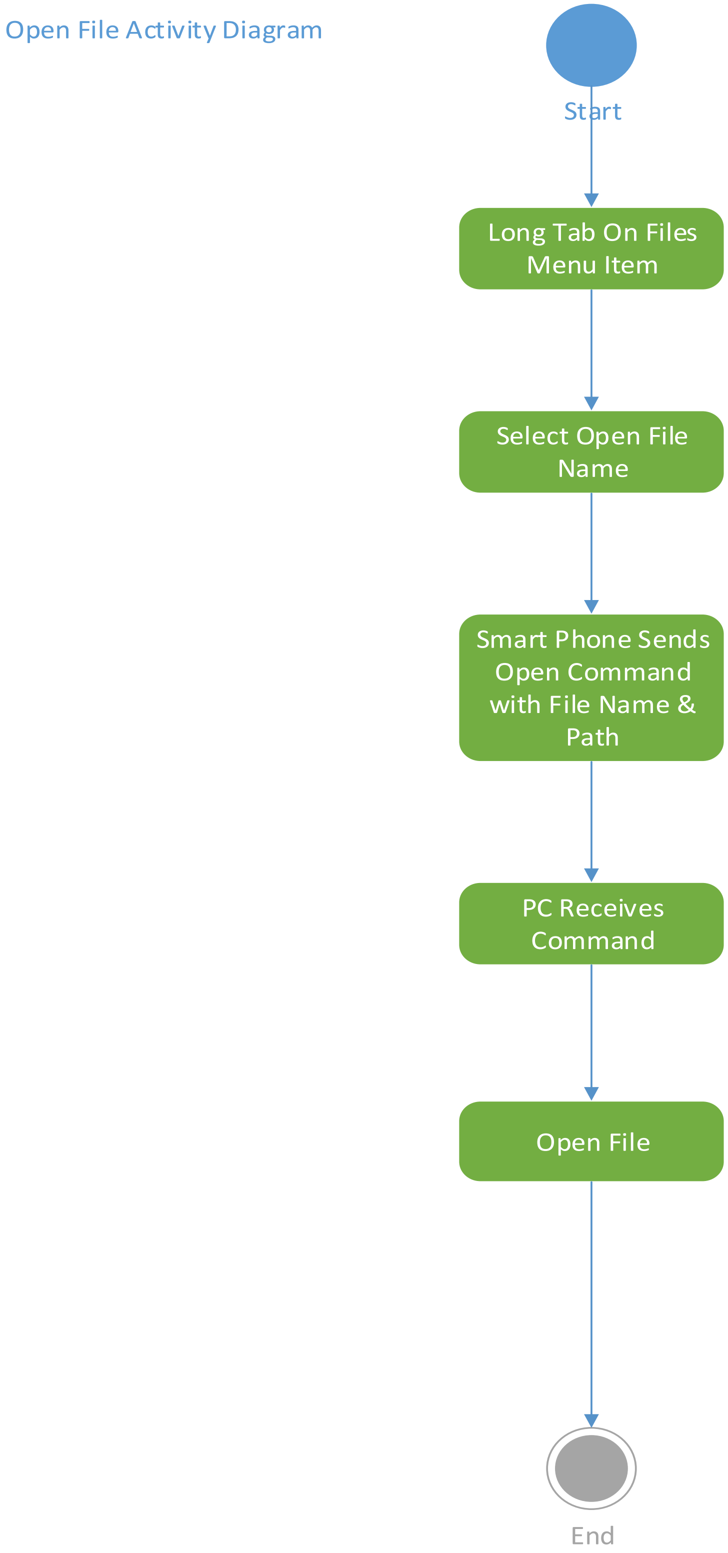
6. Explore Files Activity Diagram.


Explore Files Activity Diagram
Start
Send Command Explore Files with Path or root
PC Receives Command
Get Files for path
Send Files Names To Smart Phone
Smart Phones Recieves Files Names
Show Files Names In Menu
End

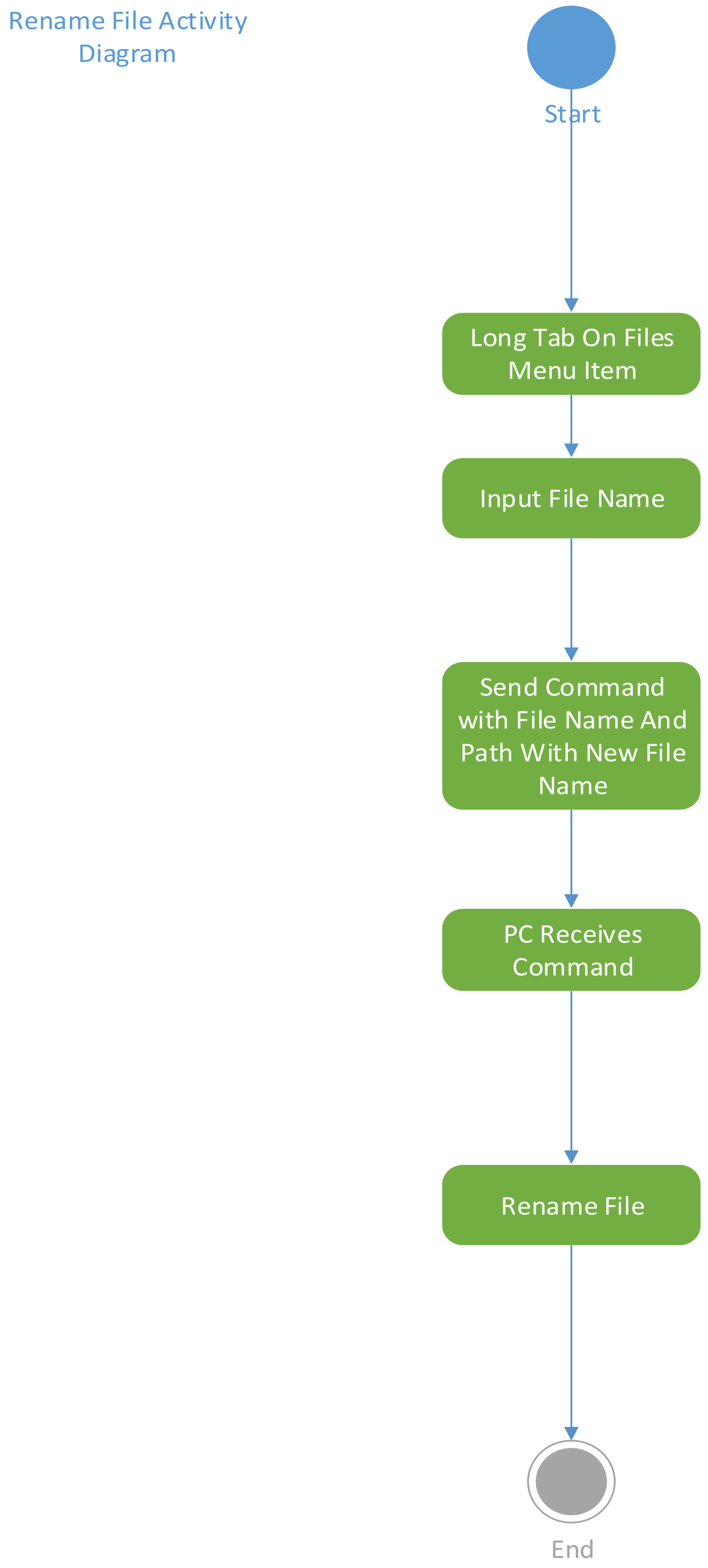
7. Delete File Activity Diagram.


Delete File Activity Diagram
Start
Long Tab On Files Menu Item
Select Delete
Send Delete Command with File Name and Path To PC
PC Recieves Delete Command
Delete File
Is Deleted
No
Yes
PC Sends Fail
PC Sends Ack
Smart Phone Shows Error Message
Smart Phone Removes File Name From Menu
End

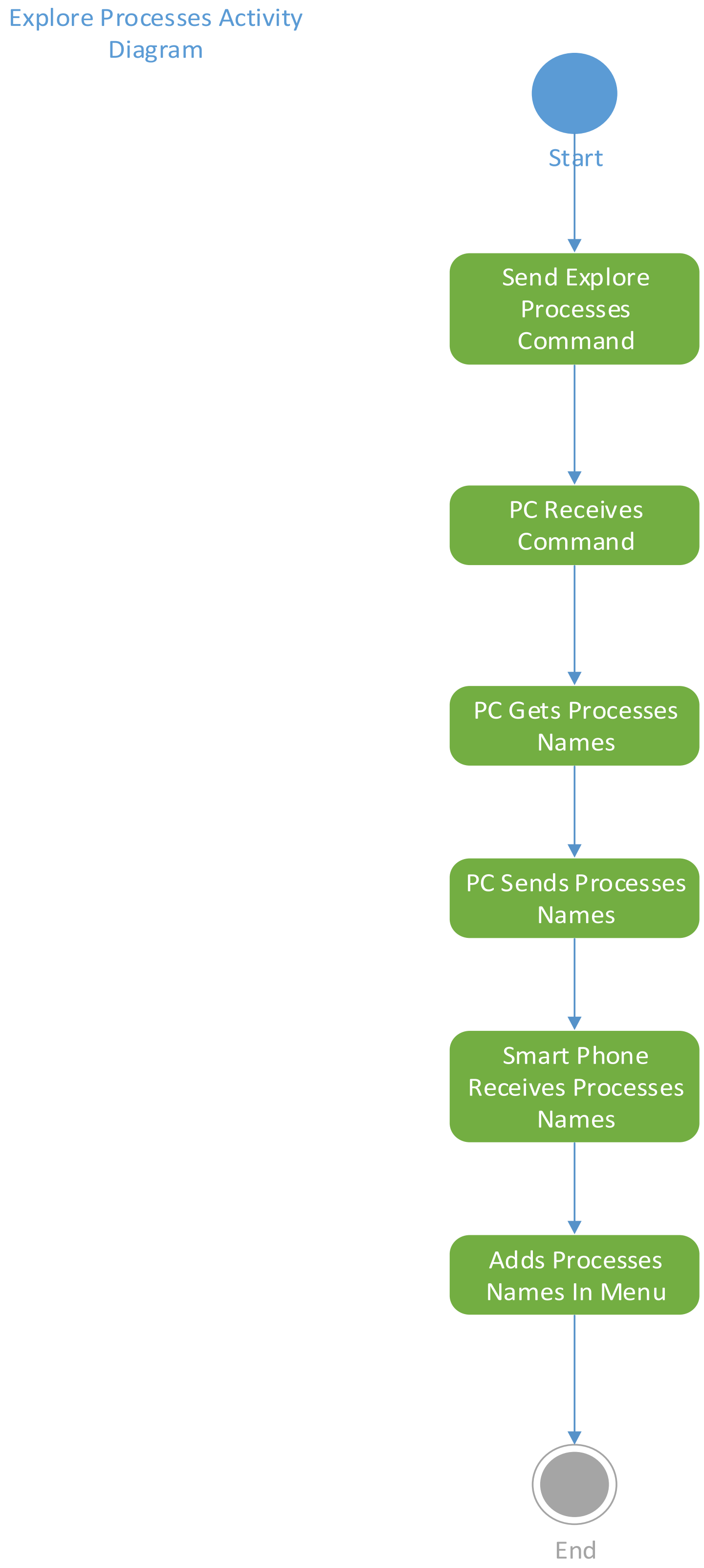
8. Open File Activity Diagram.


Open File Activity Diagram
Start
Long Tab On Files Menu Item
Select Open File Name
Smart Phone Sends Open Command with File Name & Path
PC Receives Command
Open File
End

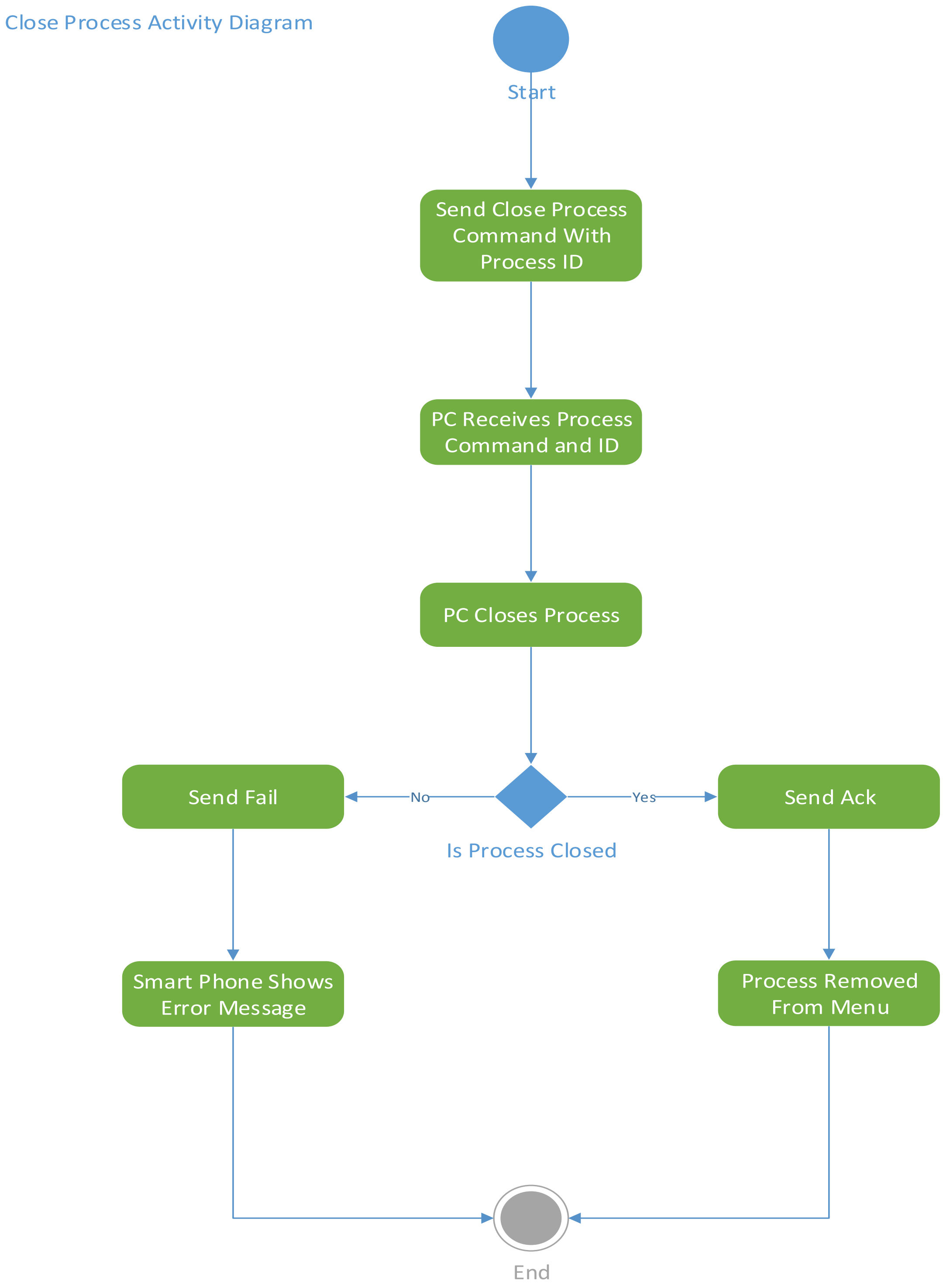
9. Rename File Activity Diagram.


Rename File Activity Diagram
Start
Long Tab On Files Menu Item
Input File Name
Send Command with File Name And Path With New File Name
PC Receives Command
Rename File
End

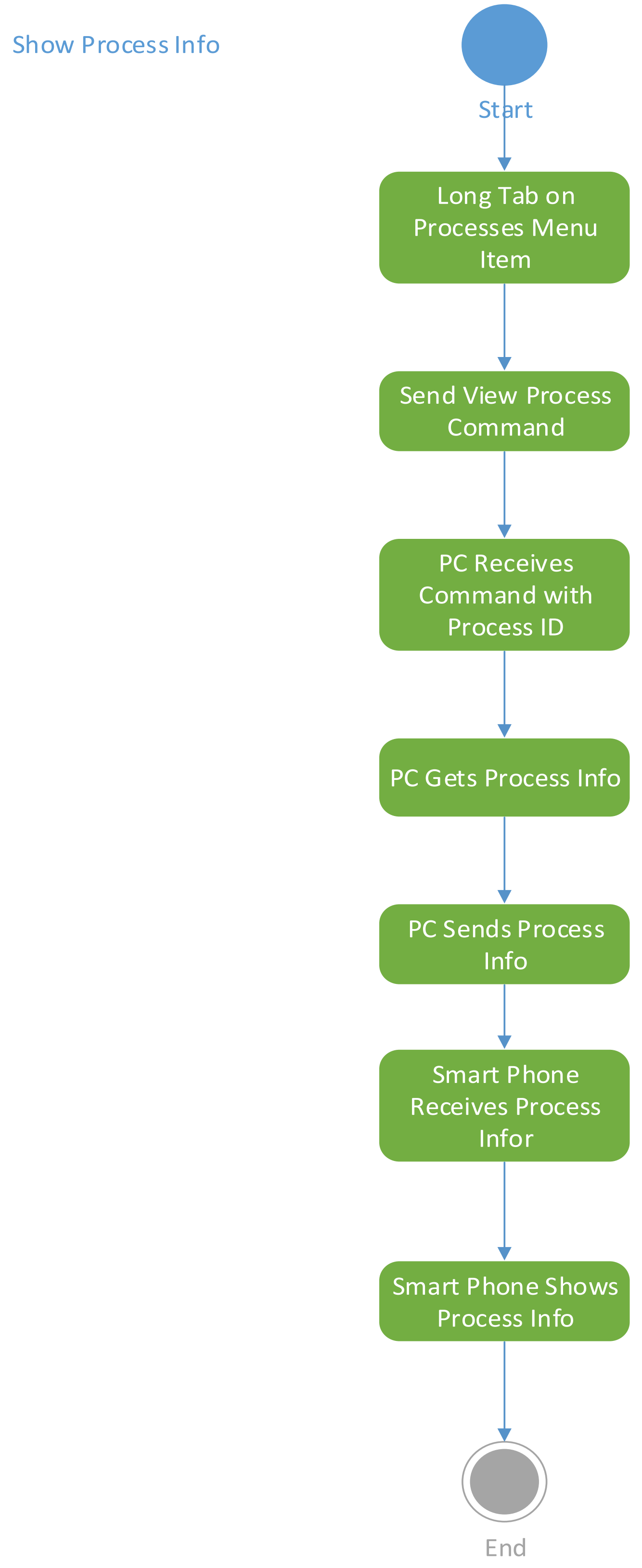
10. Explore Processes Activity Diagram.


Explore Processes Activity Diagram
Start
Send Explore Processes Command
PC Receives Command
PC Gets Processes Names
PC Sends Processes Names
Smart Phone Receives Processes Names
Adds Processes Names In Menu
End

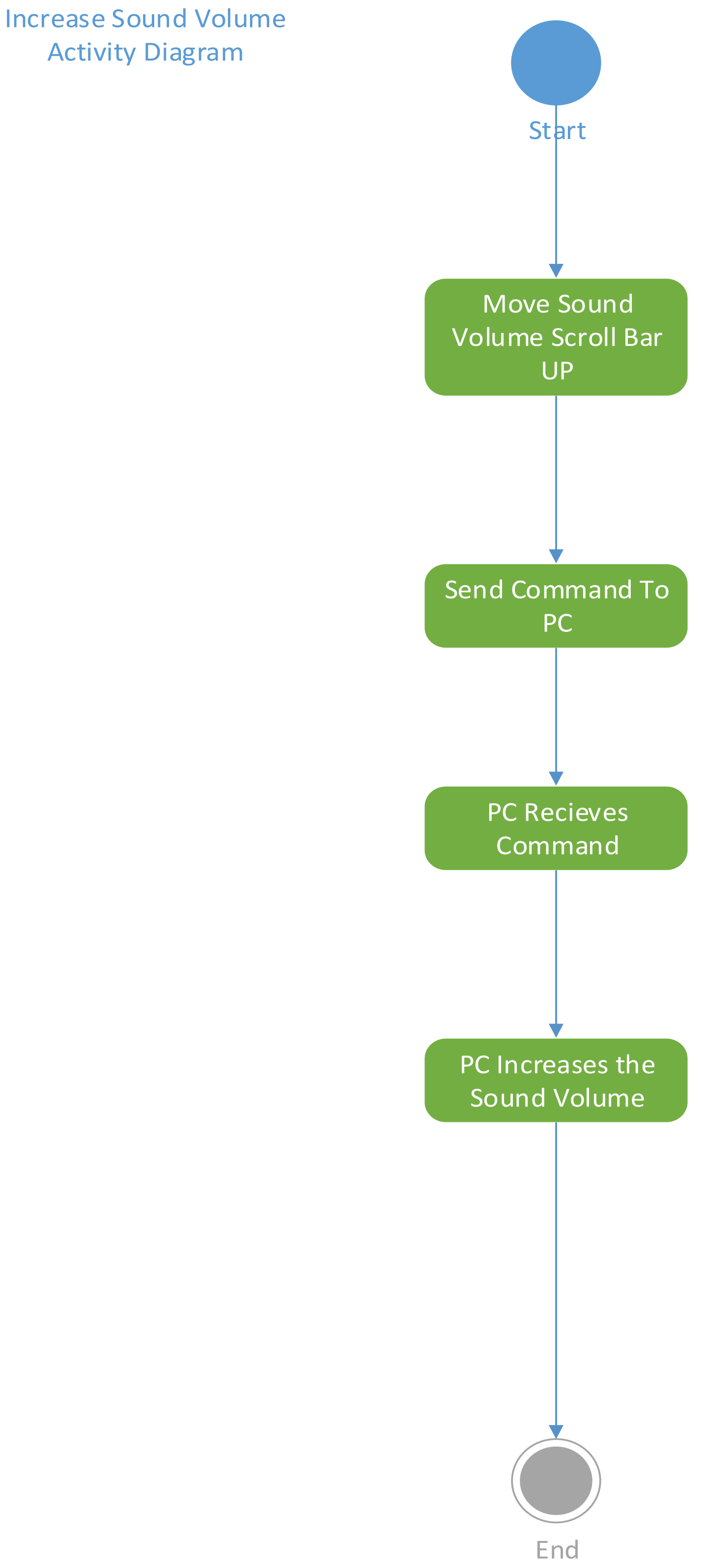
11. Close Process Activity Diagram.


Close Process Activity Diagram
Start
Send Close Process Command With Process ID
PC Receives Process Command and ID
PC Closes Process
Send Fail
No
Yes
Send Ack
Is Process Closed
Smart Phone Shows Error Message
Process Removed From Menu
End

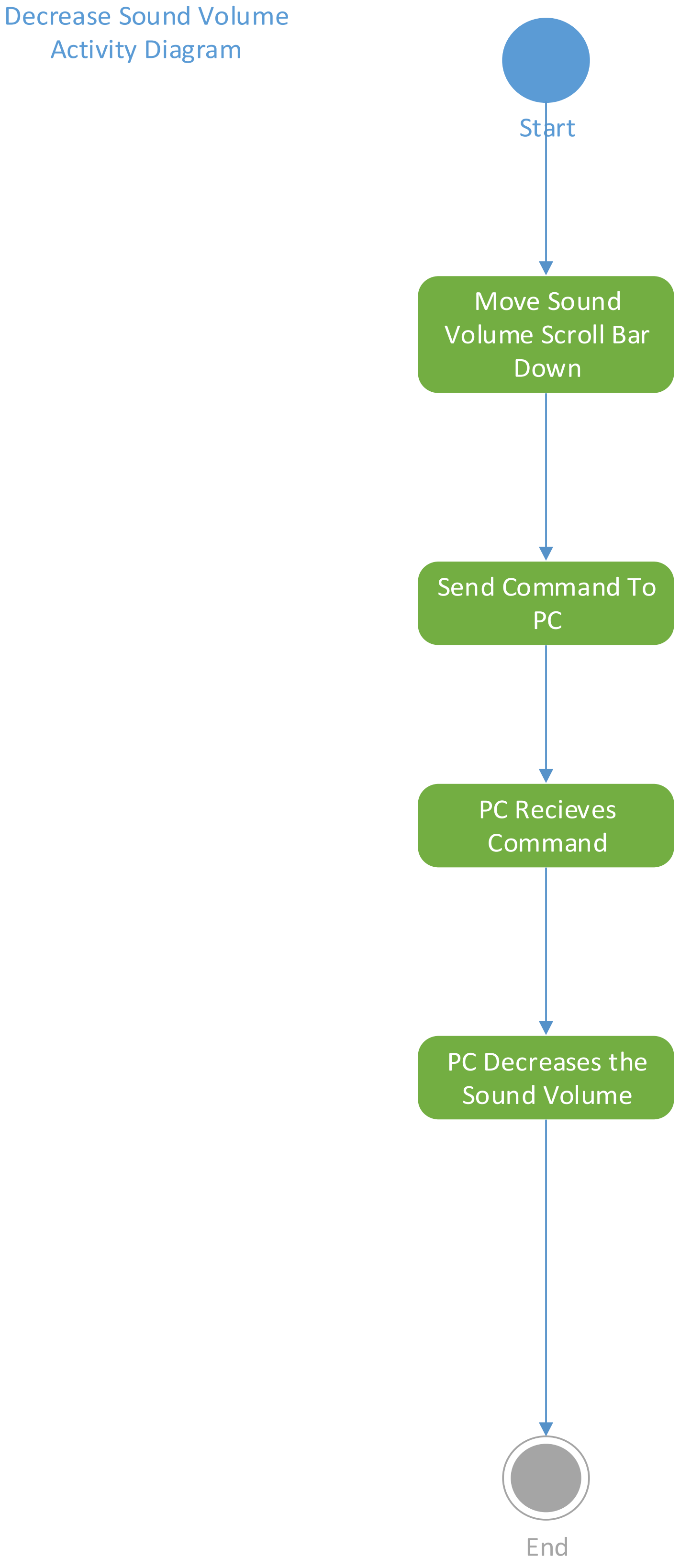
12. Show Process Info Activity Diagram.


Show Process Info
Start
Long Tab on Processes Menu Item
Send View Process Command
PC Receives Command with Process ID
PC Gets Process Info
PC Sends Process Info
Smart Phone Receives Process Infor
Smart Phone Shows Process Info
End

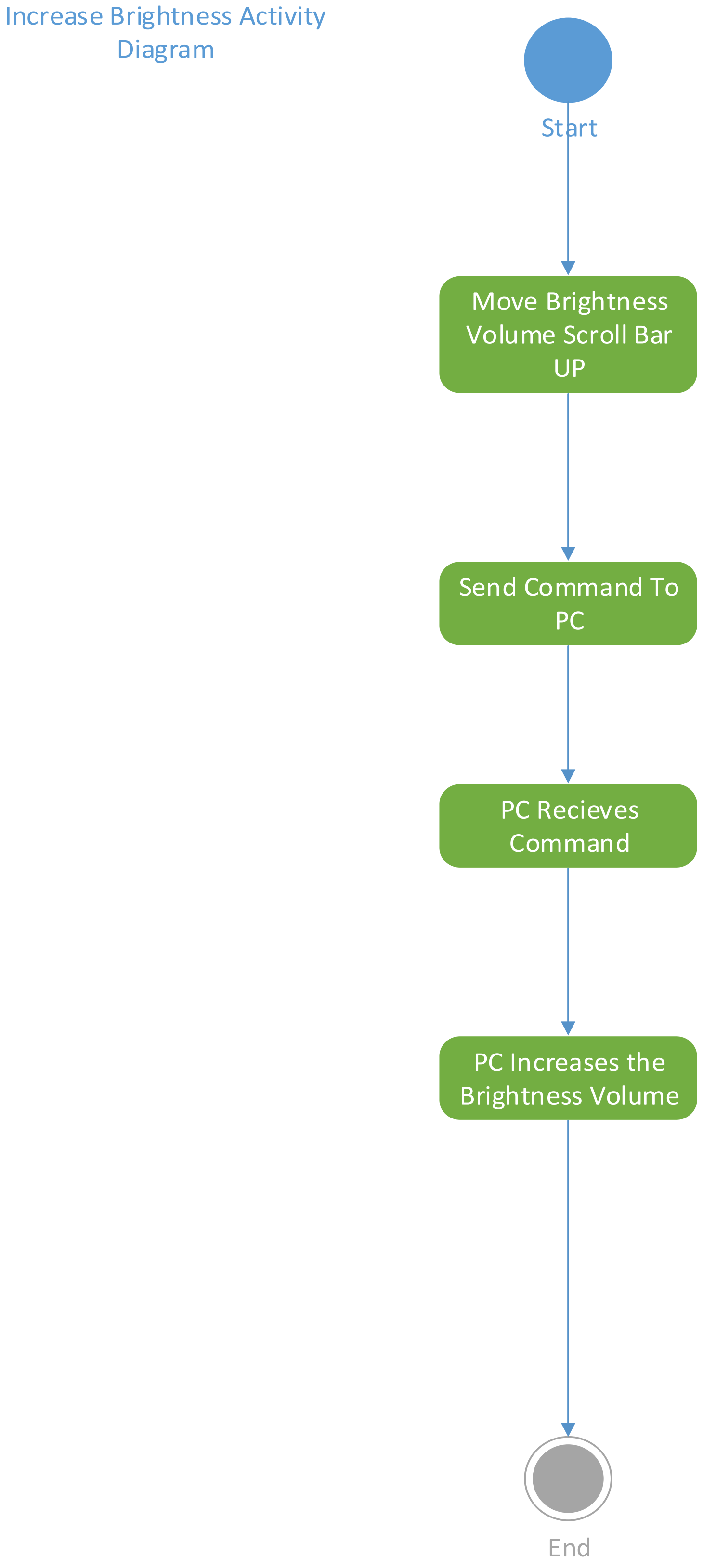
13. Increase Sound Volume Activity Diagram.


Increase Sound Volume Activity Diagram
Start
Move Sound Volume Scroll Bar UP
Send Command To PC
PC Recieves Command
PC Increases the Sound Volume
End

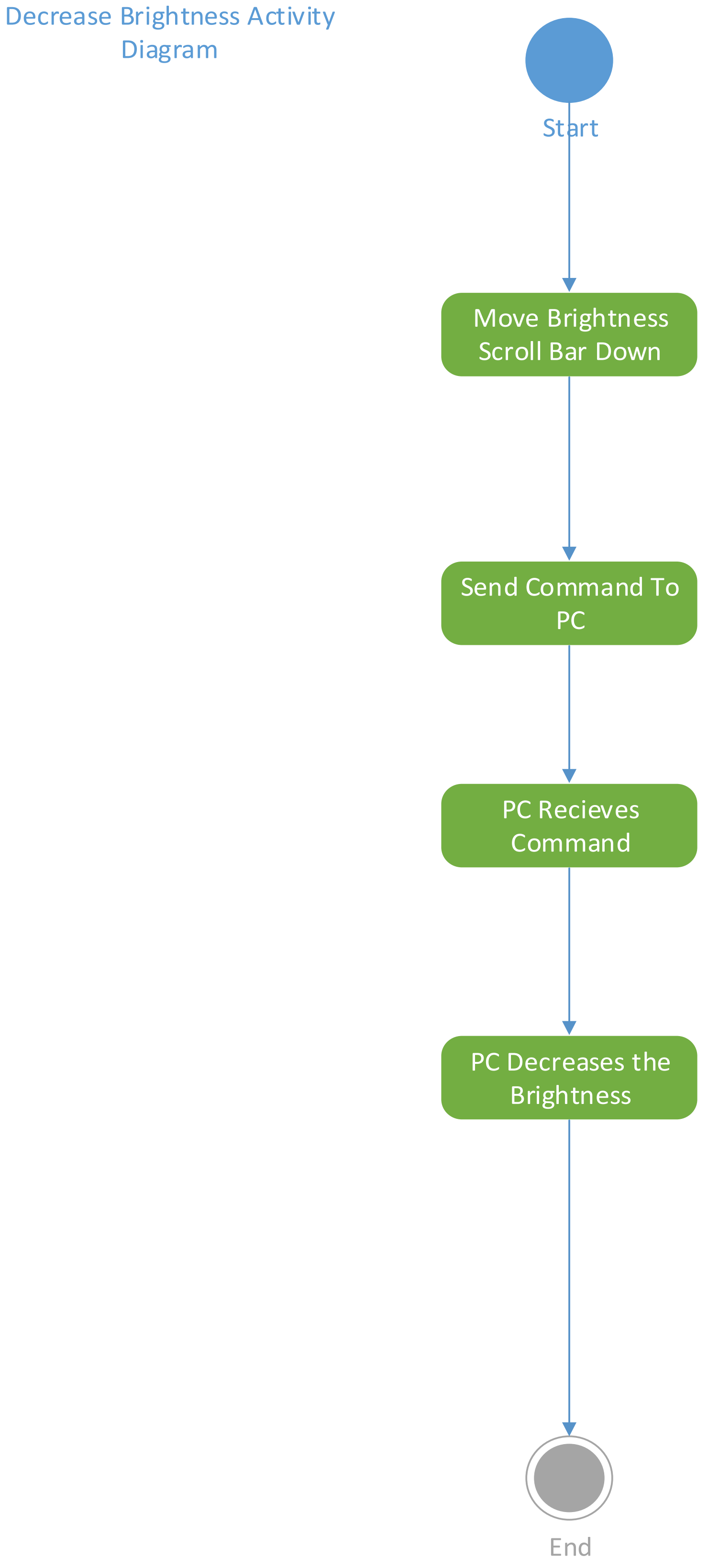
14. Decrease Sound Volume Activity Diagram.



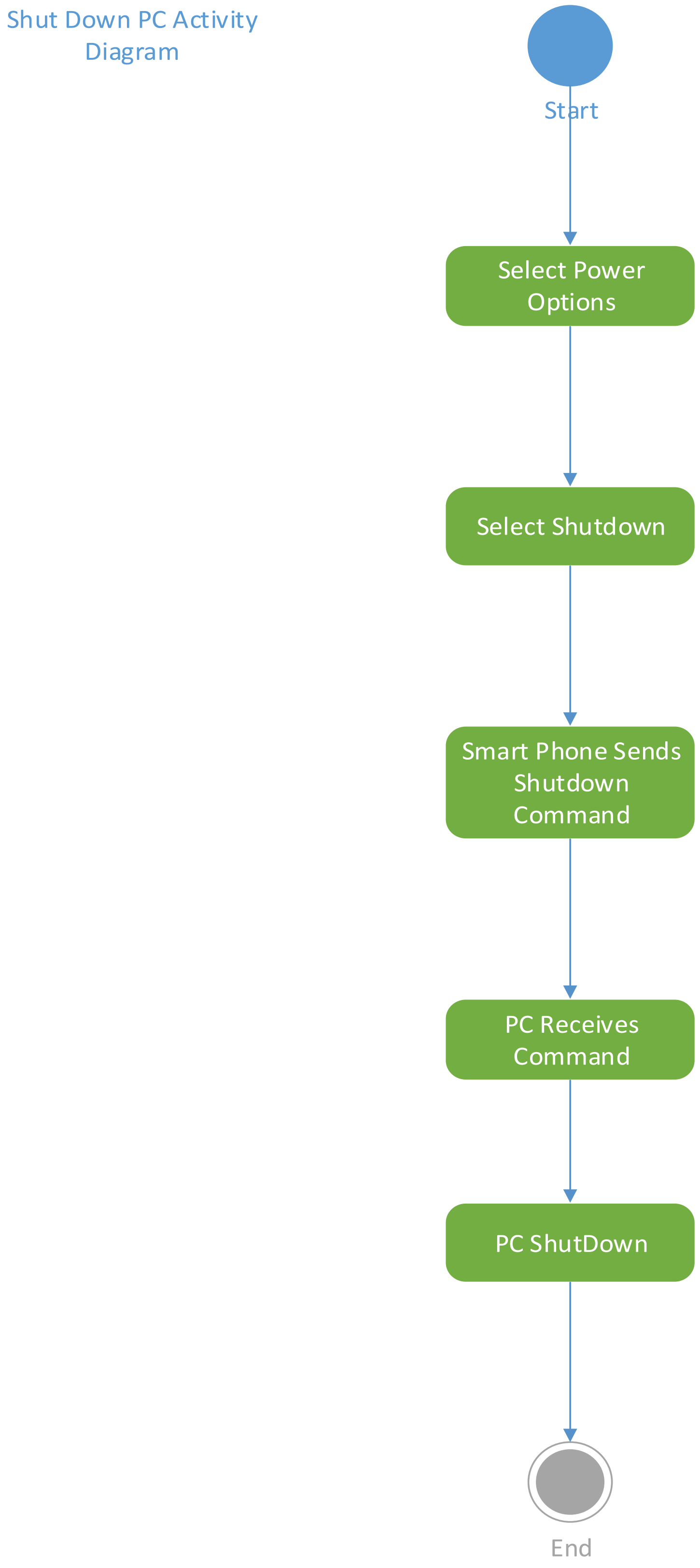
15. Increase Brightness Activity Diagram.


Increase Brightness Activity Diagram
Start
Move Brightness Volume Scroll Bar UP
Send Command To PC
PC Recieves Command
PC Increases the Brightness Volume
End

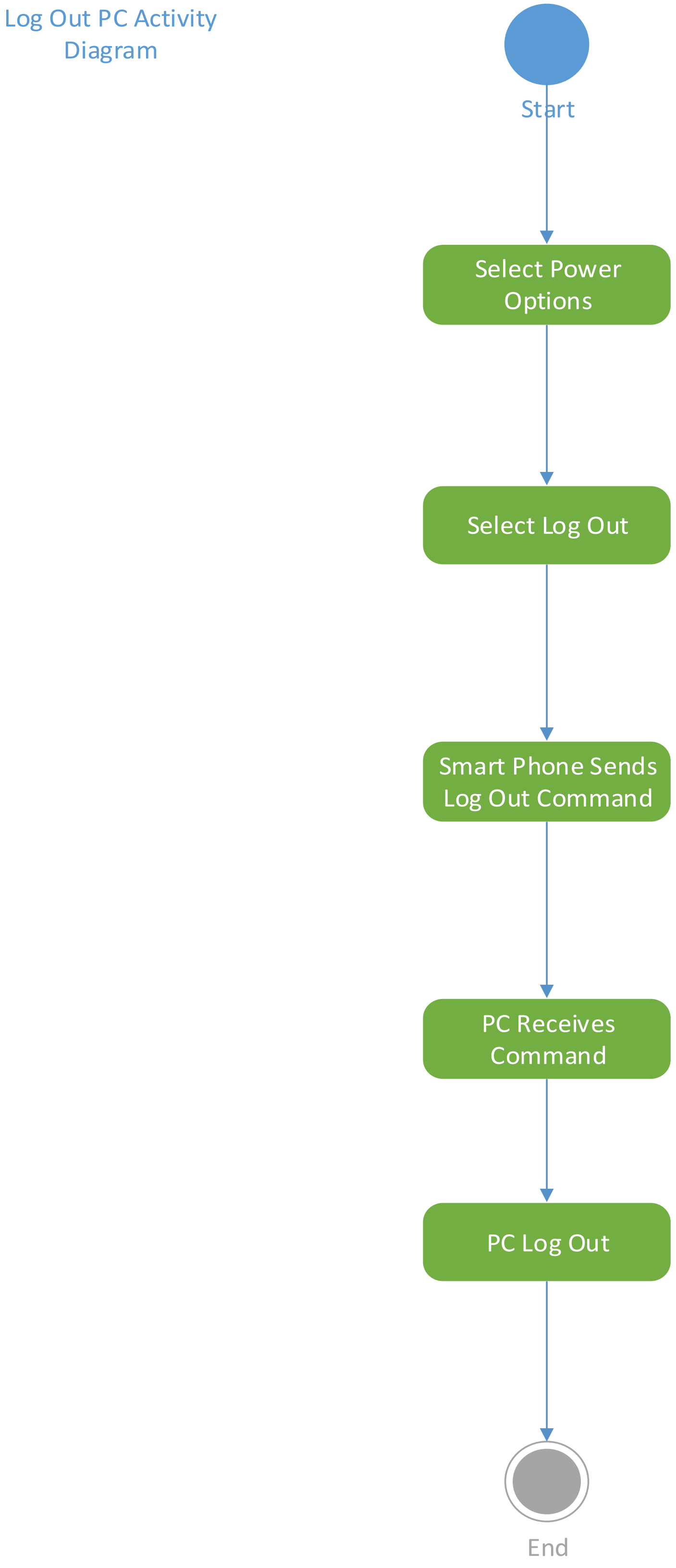
16. Decrease Brightness Activity Diagram.


Decrease Brightness Activity Diagram
Start
Move Brightness Scroll Bar Down
Send Command To PC
PC Recieves Command
PC Decreases the Brightness
End

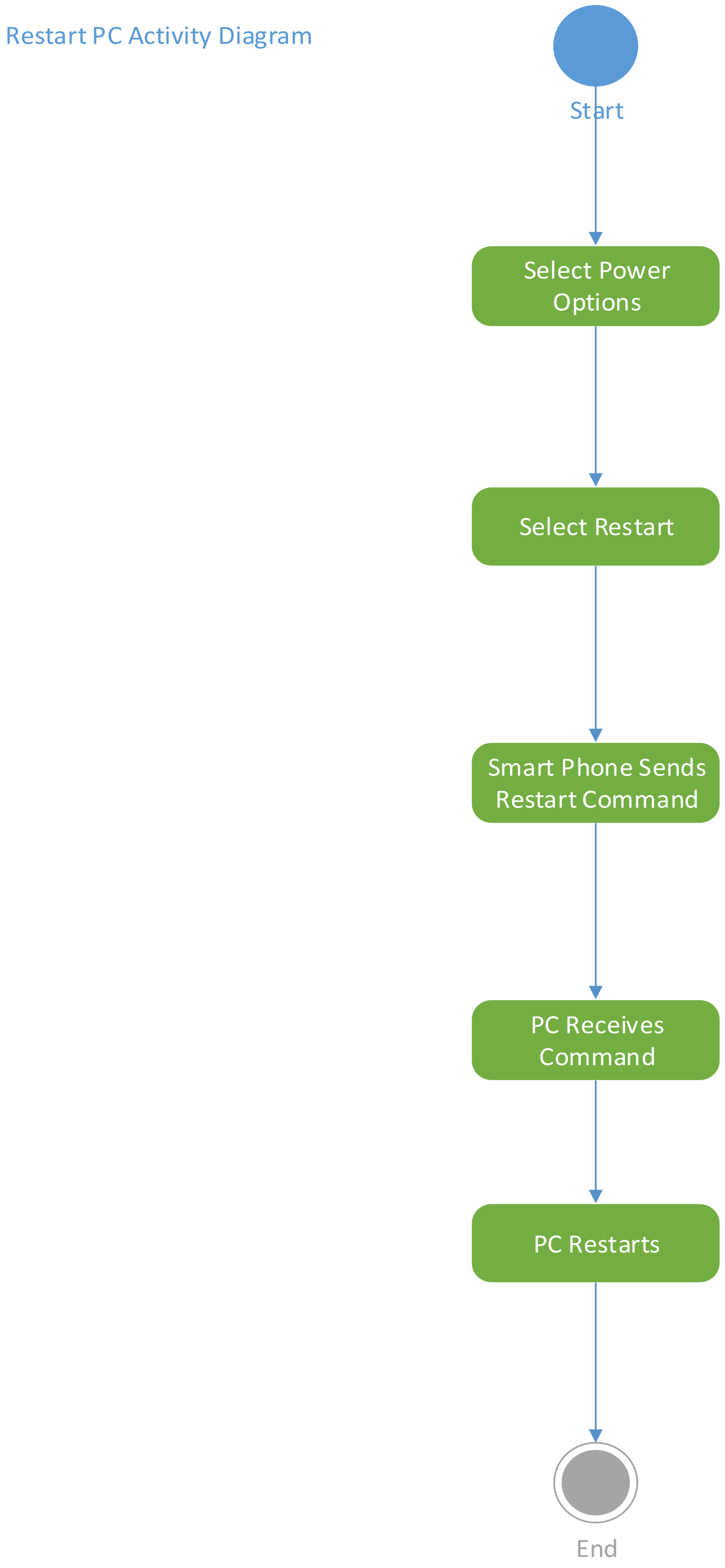
17. Shut Down PC Activity Diagram.



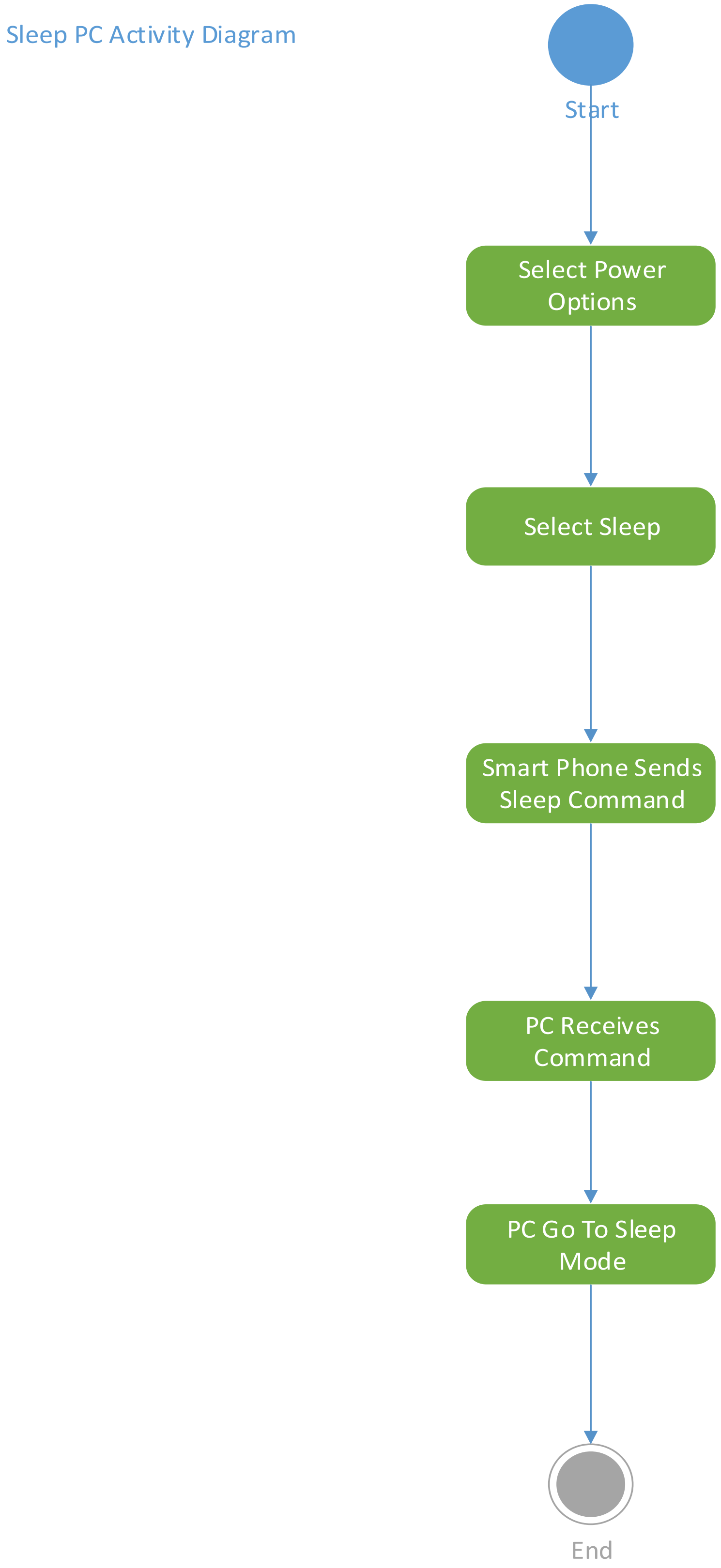
18. Log out PC Activity Diagram.



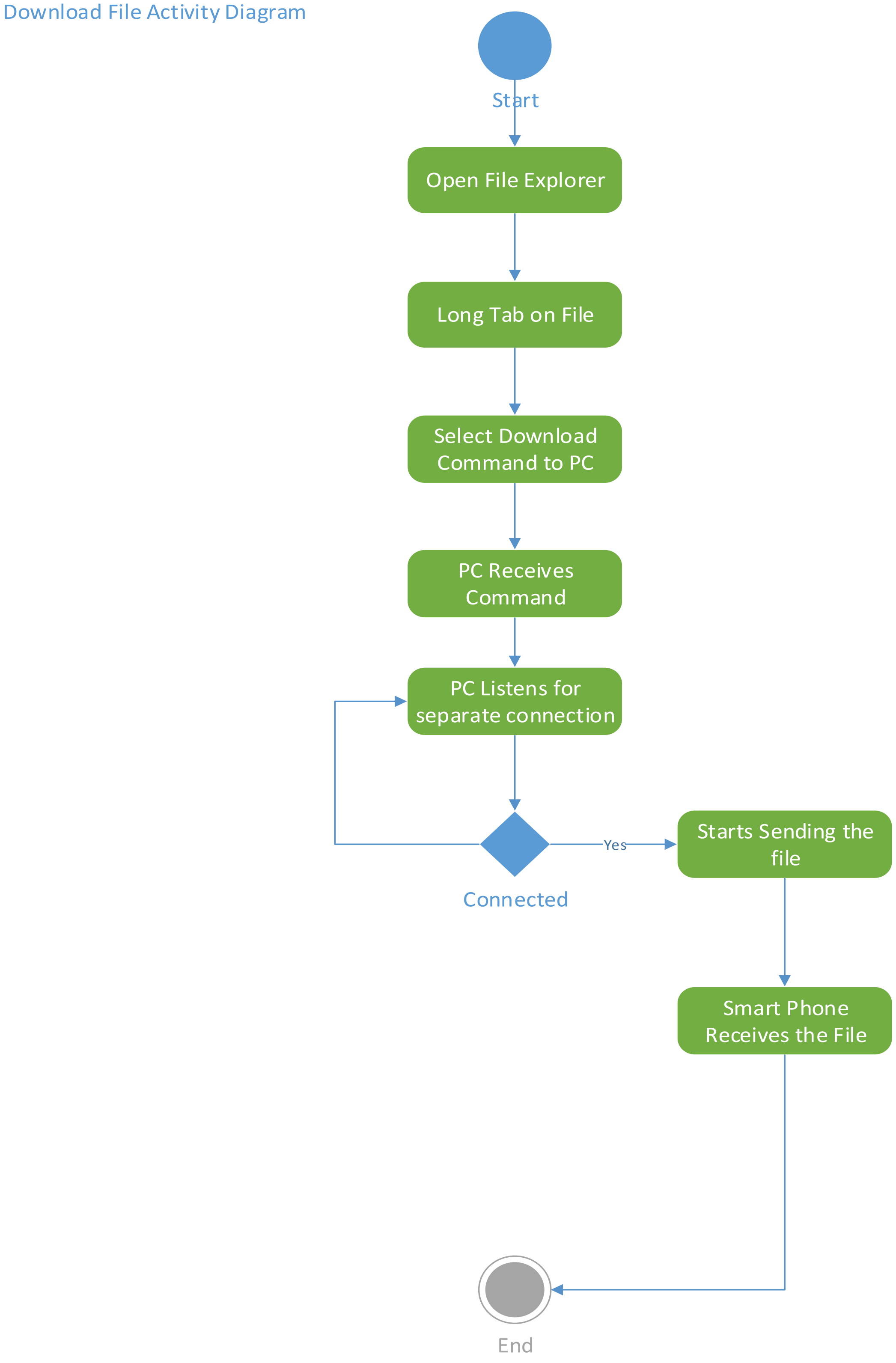
19. Restart PC Activity Diagram.



20. Sleep PC Activity Diagram.



21. Download File Activity Diagram.



Download File Activity Diagram
Start
Open File Explorer
Long Tab on File
Select Download Command to PC
PC Receives Command
PC Listens for separate connection
Connected
Yes
Starts Sending the file
Smart Phone Receives the File
End

Chapter 4: Design.

1. Overview

1. Objects-Relational Model.

2. Designing State Behavioral Modeling.

3. Subsystem Design.

4. Class and Object Design.

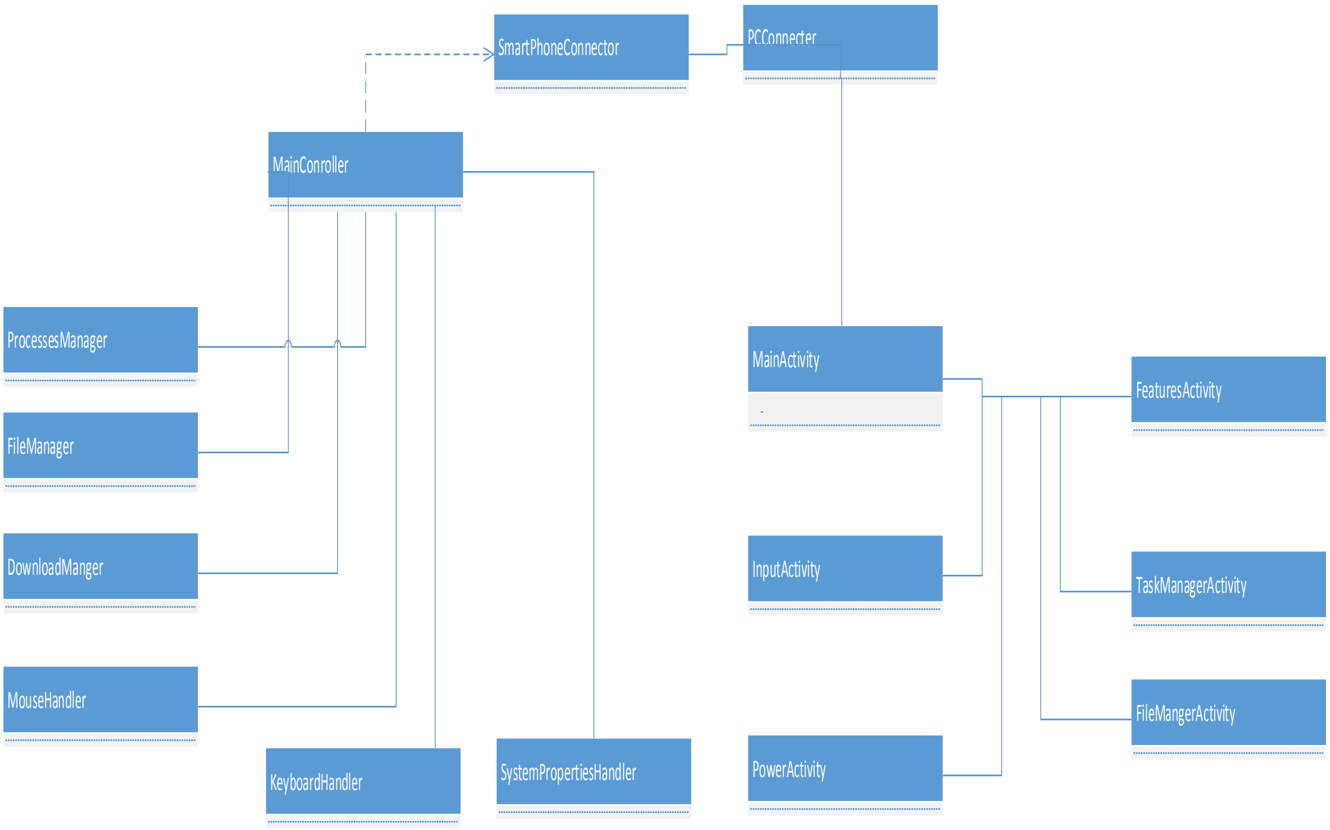
5. Object Interfacing.

4.1 Overview.

This chapter is the design chapter of the graduation project, and will propose the objects relational model, the state behavioral modeling, the subsystems design, and class object design, and the object interfaces and the User Interface.

4.2. Objects Relational Model.

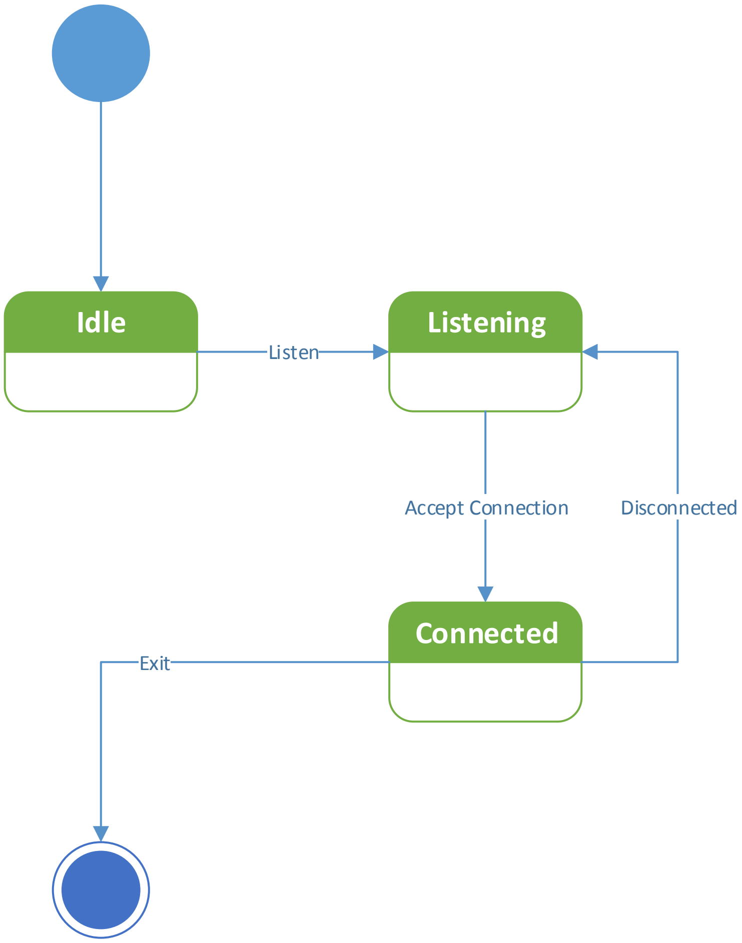
The following UML diagram represents the UML objects relation model for the PC and the smart phone together with a connection.



4.3. State Behavioral Modeling.

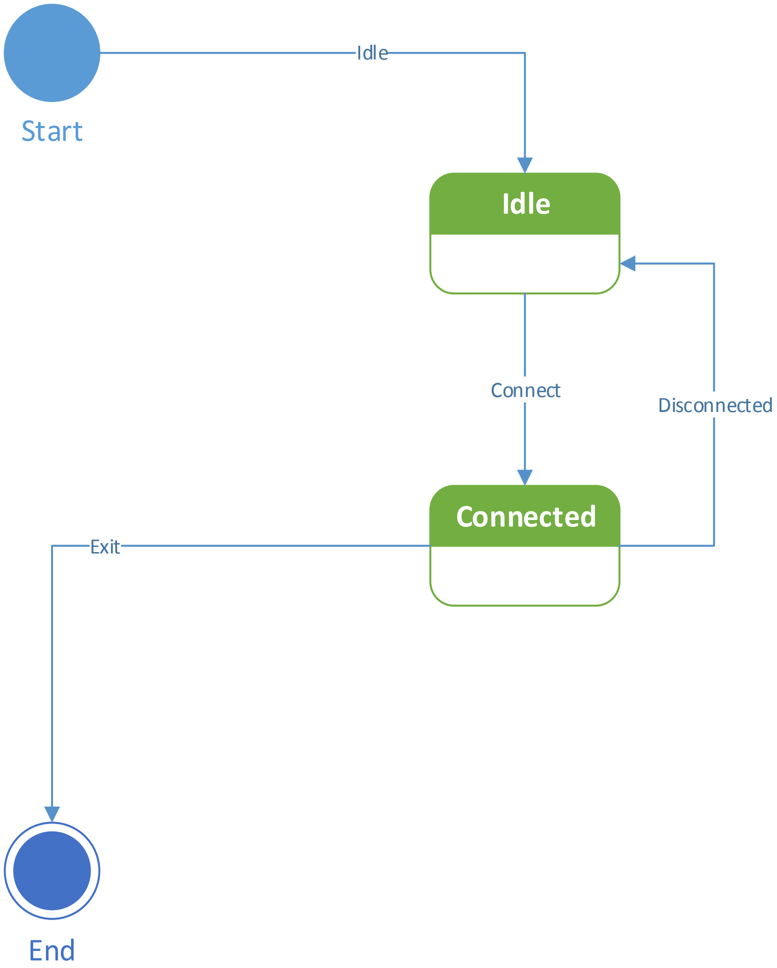
In this section the behavioral state diagrams are shown to show the connection state for the PC and the android application.

1. PC Behavioral state model.
   This diagram shows the state of the server when listening on a specific port for TCP connection.



2. Android Application Behavioral State diagram.

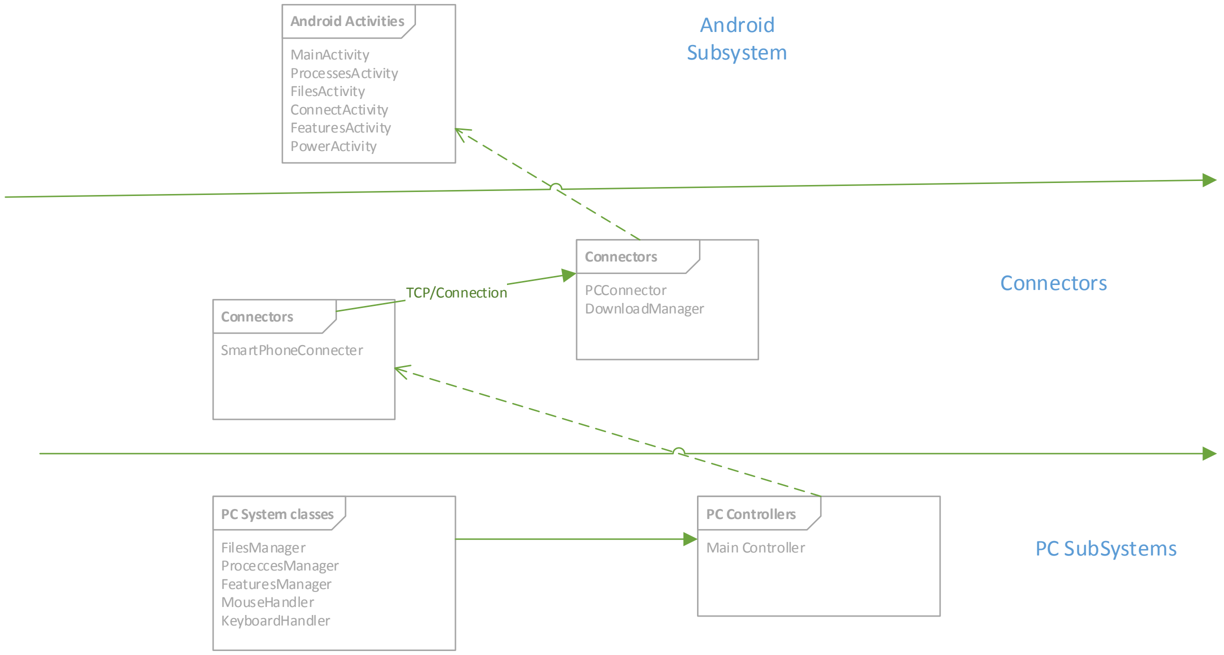
This state diagram shows the state of the connection in the android application when establishing TCP connection.



4.3. Subsystem Design.

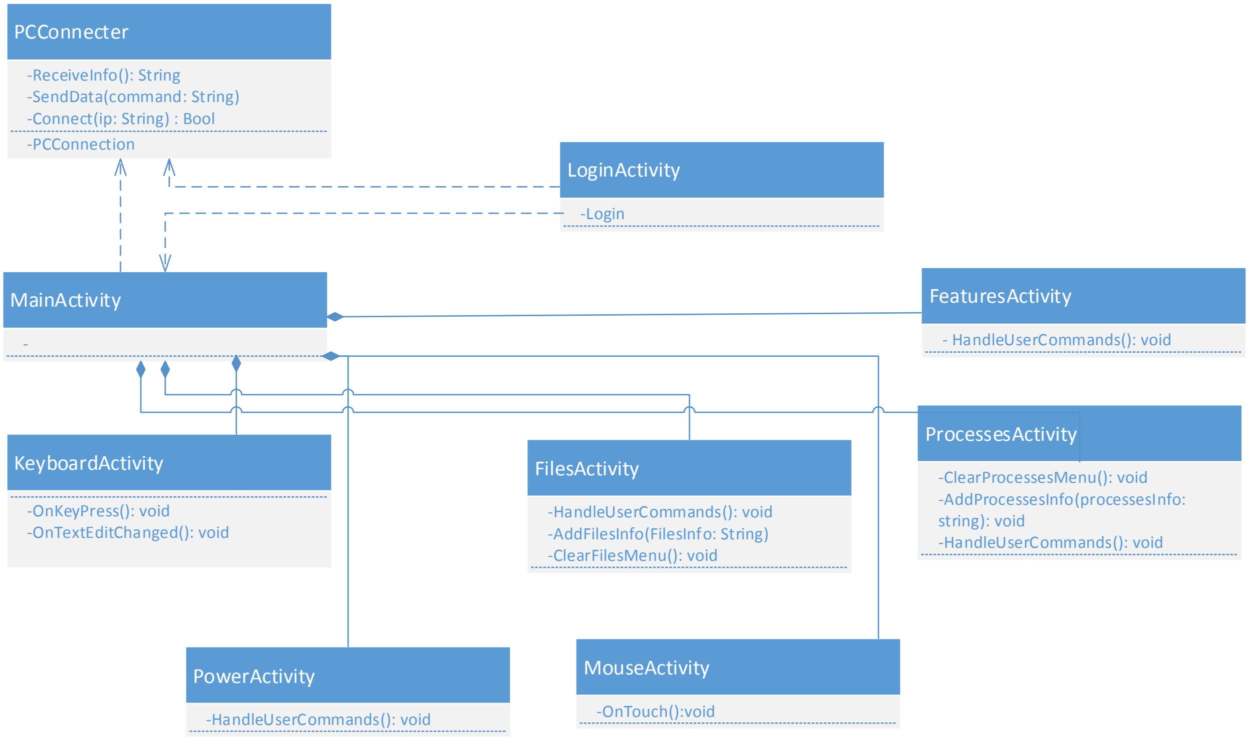
The following diagram shows the subsystems design in the project in the server and the client,

The server side has the classes that control and access the resources of the computer, and has either the main controller that works as a controller between the connector and the other classes.



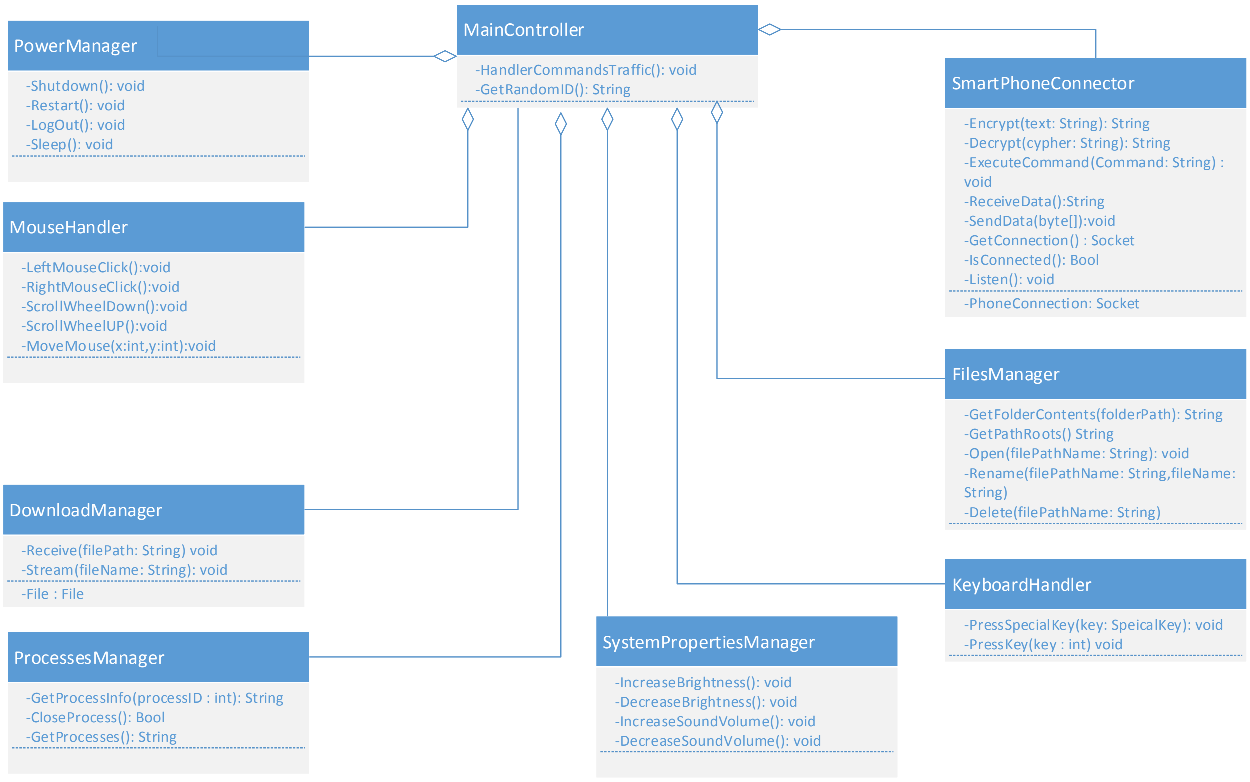
4.4 Class and Object Design.

a. In the following diagram the UML Class diagram is shown for the android application.


PCConnecter
-ReceiveInfo(): String
-SendData(command: String)
-Connect(ip: String) : Bool
-PCConnection
LoginActivity
-Login
MainActivity
-
FeaturesActivity
- HandleUserCommands(): void
KeyboardActivity
-OnKeyPress(): void
-OnTextEditChanged(): void
FilesActivity
-HandleUserCommands(): void
-AddFilesInfo(FilesInfo: String)
-ClearFilesMenu(): void
ProcessesActivity
-ClearProcessesMenu(): void
-AddProcessesInfo(processesInfo: string): void
-HandleUserCommands(): void
PowerActivity
-HandleUserCommands(): void
MouseActivity
-OnTouch():void

b. This diagram represents the UML class diagram for the PC application.

PC UML Class Diagram


MainController
-HandlerCommandsTraffic(): void
-GetRandomID(): String
PowerManager
-Shutdown(): void
-Restart(): void
-LogOut(): void
-Sleep(): void
SmartPhoneConnector
-Encrypt(text: String): String
-Decrypt(cypher: String): String
-ExecuteCommand(Command: String) : void
-ReceiveData():String
-SendData(byte[]):void
-GetConnection() : Socket
-IsConnected(): Bool
-Listen(): void
-PhoneConnection: Socket
MouseHandler
-LeftMouseClick():void
-RightMouseClick():void
-ScrollWheelDown():void
-ScrollWheelUP():void
-MoveMouse(x:int,y:int):void
FilesManager
-GetFolderContents(folderPath): String
-GetPathRoots() String
-Open(filePathName: String): void
-Rename(filePathName: String,fileName: String)
-Delete(filePathName: String)
DownloadManager
-Receive(filePath: String) void
-Stream(fileName: String): void
-File : File
KeyboardHandler
-PressSpecialKey(key: SpeicalKey): void
-PressKey(key : int) void
SystemPropertiesManager
-IncreaseBrightness(): void
-DecreaseBrightness(): void
-IncreaseSoundVolume(): void
-DecreaseSoundVolume(): void
ProcessesManager
-GetProcessInfo(processID : int): String
-CloseProcess(): Bool
-GetProcesses(): String

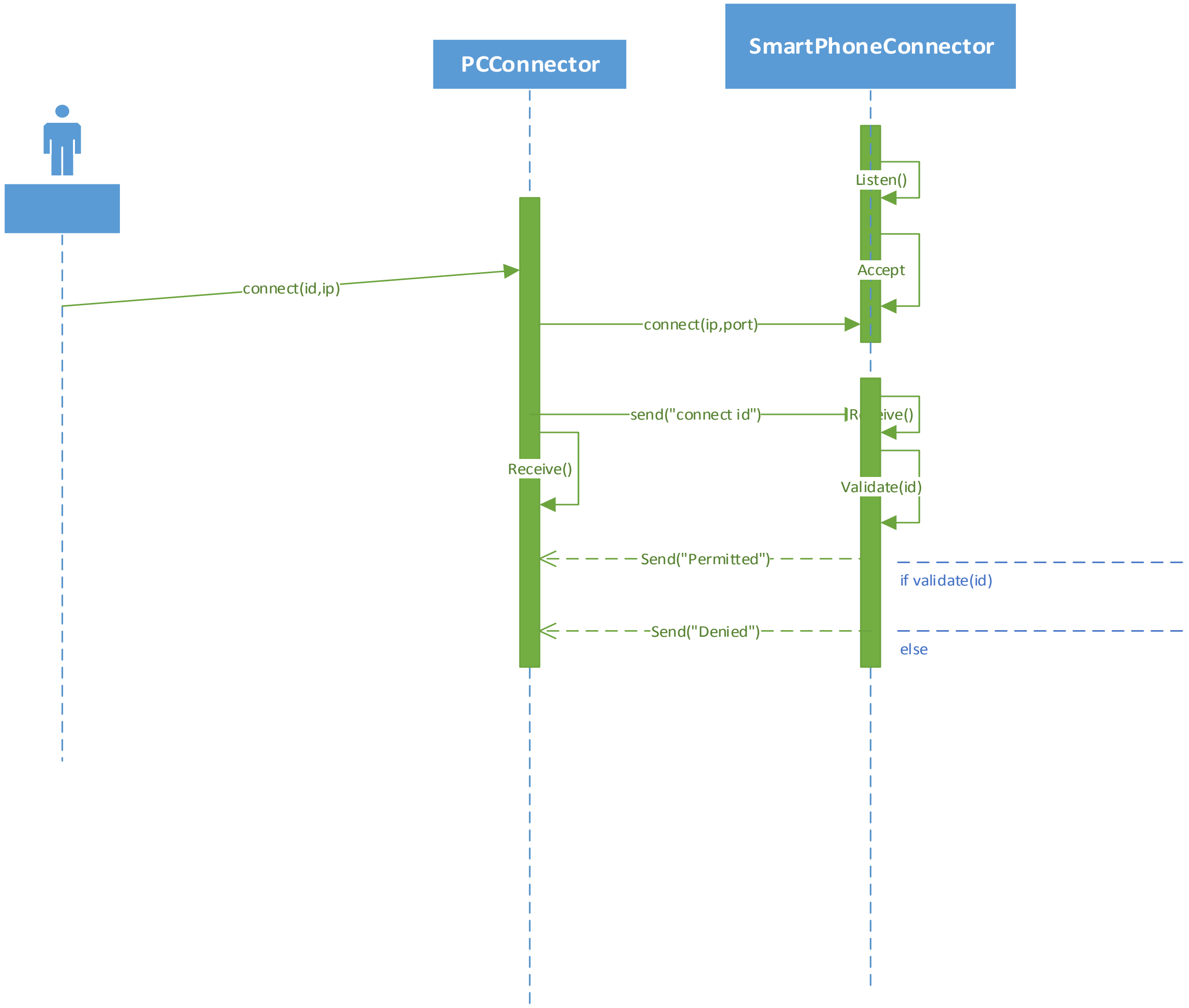
4.6 Interface (Message) Design.

In this section the sequence diagrams is shown for the use cases, and then the graphical user interface.

5.6.1 Object Interfacing.

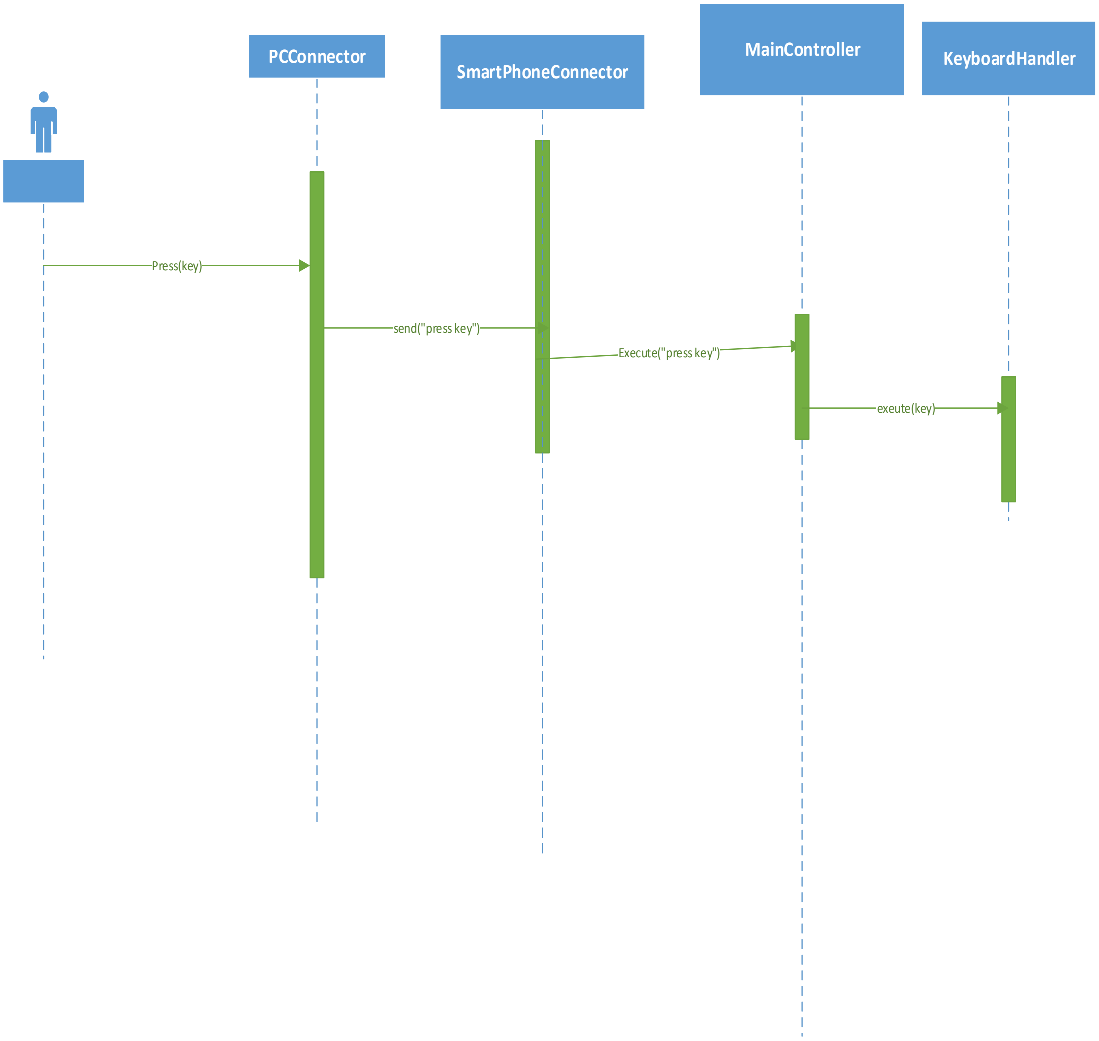
1. Connect sequence diagram.

This sequence diagram represents the connect scenario.



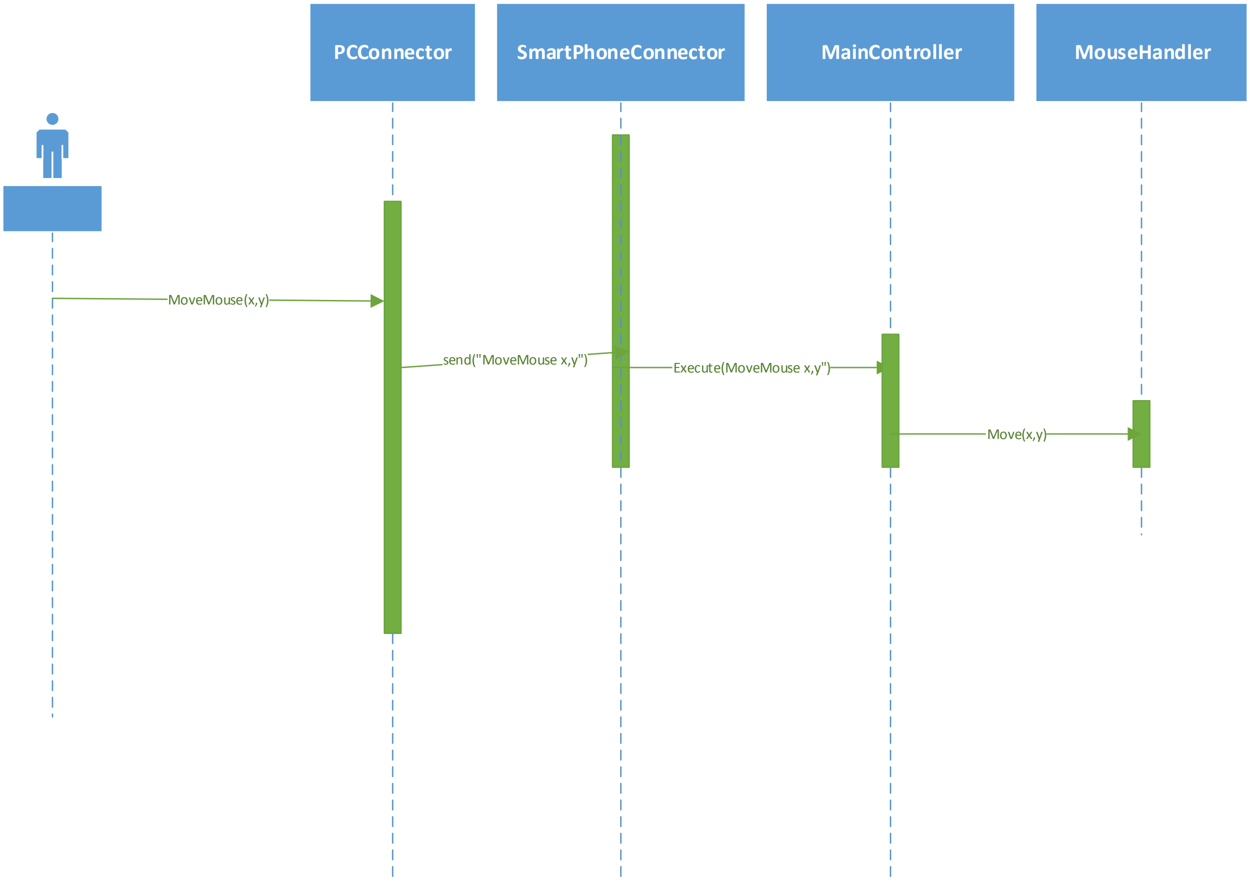
2. Keystroke sequence diagram.

This sequence diagram represents the keystroke scenario when the user uses the keyboard.



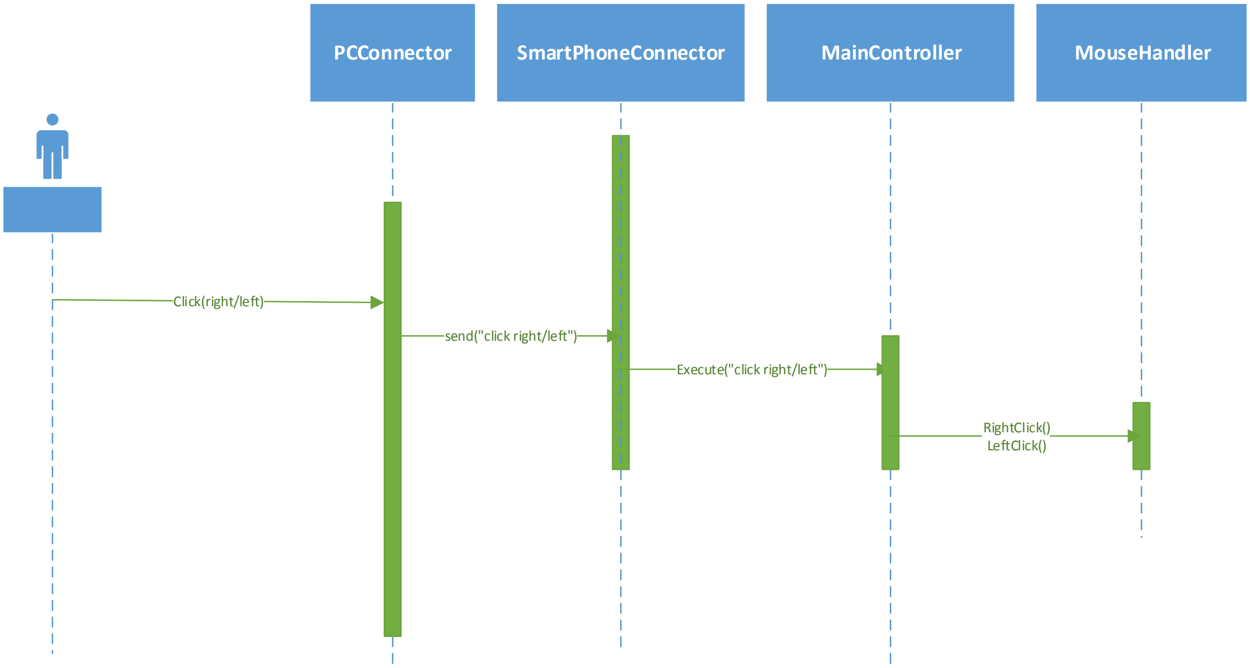
3. Use mouse sequence diagram.

This sequence diagram shows how to move a mouse cursor on the PC.



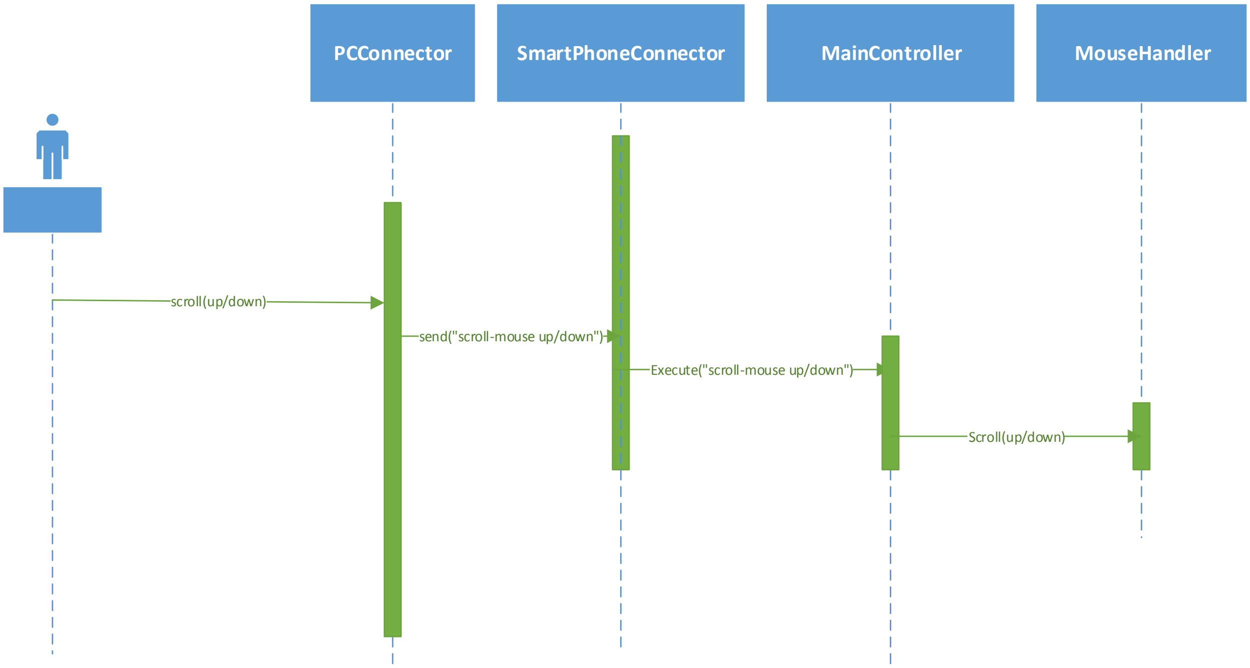
4. Left/Right mouse click sequence diagram.

In this sequence diagram the right/left click mouse is shown.



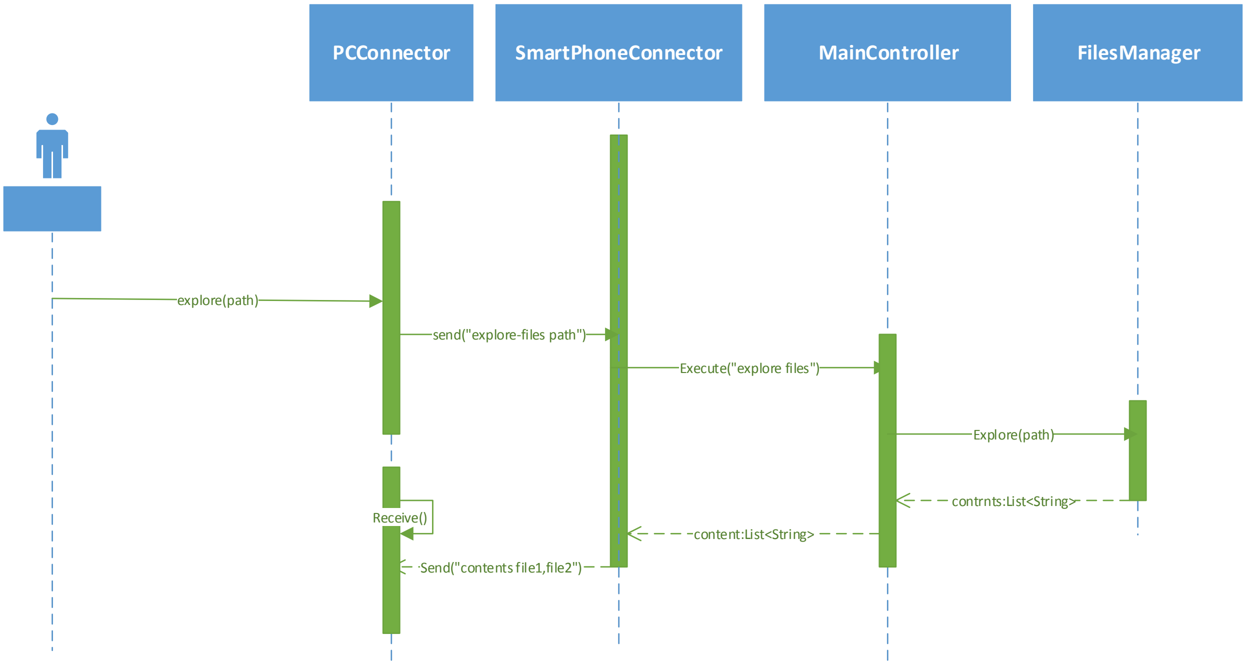
5. Mouse scroll sequence diagram.

In this sequence diagram the scroll up/down scenario is shown.

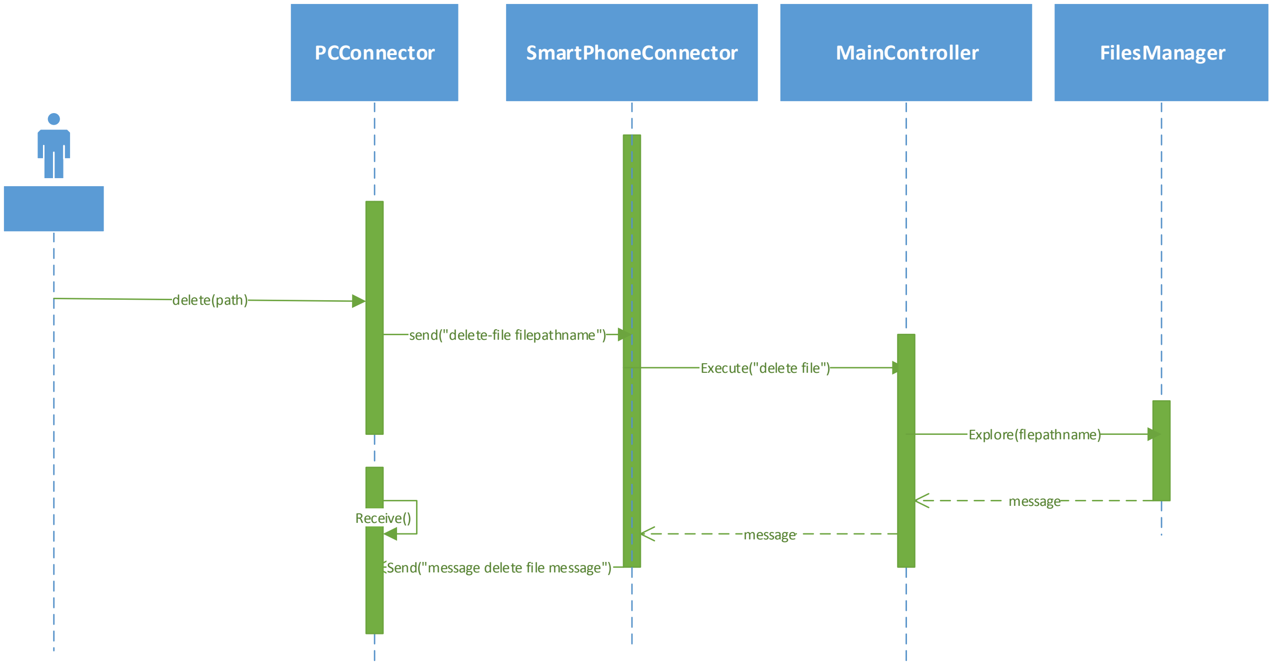


6. Explore files sequence diagram.

In this sequence diagram the explore files scenario is shown.

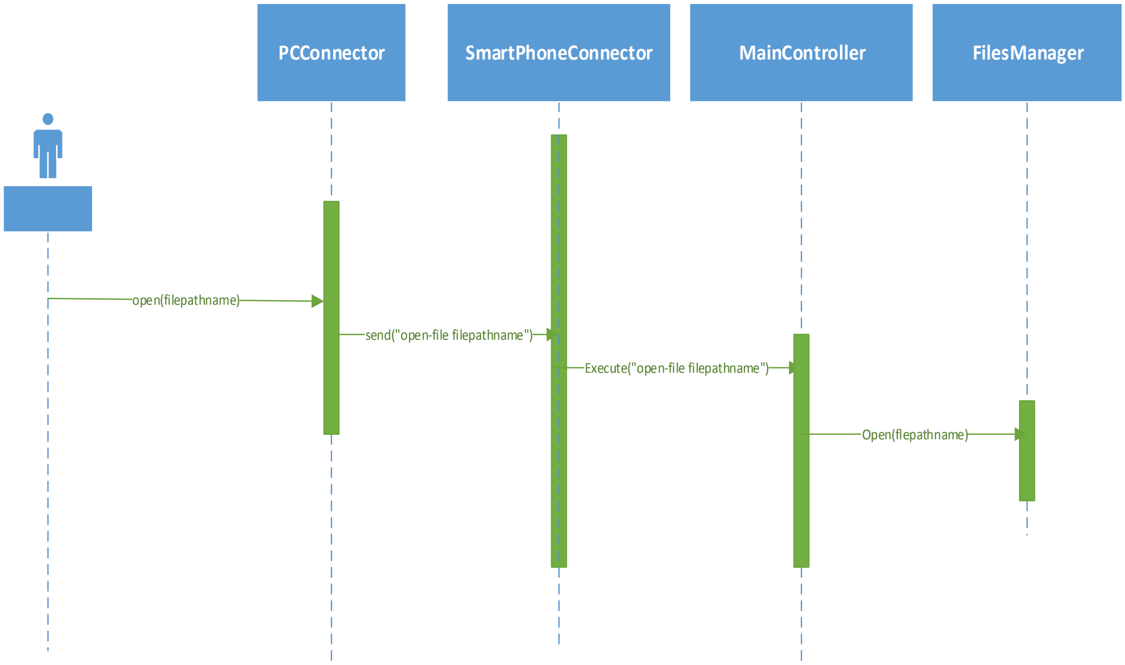


7. Delete file sequence



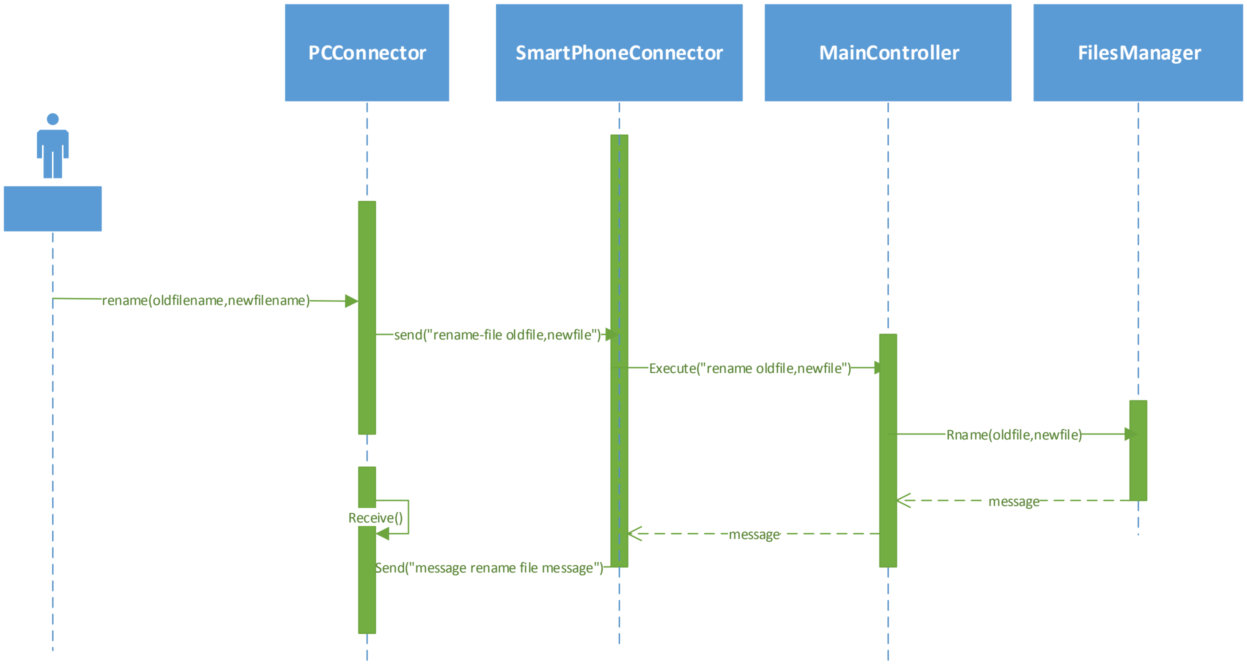
8. Open file sequence diagram.

In this sequence diagram the open file scenario is shown.



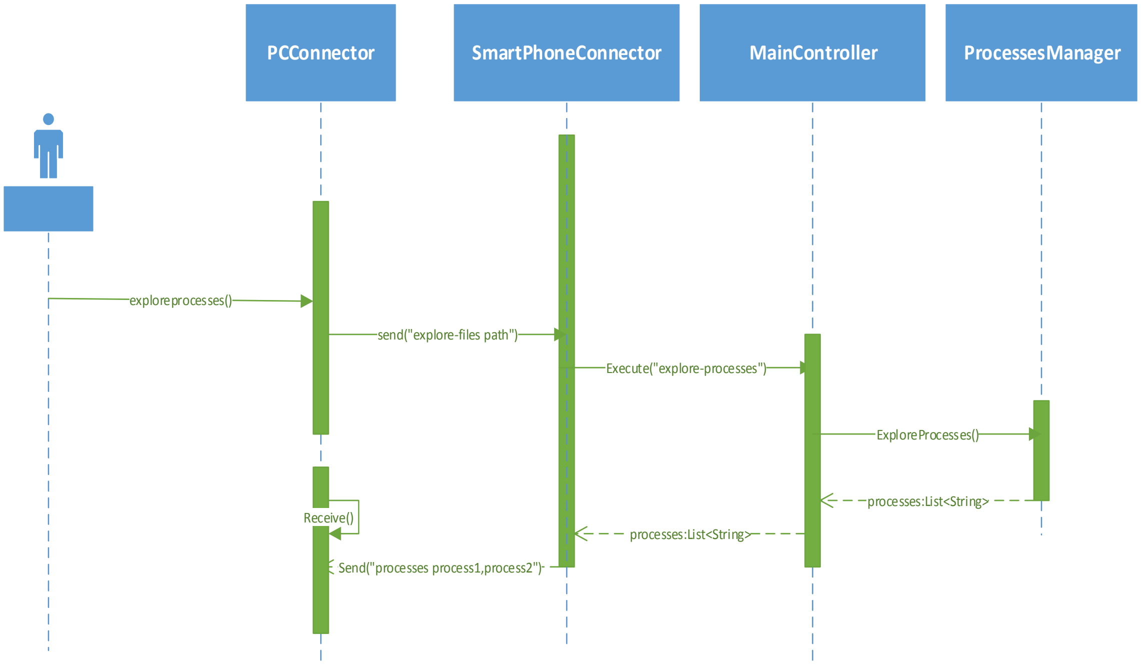
9. Rename file sequence diagram.

In this sequence diagram the open file scenario is shown.


PCConnector
SmartPhoneConnector
MainController
FilesManager
rename(oldfilename,newfilename)
send("rename-file oldfile,newfile")
Execute("rename oldfile,newfile")
Rname(oldfile,newfile)
message
message
Receive()
Send("message rename file message")

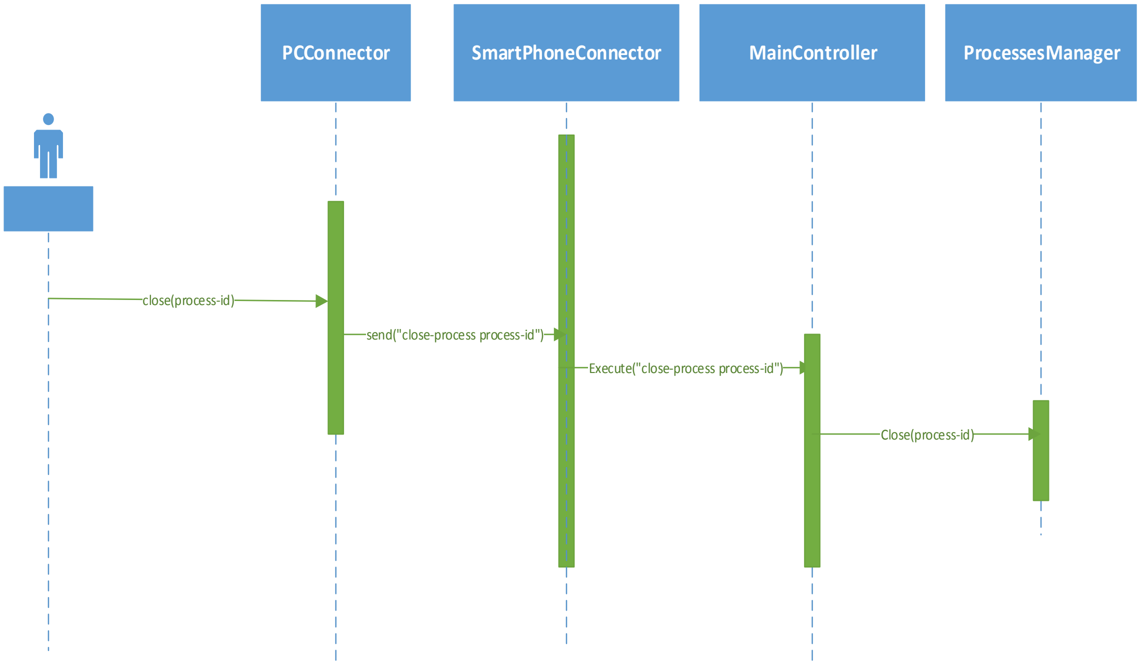
10. Explore processes sequence diagram.

In this sequence diagram the explore processes scenario is shown.



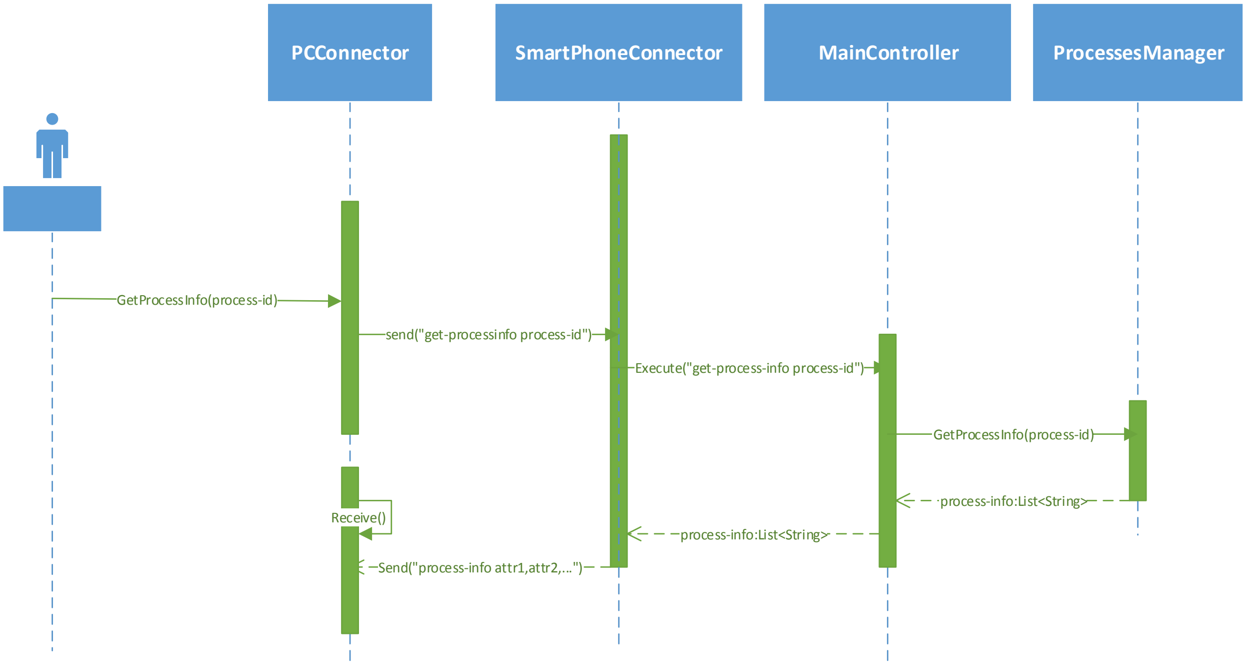
11. Close process sequence diagram.

In this sequence diagram the close process scenario is shown.



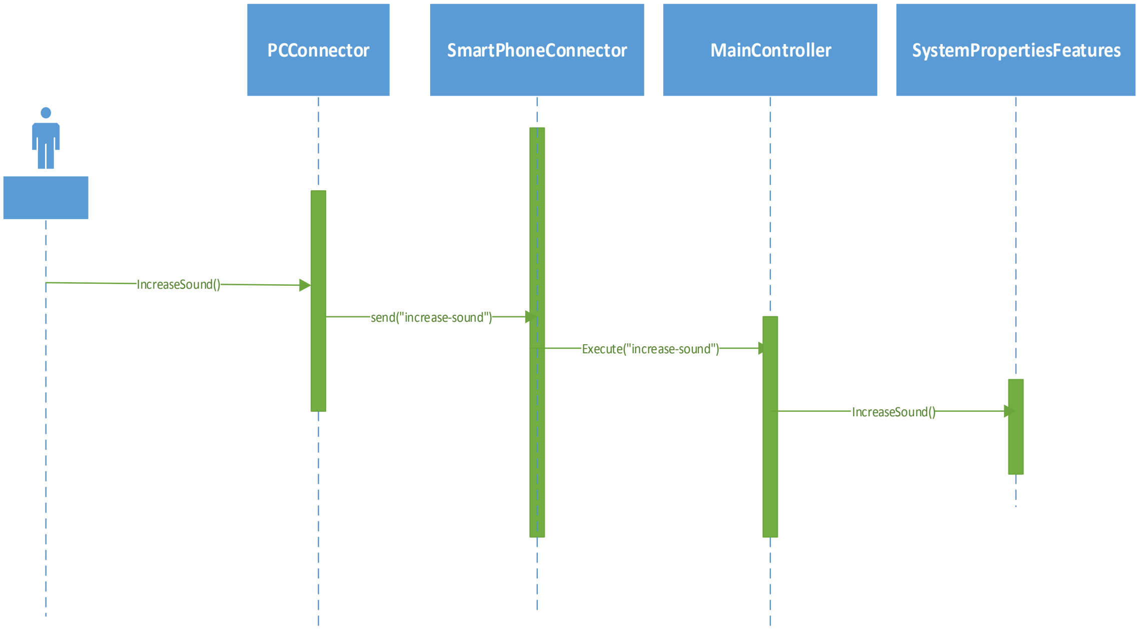
12. Show process info sequence diagram.

In this sequence diagram the get process info scenario is shown.


PCConnector
SmartPhoneConnector
MainController
ProcessesManager
GetProcessInfo(process-id)
send("get-processinfo process-id")
Execute("get-process-info process-id")
GetProcessInfo(process-id)
process-info:List<String>
process-info:List<String>
Receive()
Send("process-info attr1,attr2,...")

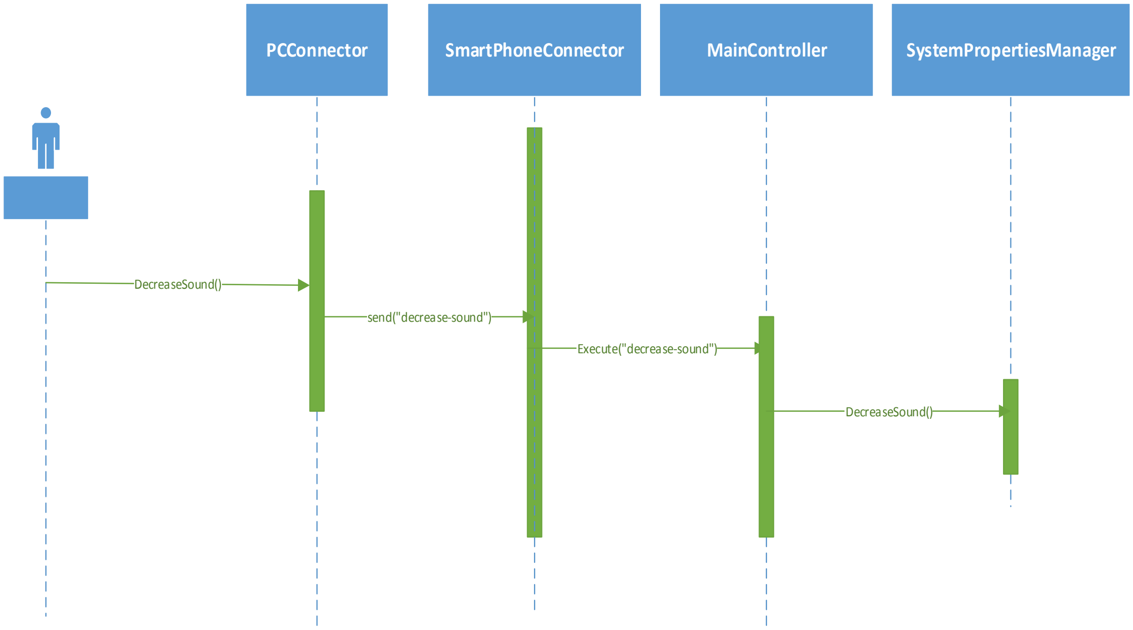
13. Increase sound volume sequence diagram.

In this sequence diagram the increase sound volume scenario is shown.



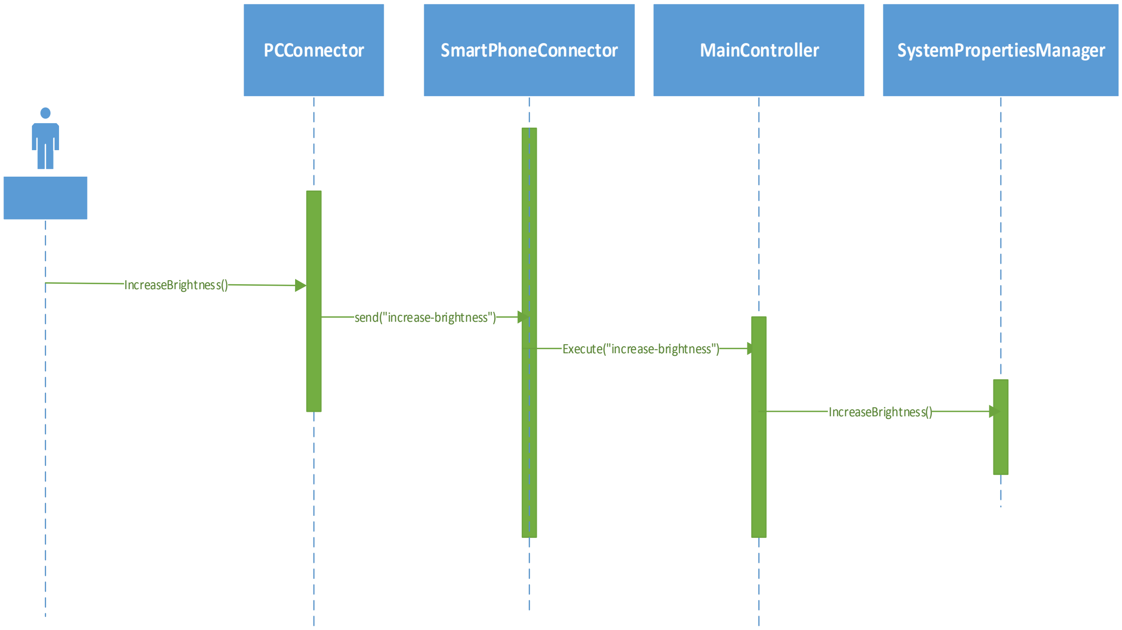
14. Decrease sound volume sequence diagram.

In this sequence diagram the decrease sound volume scenario is shown.



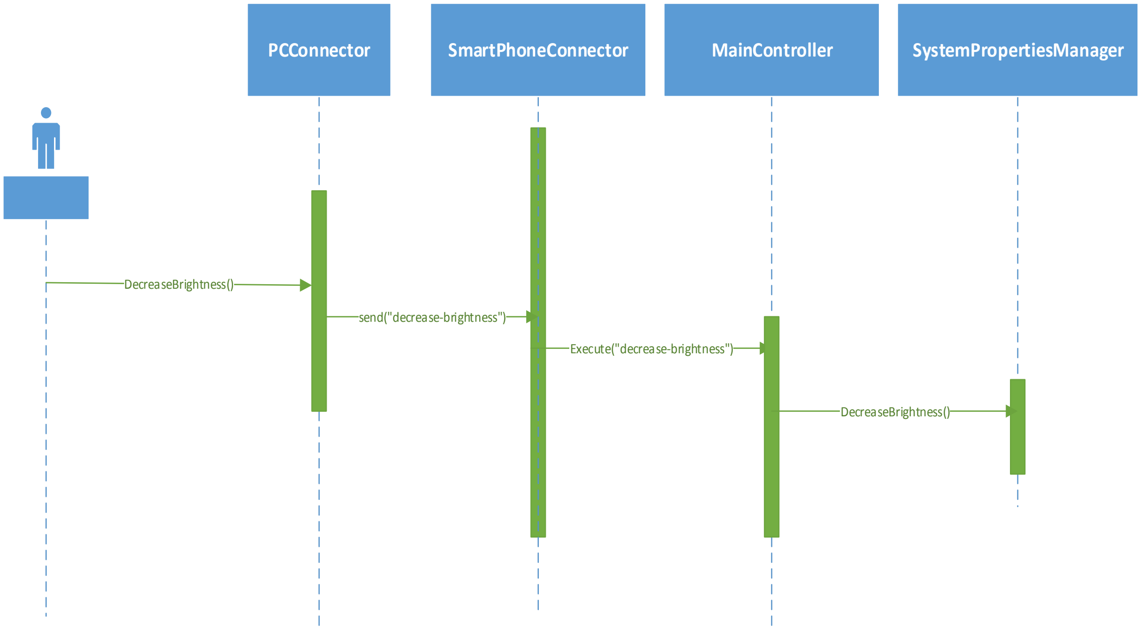
15. Increase brightness volume sequence diagram.

In this sequence diagram the increase brightness volume scenario is shown.



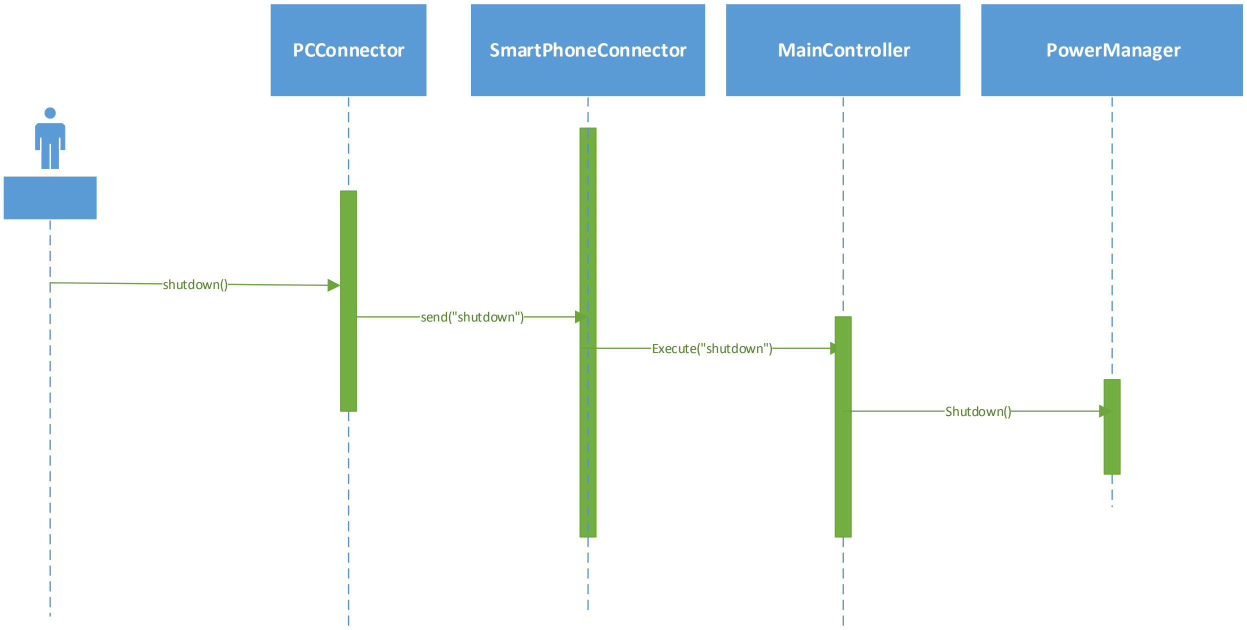
16. Decrease brightness volume sequence diagram.

In this sequence diagram the decrease brightness volume scenario is shown.



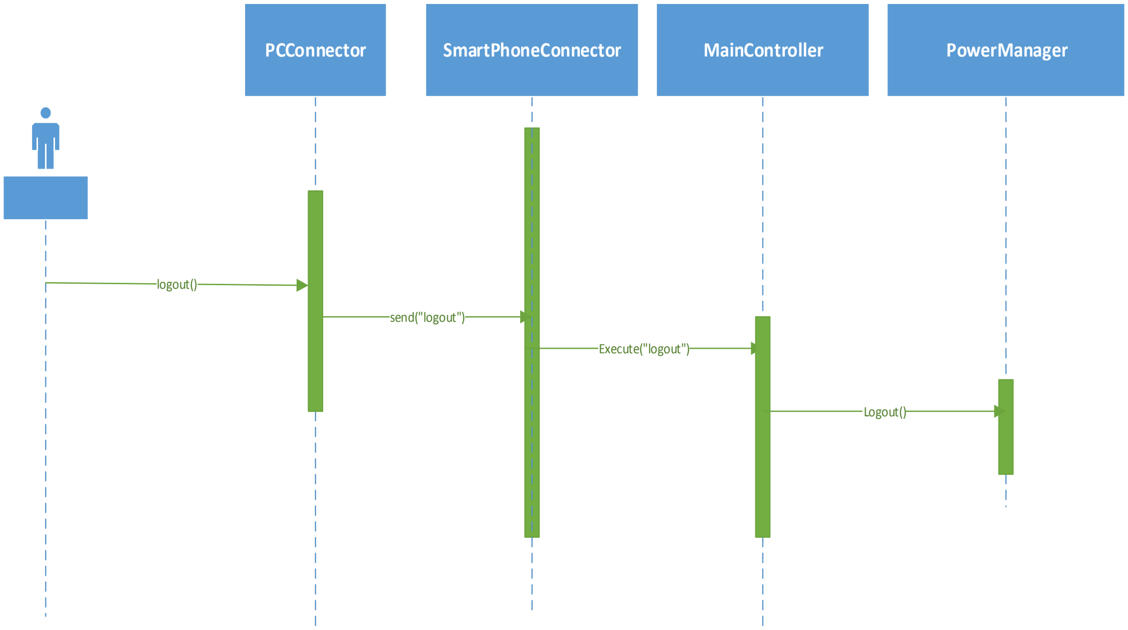
17. Shutdown PC sequence diagram.

In this sequence diagram shutdown scenario is shown.



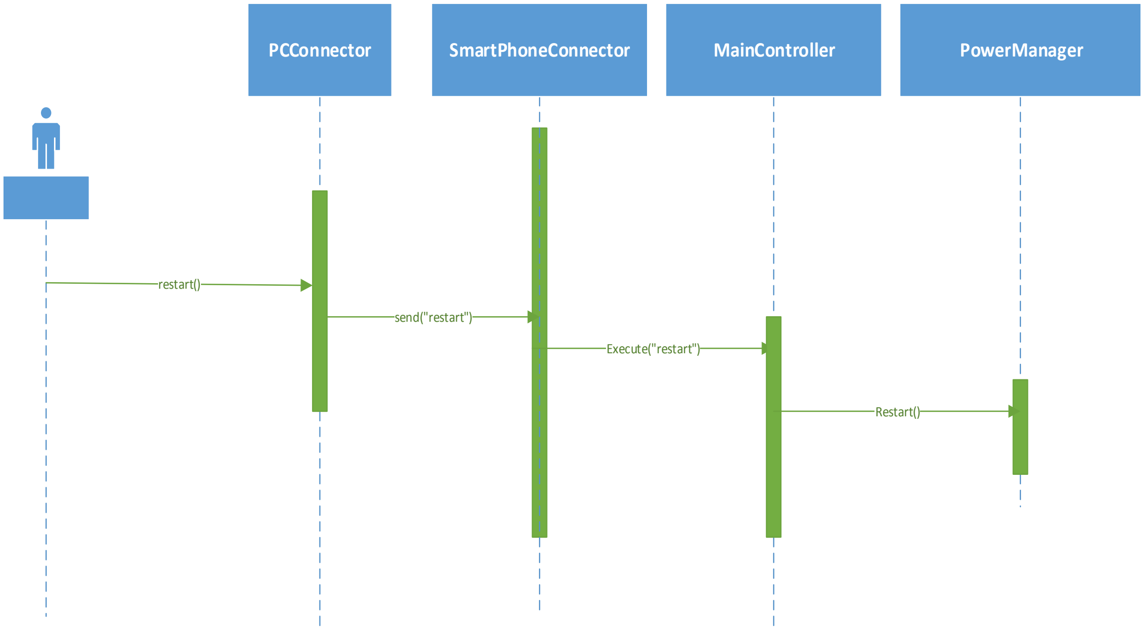
18. Logout from PC sequence diagram.

In this sequence diagram logout from PC scenario is shown.



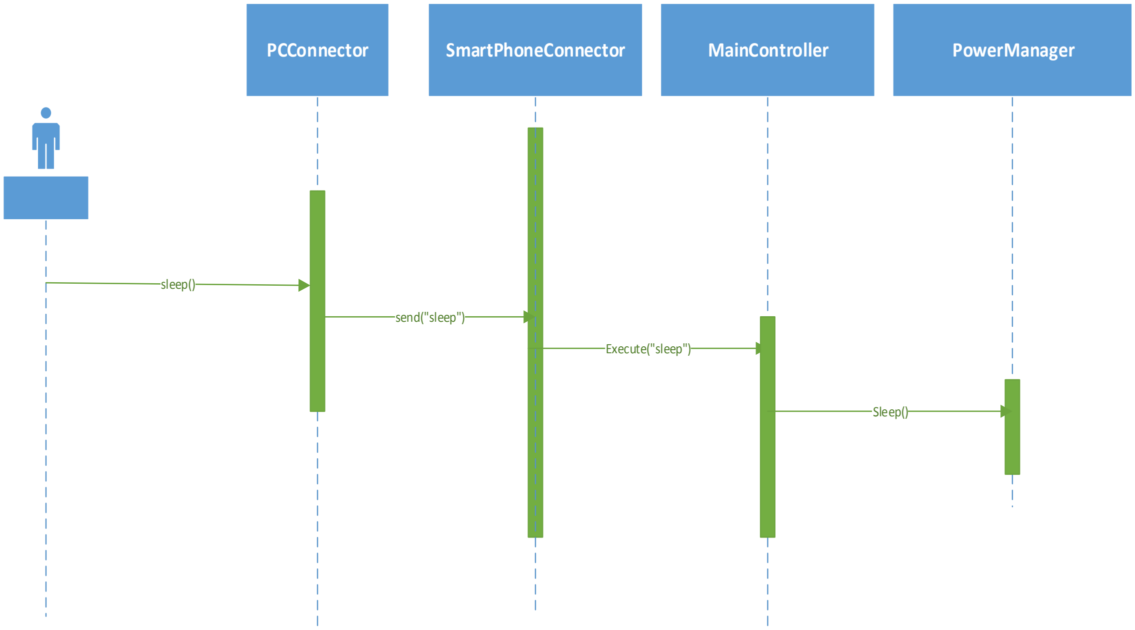
19. Restart the PC sequence diagram.

In this sequence diagram restart PC scenario is shown.



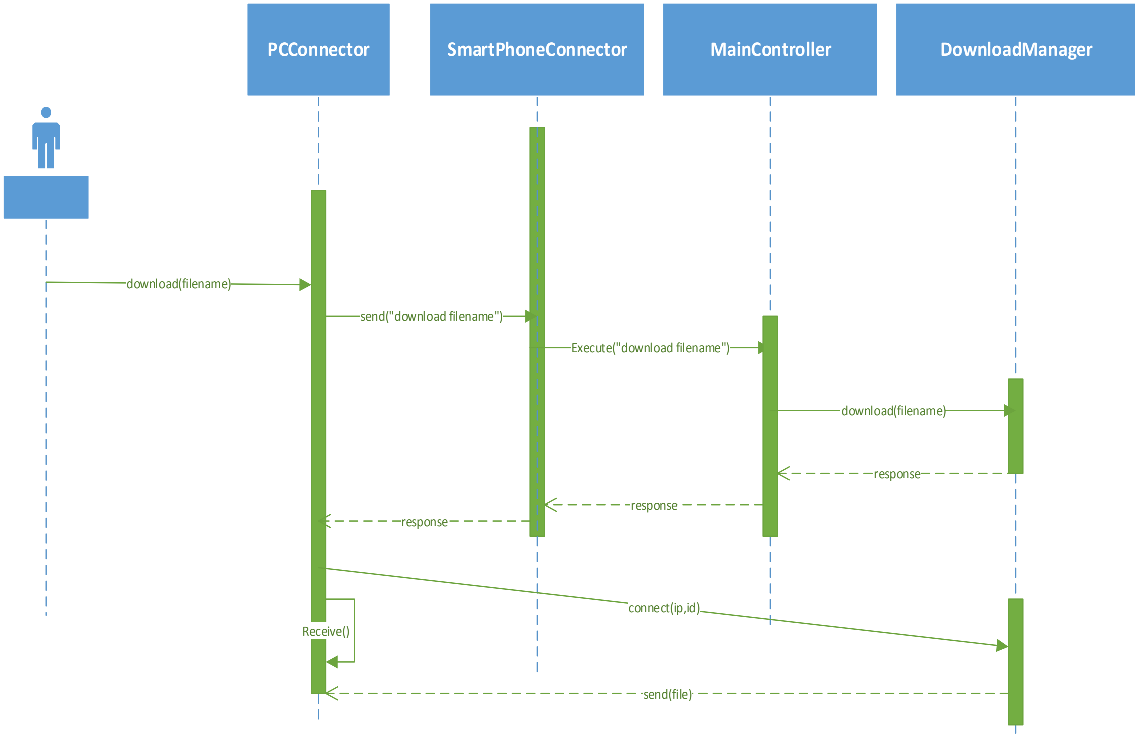
20. PC sleep sequence diagram.

In this sequence diagram PC sleep scenario is shown.



21. File download sequence diagram.

In this sequence diagram file download scenario is shown.



Chapter 5: Implementation.

1. Brief description about chosen programs (tools).
2. Problems occurred during implantation and solutions.
3. Selected code for necessary algorithms with little description.
4. Snap shots for selected screens.

5.1 Overview.

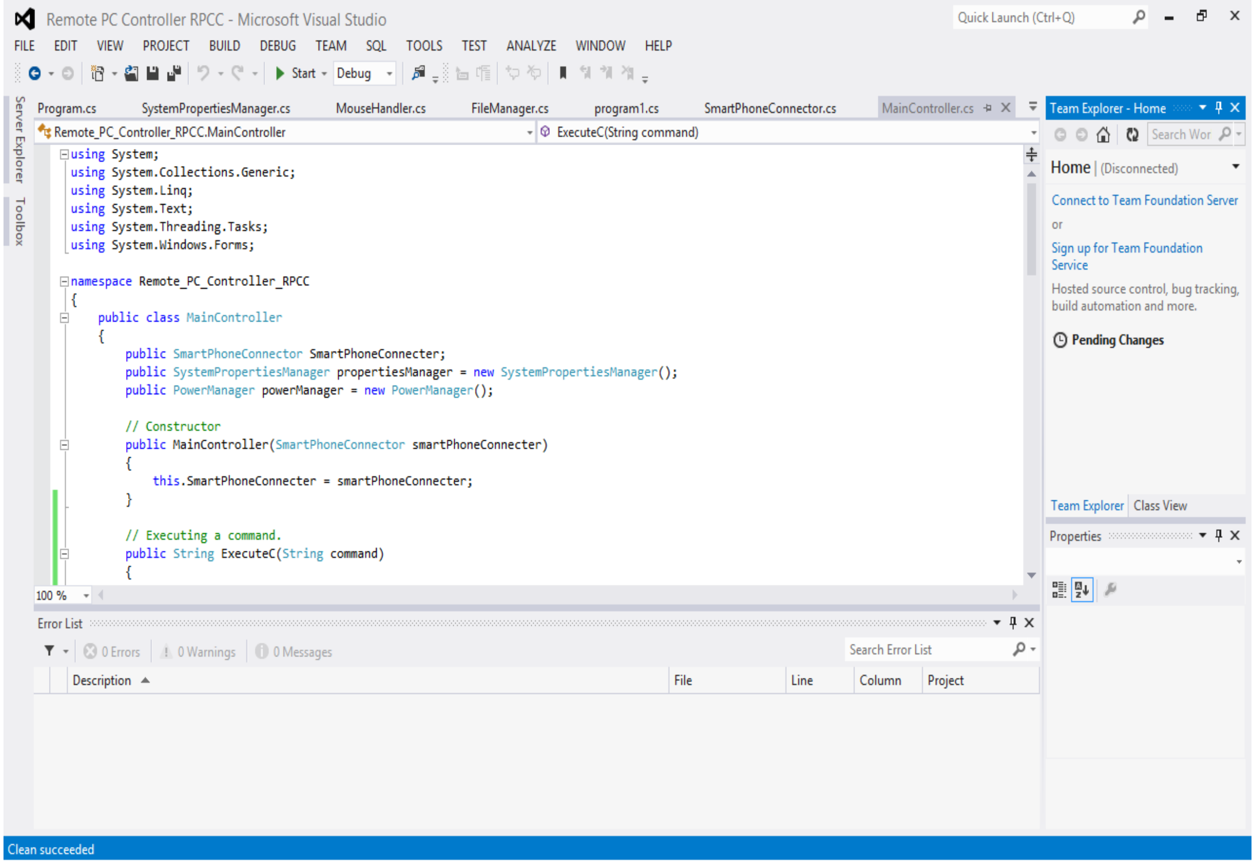
In this chapter the implementation part of the project will be shown, that contains brief description of the tools that used in the application, and the problems occurred during the implementation, and selected code for necessary algorithms, and finally some snapshots for the screens.

5.2 Brief description about chosen programs (tools).

The tools used to develop this project as planed and mentioned in the planning phase, are:

1. Visual Studio 2012.

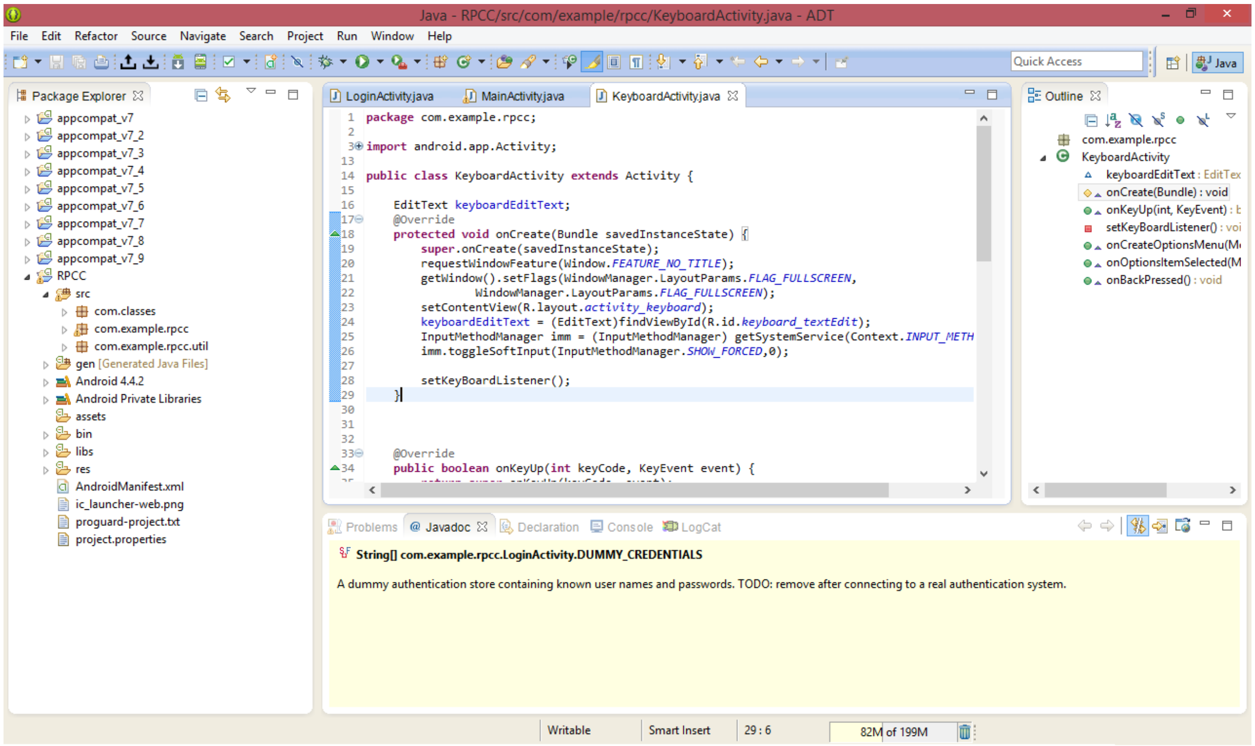
The Visual Studio Profissional 2012 is used to develop the server side application that exists on the PC, using C# language.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Remote_PC_Controller_RPCC
{
    public class MainController
    {
        public SmartPhoneConnector SmartPhoneConnecter;
        public SystemPropertiesManager propertiesManager = new SystemPropertiesManager();
        public PowerManager powerManager = new PowerManager();

        // Constructor
        public MainController(SmartPhoneConnector smartPhoneConnecter)
        {
            this.SmartPhoneConnecter = smartPhoneConnecter;
        }

        // Executing a command.
        public String ExecuteC(String command)
        {
```

2. Eclipse for android.

Eclipse for android with ADT (Android Development Tools) used to implement the application on the smart phone using java, and XML to develop the UI.



```java
package com.example.rpcc;

import android.app.Activity;

public class KeyboardActivity extends Activity {

    EditText keyboardEditText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_keyboard);
        keyboardEditText = (EditText)findViewById(R.id.keyboard_textEdit);
        InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METH
        imm.toggleSoftInput(InputMethodManager.SHOW_FORCED,0);

        setKeyBoardListener();
    }



    @Override
    public boolean onKeyUp(int keyCode, KeyEvent event) {
```

String[] com.example.rpcc.LoginActivity.DUMMY_CREDENTIALS

A dummy authentication store containing known user names and passwords. TODO: remove after connecting to a real authentication system.

5.3 Problems occurred during implantation and solutions.

Many problems occurred during the development process and they are described below:

1. Debugging the networking classes when sending the commands to be executed, the commands contained unprintable characters after sending them and that made a problem when printing it to see the string variables and when comparing the strings with the appropriate command string to result false, the problem solved after long time and many tries.
   The solution was using char class and IsControl method to check each string character by character within a loop to remove the unprintable characters to be able to compare them with the wanted strings.

2. Testing the power options, when the power options tested each time we call the shutdown, logout and restart while debugging the computer executes them and closes everything and exits without saving the code, so a virtual machine used to VMWare to solve it, so we don't have to wait the whole computer to be restarted or to shut down when we test the application.

3. Downloading a file feature made a problem because it used the same connection to send the commands and to send a file, so a new separate connection established to transfer the file without mixing the commands with the binary data.

5.4. Selected code for necessary algorithms with little description.

In this section some of the necessary code with explanation will be shown:

In the server side:

1. Initiating sockets and Listening on the port.
2. Accepting the connections and verifying the ID.
3. Removing the unprintable characters.
4. Receiving commands.
5. Handling commands.

In the SmartPhoneConnector:

In the constructor the class initiates the connection and instantiates the MainController, and then accepting the connections.

```
public SmartPhoneConnector()
{
     InitConnections();
     MainController = new MainController(this);
     AcceptConnections();
}
```

This method initializes the passive socket and bind it to the port 1600, and then listens for connections.

```
public void InitConnections()
{
     try
     {
       PassiveSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
       ProtocolType.Tcp);
       PassiveSocket.Bind(new IPEndPoint(IPAddress.Loopback, 1600));
       PassiveSocket.Listen(20);
     }
     catch (Exception ex) {
       Log.print(ex.Message);
       }
 }
```

This method accepts connection and verifying the ID, if the connection permitted the class will handle the requests.

```
public void AcceptConnections(){
                while (true && PassiveSocket.IsBound)
                {
                    TempConnection = PassiveSocket.Accept();

                    String connectionCommand = ReceiveCommandFrom(TempConnection);
                    if (ValidateConnection(connectionCommand))
                    {
                        SendCommand("Connection Permitted");
                        Connection = TempConnection;
                        HandleClientRequests();
                    }                    }
                    else
                    {
                        SendCommand("Connection Denied");
                        TempConnection.Close();
                    }

                }
}
```

This method receives the command and returns it to the caller.

```
public String ReceiveCommandFrom(Socket connection)
        {
            byte[] buffer = new byte[246];
            connection.Receive(buffer);

            return Encoding.ASCII.GetString(buffer);
}
```

This method handles the requests to executes them.

```
public void HandleClientRequests()
{
            try
            {
                while (Connection.Connected)
                {
                    Command command = ReceiveCommand();
                    SendCommandResponse(MainController.Execute(command));
                }
            }
            catch (Exception ex) {
                Log.print(ex.Message);
            }
}
```

In the client side (the smart phone):

Handling the keyboard keystrokes and Sending commands.

This code is an example on keyboard keystrokes handling.

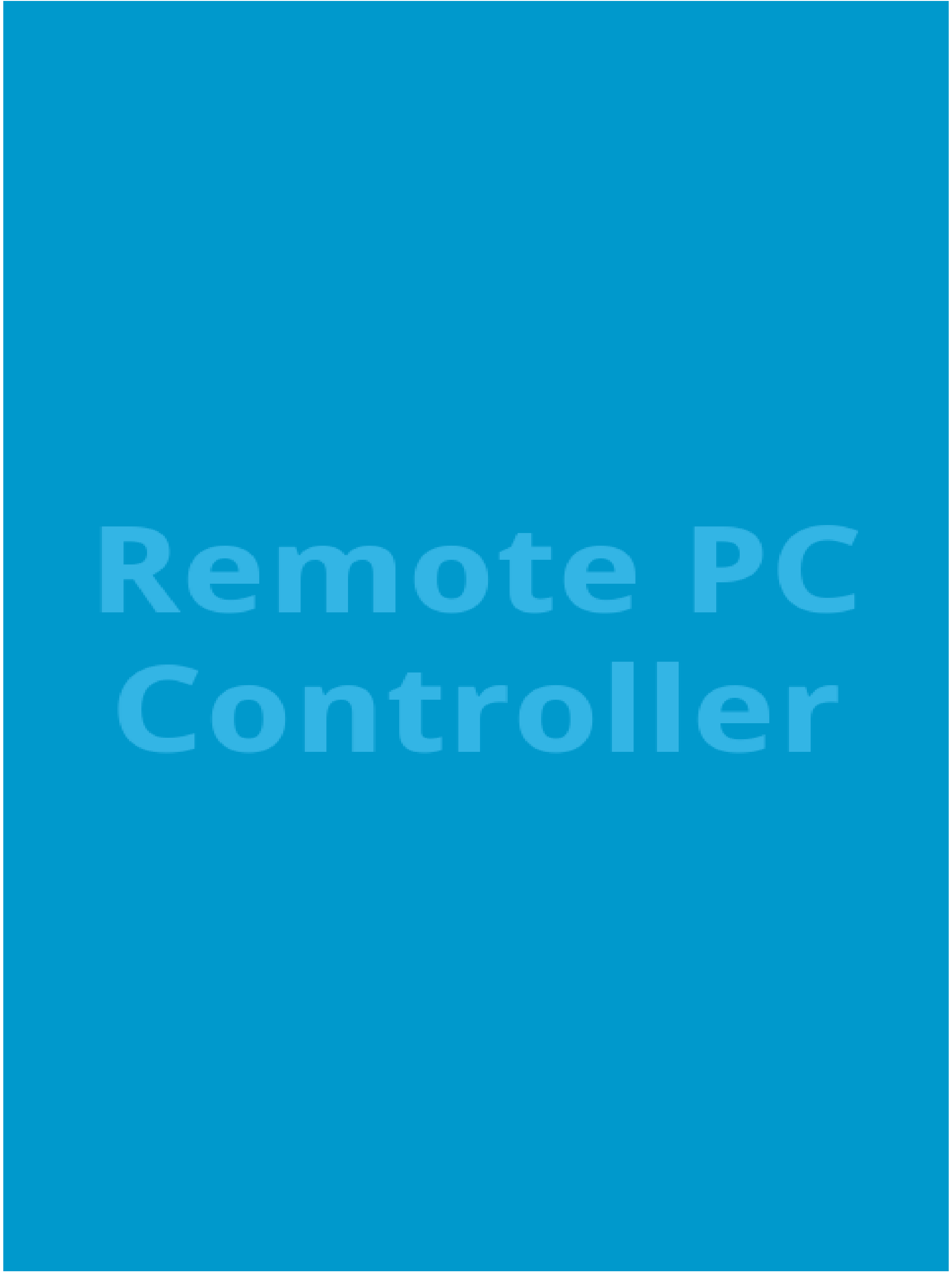
```
@Override
public boolean onKeyUp(int keyCode, KeyEvent event) {
      PCConnector.sendCommand("Key "+ event.getKeyCode());
      return super.onKeyUp(keyCode, event);
}
```

5.5 Snapshots for selected screens.

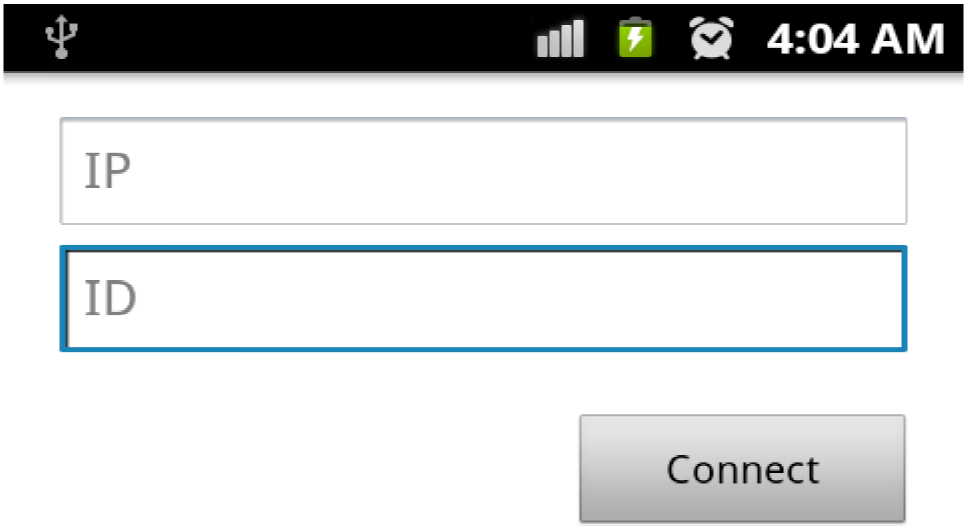
In this section the snapshots for the application will be shown.

1. On the Android Application:
    1. The start screen.

       The start Activity appears when the user starts the application.

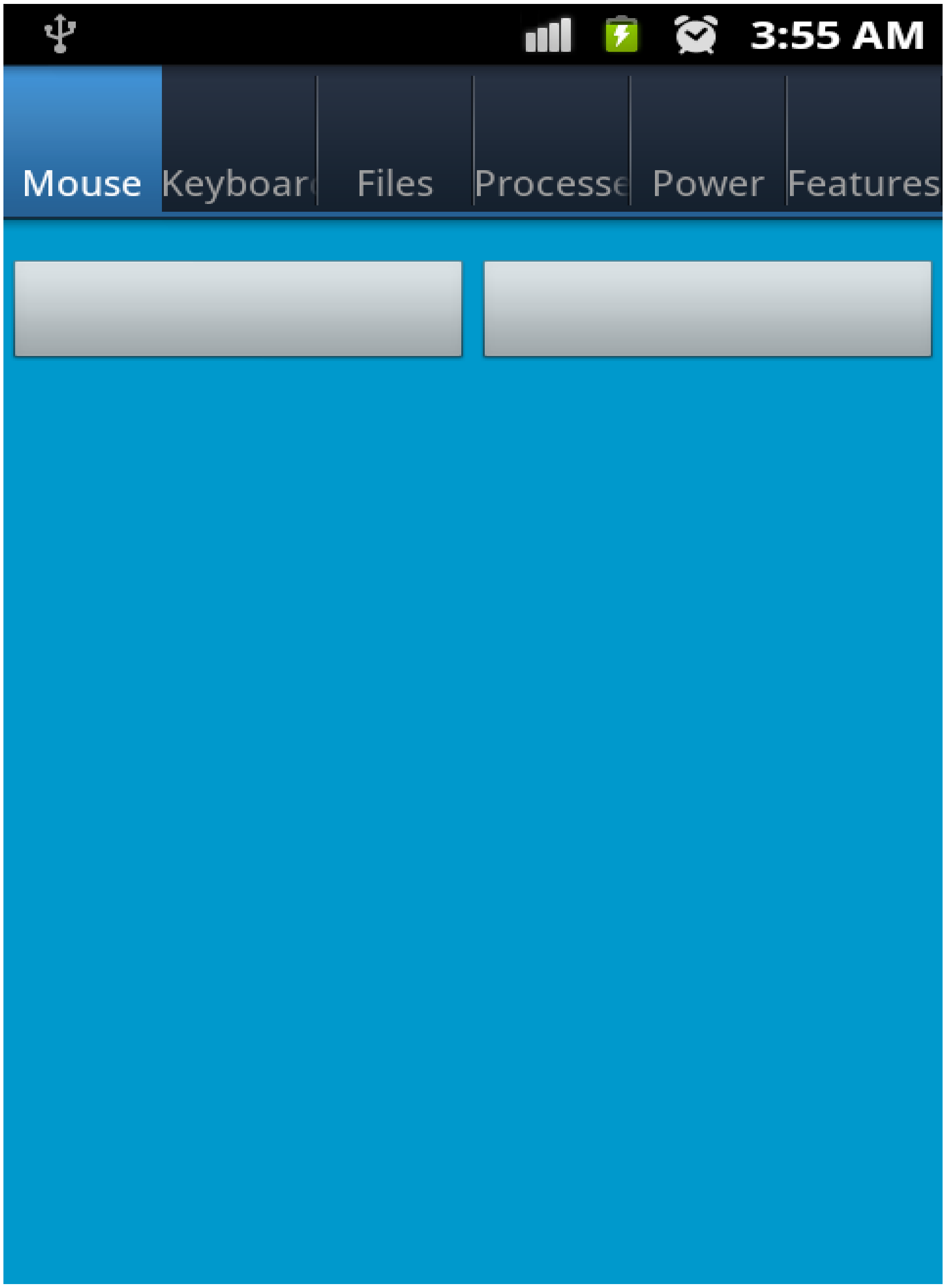


2. The Connect Activity.
   In this activity the user enters the IP of the PC on the IP text field and the ID in the ID Text field that taken from the PC Application

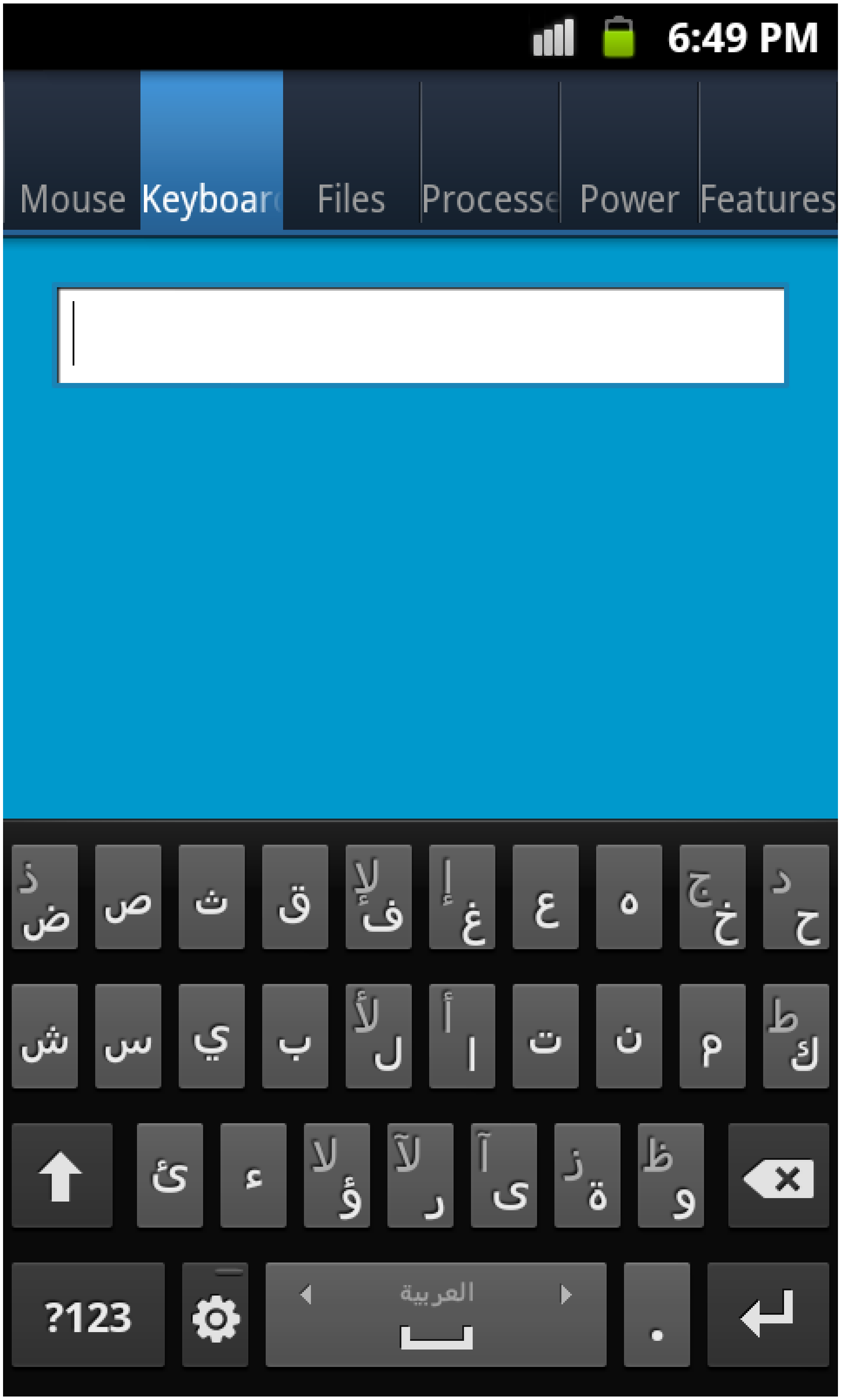


3. The Mouse Activity.

   This Activity is the mouse activity that the user can move his figure to control the mouse, or can scroll to scroll the wheel, or clicking the buttons for right or left click.

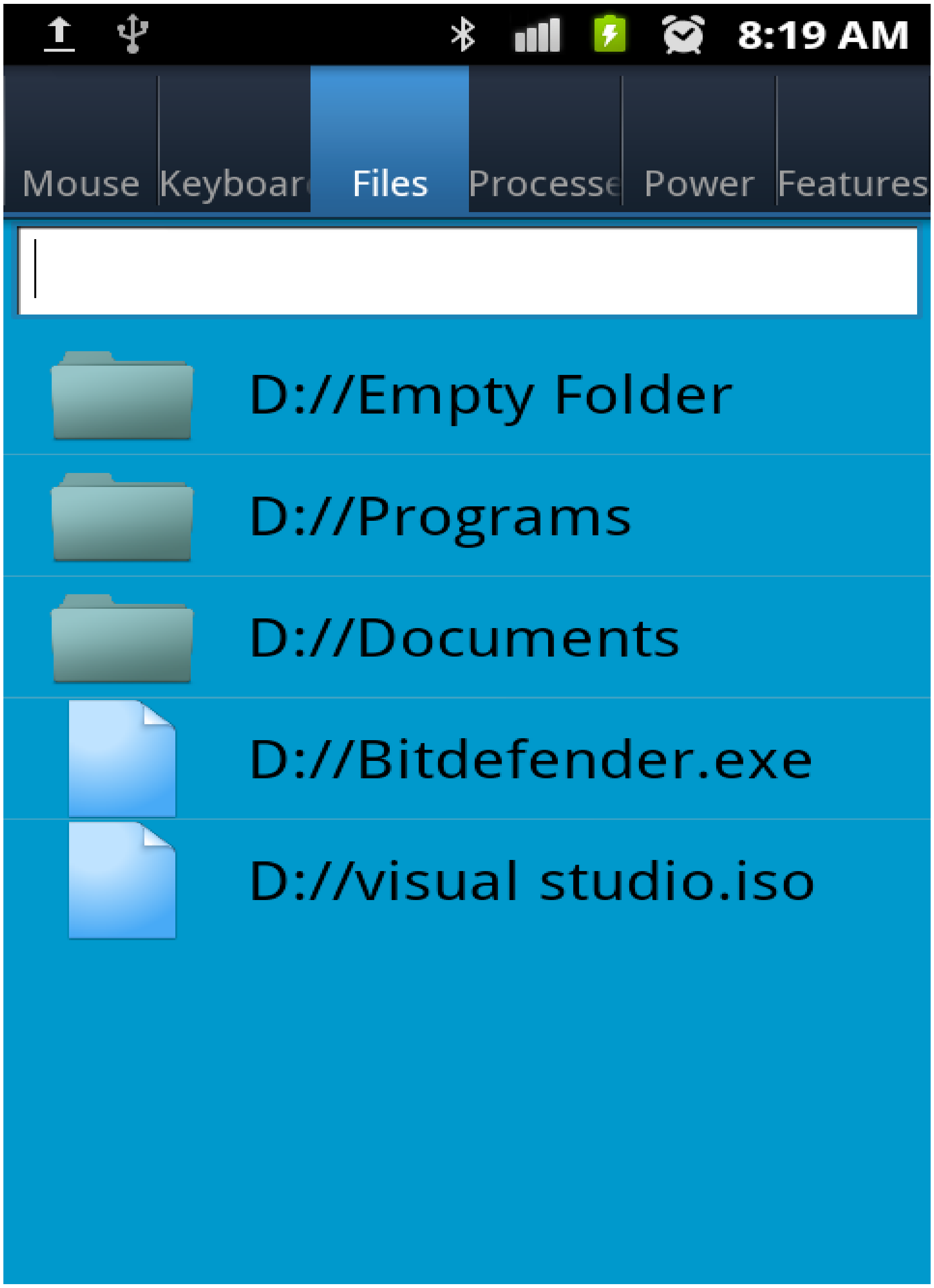


4. The Keyboard Activity.
   This activity enables the user to press keys to be sent to the keyboard.


6:49 PM
Mouse
Keyboar
Files
Processe
Power
Features
ذ ض ص ث ق لإ ف إ غ ع ه ج خ د ح
ش س ي ب لأ ل أ ا ت ن م ط ك
ئ ء لا ؤ لآ ر آ ى ز ة ظ و
?123
العربية
.

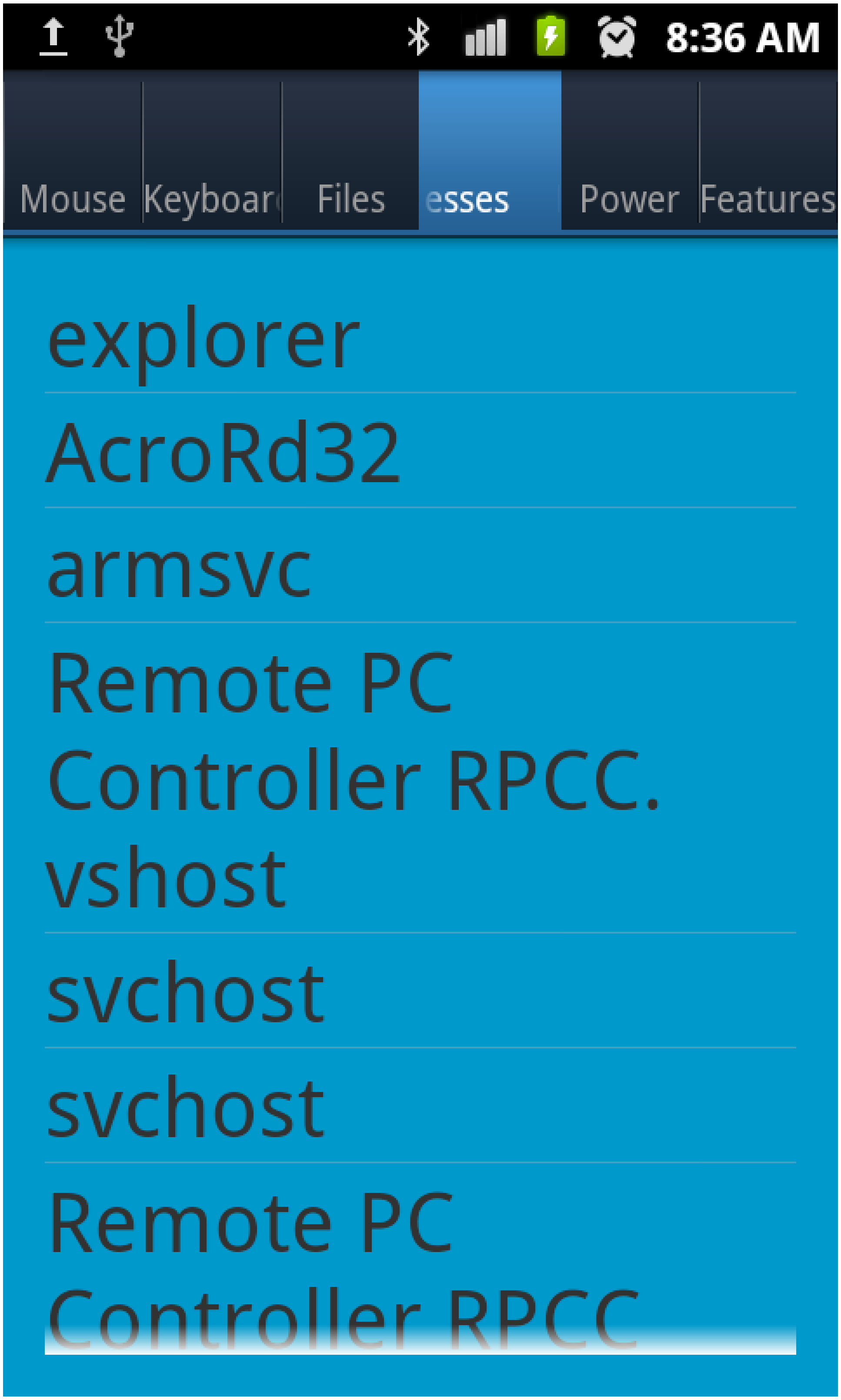
5. Files Activity.

   This activity enables the user to explore the file and perform specific operations on them by clicking long click on the item.



6. Processes Activity.

   This activity is responsible for retrieving and managing the processes.



explorer

AcroRd32

armsvc

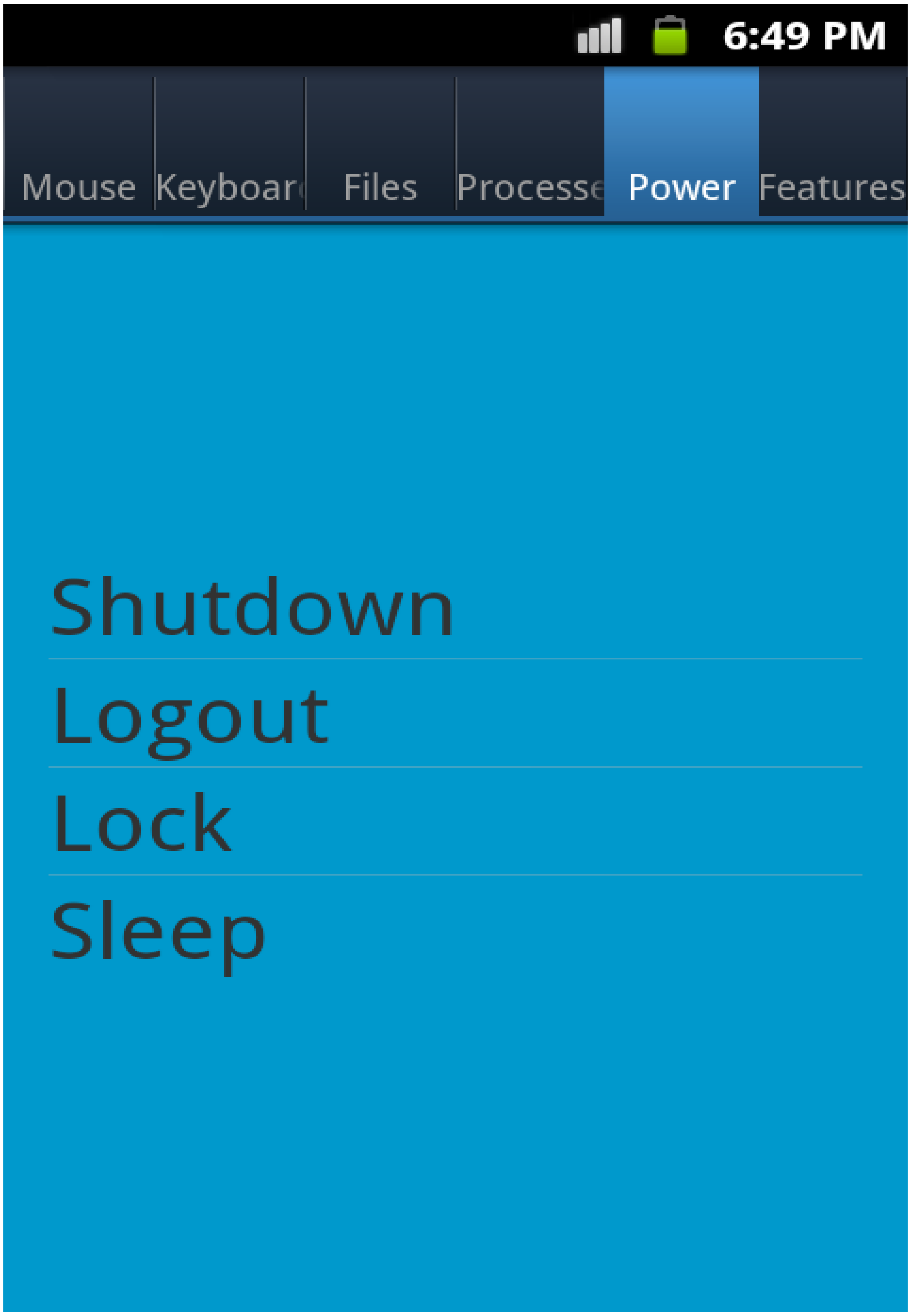
Remote PC Controller RPCC.vshost

svchost

svchost

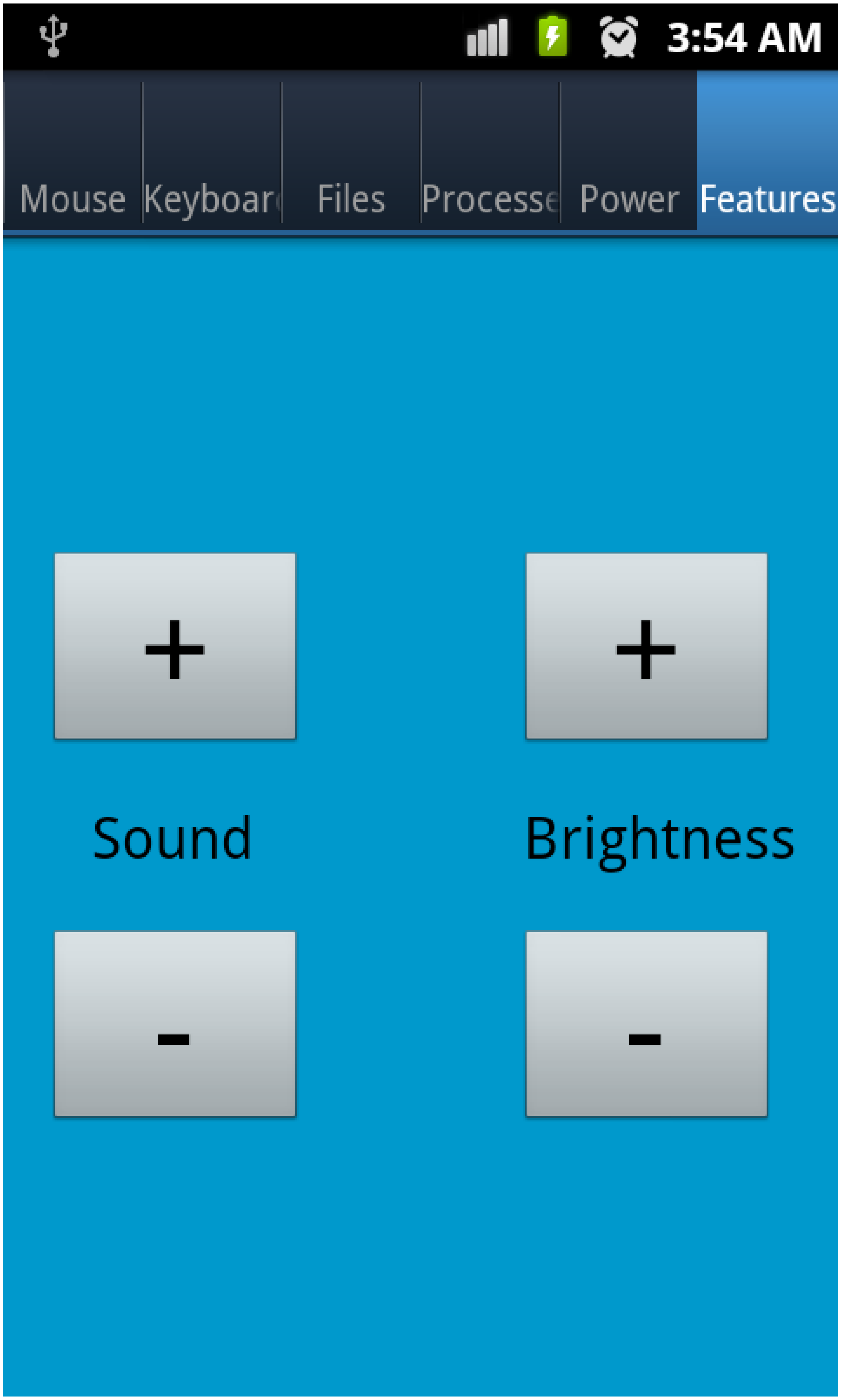
Remote PC Controller RPCC

7. Power Options Activity.
   Here you can control the power of the PC, so you can shutdown, logout, sleep, or lock the PC.

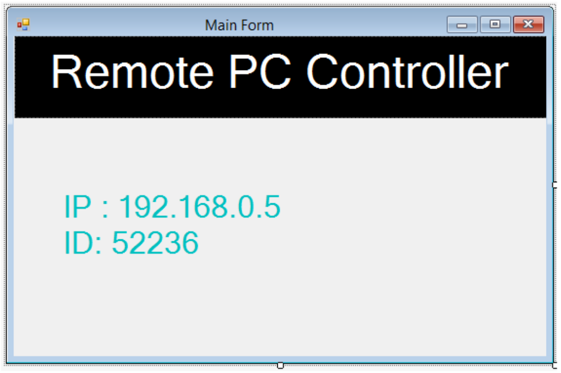


8. Features Activity.

   This activity can be used to control the brightness volume and the sound volume.



2. On the PC.
   There is one form shows the IP and the ID for the users, so the user can type them in the connect activity and connect to the PC to start using the application.



Remote PC Controller

IP : 192.168.0.5
ID: 52236

Chapter 6: Testing

1. Unit Testing.
2. Integration Testing.

6.1 Overview.

In this chapter, the unit testing and the integration testing will be explained.

6.2 Unit Testing.

For Exception handling the following code attached within each method, since the errors cannot be recovered but treated as done or not.

The log just used for debugging purposes.

```
try{


}catch(Exception ex){

	Log.print(ex.Message);

}
```

The classes that tested are:

1. SmartPhoneConnector class.
   The testing was for these methods:

   1. void InitConnection() :
   It initializes the sockets.
   2. void AcceptConnections():
   Tested for accepting connections.
   3. void SendCommand(String command):
      Tested for sending commands.
   4. String ReceiveCommand():
      Tested for receiving command.
   5. void HandleRequests():
      Tested for handling requests.

2. FilesManager Class.
   The testing was for these methods:

   1. void Open(filename):
      The file name: D:\\visual studio.iso
      The result: the file opened.
   2. void Delete(filename):
      The file name: D:\\file1.txt
      The result: the file deleted.
   3. void Rename(oldFileName,newFileName):
      The old file name: D:\\file2.txt
      The new file name: D:\\file3.txt
      The result: the file renamed to D:\\file3.txt.

3. ProcessesManager Class.
   The testing was for these methods:

   1. List<String> GetProcesses():
      Returns the running processes.
   2. void CloseProcess(String processname):
      Closes the Process.
   3. String ViewProcessInfo():
      Process Information retrieved.
   4. String ViewProcessInfo():
      Process Information retrieved.

4. MouseHandler class:

   The testing was for these methods:

   1. void RightClick():

      Result: Right click performed.

   2. void LeftClick():

      Result: Left click performed.

   3. void Move(int x, int y):

      Result: Cursor moved to x,y point.

   4. void Scroll(String direction):

   5. Result: Page scrolled.

5. SystemFeaturesManager class.

   1. void SoundUP():

      Result: action performed.

   2. void SoundDown():

      Result: action performed.

   3. void BrightnessUP():

      Result: action performed.

   4. void BrightnessDown():

      Result: action performed.

6. KeyboardHandler Class:

   1. void Execute(int key):

      "A" key entered and the action executed.

7. MainController Class:

   1. void Execute(String Command):

      A switch statement tested to ensure that execute gives the command to the class.

6.3 Integration Testing.

The classes integrated and worked properly, after the unit testing performed.