

A Project on

Remote Access in Android



GROUP MEMBERS:

SANCHITA SANTRA

SOMNATH GHOSE ROY

SABYASACHI PAHARI

Certificate

This is to certify that this Project entitled “Remote Access in Android” is a bona fide record of work done.

Group Member

Signature

Sabyasachi Pahari

.....

Sanchita Santra

.....

Somnath Ghose Roy

.....

Under my guidance and supervision and submitted in partial fulfilment of the requirements for the award of B.Tech degree in Computer Science and Engineering by the West Bengal University of Technology.

Dr. Anupam Ghosh

Head of the Department
Department of CSE
Netaji Subhash Engineering College

Mr. Sourav Dutta

(Project Coordinator)
Department of CSE
Netaji Subhash Engineering College

Ms. Ranu Sarkar

(Project Guide)
Assistant Professor,
Department of CSE
Netaji Subhash Engineering College

Group Details

Roll No	NAME	WBUT Roll No
77	Sabyasachi Pahari	10900111077
82	Sanchita Santra	10900111082
97	Somnath Ghose Roy	10900111097



Netaji Subhash Engineering College

Final Year Project

Year: 2015

Index

Sl. No.	Topic	Page no.
1	Acknowledgement	4
2	Project Abstract	5
3	Introduction	6
4	Literature Survey	10
5	System Specification	12
6	ERD	13
7	Database Table Structure	14
8	DFD	15
9	UML Diagrams	18
10	Screenshots	23
11	Smart Features	32
12	Future Scope of Improvement	36
13	Conclusion	38
14	Bibliography	39

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mentioning the names of people whose constant guidance and encouragement crowns all efforts with our success.

We take this opportunity to express our profound gratitude and deep regards to our mentor, Mrs Ranu Sarkar, for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark.

We also wish to thank all faculties specially Dr. Anupam Ghosh (HOD of CSE Dept.) and Mr. Sourav Dutta(Project Coordinator) and non-teaching staffs for supporting us during our whole project work and our college management for providing us with adequate technical infrastructure.

We are obliged to our project team members for the valuable information provided by them in their respective fields. We are grateful for each other's cooperation during the period of this assignment.

Finally we extend our profound gratitude to our respective families for helping us through the busiest times and putting up with the late nights filled with development time to keep to the deadlines of this project.

Project Abstract

This project is about the developing an android application. Nowadays, the gadgets are rolling the world. Many people cannot imagine even one day without their favourite mobile device. We use them for everything: find information, stay connected with our friends and families, find the way around, decide what to do, and many other things. But very often we come to the point when we would like to have an application for particular situation or for certain need, but there is no such one.

The advent of smart phones has drastically changed the use of internet in everyday life. Applications that we used to have on personal computers are simpler and more user friendly via smart phones. This project plans to make smart phones even more integrated with our daily lives.

In this project we plan to control a smartphone by another smartphone remotely over the Internet via HTTP request and response. To make our objective successful we have built an app, which provides features to control flashlight and camera of other devices from anywhere in the world.

In our app firstly when a user registers himself/herself, all the registering credentials are stored in our local database as well as server. After successful registration, the user login in the app. Then the user can add a device which he/she desires to control remotely. This involves filling the required device username with proper registered name of the device. Thus, a request is sent to the device to be controlled. After successful acceptance both these devices gets connected over the internet. Now, we can turn on the flashlight and can eventually turn it off. We can also turn on the camera, enable it to take pictures and send it to the person who controls it remotely. All these are done in the background.

Introduction

Android Technology

Android is rocking the global smartphone market, owning 85 percent of marketshare, according to new Strategy Analytic research, which reports worldwide smartphone shipments hit 295 million units in the second quarter of 2014.



Like the PC market, Android is on the verge of turning smartphone platforms into a one-horse race. Its low-cost services and user-friendly software remain wildly attractive to hardware makers, operators and consumers worldwide.

Android framework:

Android is one of an Open source platforms. It is created by Google and owned by Open Handset Alliance. It is designed with goal “accelerate innovation in mobile” As such android has taken over a field of mobile innovation. It is definitely free and open platform that differs hardware from software that runs on it. It results for much more devices be running the same application. Also it gives possibility of friendlier environment for developers and consumers. Android it is complete software package for a mobile device. Since the beginning android team offers the developing kit (tool and frameworks) for creating mobile applications quick and easy as possible. In some cases you do not specially need an android phone but you are very welcome to have one. It can work right out of the box, but of course users can customize it for their particular needs. For manufactures it is ready and free solution for their devices. Except specific drivers android community provides everything else to create their devices.

The objective behind making this application was to bring the functionalities of a network service provider onto a mobile device. So while surveying as to on which platform or rather operating system the project has to be implemented ,we selected android for the following reasons:

- Android is an open source platform
- Supports multifunction
- Provides rich tools to make interactive application
- Downloading the software's required for making the application are absolutely free.

This initial survey provides an insight into the preferences of the target audience. The target audience include those who meet three criteria:

- Own an Android device.
- Have used, or may use their android device for entertainment purposes.
- Have downloaded an Android application from the Android Market.

Initial setup:

The initial setup for using the android platform has been done on both Windows and Linux machines. We are using Android Studio, the official IDE for Android application development, based on IntelliJ IDEA. Android applications can be easily developed and debugged using ADT. We tested our initialsetup by running a simple "Hello, World" program.

Application Signing:

All the Android applications needs to be digitally signed else they won't be installed on emulator or mobile device. There were some problems related to this since some of the required environment variables were not set. Those problems are fixed now.

Project folder structure:

Basic Android project would have six directories such as: assets, bin, gen, libs, res,src. Also there are some files in project root directory such as: AndroidManifest.xml, licenses, project.properties and other files.

The most important for the developers are “res” and “src” directories. “res” directory contains all the current project resources such as: images, layouts, custom strings and other values. Images are stored in different directories depending on their size that application can automatically choose right image depending on the device specifications. Layouts are store in the “layout” folder. Basically layout file example would be an XML file which would specify elements and their position in current view. Also it is possible to code custom strings and colors so the parser can display them in application. It is recommended approach to store them in values directory rather than hard code to the actual code or XML file. It would make easy further development at translating the application to other languages.

The other important directory is “src”. This directory would usually consist of Java files which are adding functionality to the application. Then developer would create classes as separated Java files. If the class is created in GUI ADT environment the tool would generate automatically the statement in “AndroidManifest” file. If other programming environment is used, user must specify any new class activity by hard coding the “AndroidManifest” file. Android manifest file usually would be placed to the root directory of the project and state required version of android, needed permissions and all activities which are run within the application.

Remote Access:

Remote access is just what it sounds like -- the ability to access your computer from a remote location. Programs like PC Anywhere (Windows), Remote Access (Mac), and Timbuktu (Windows and Mac) allow users to control remote computers from their local machine. In order for a remote access connection to take place, the local machine must have the remote client software installed and the remote machine must have the remote server software installed. Also, a username and password is almost always required to authenticate the connecting user.

Remote access is more than just being able to connect to a remote machine. It is the ability to control the machine once the connection has been made. A remote access program can basically transform your local device into the remote device you connect to. This is great for people who sometimes work from home and for server administrators who frequently need to update and make changes on their server machines. Most remote access programs also allow users to transfer files between the local and remote machines, which can save a lot of commuting time.

World is contracting with the growth of mobile phone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular handsets which were used just for making phone calls, mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses and can be used as a Camera, Music player, Tablet PC, T.V., Web browser etc .With the advent of android technology, more and more people are switching to smartphones. Statistics suggest that in the next 10 years smartphones will replace PCs.

It is not that we can remotely control computers only. In our project we have taken a step further towards achieving remote access over the internet via android app.

Literature Survey

To implement our app “MobileControl” we researched extensively on the topic remote access since our project is built on this particular concept. We searched a lot of existing applications which support remote access and made a detailed analysis report on their pros and cons. Some of the applications which really fascinated and encouraged us in building this app are as follows:

TeamViewer

TeamViewer connects to any PC or server around the world within a few seconds. We can administer remote computers and servers anytime, anywhere as if we were sitting directly in front of them. If we save the computers or servers as a contact in the Computers & Contacts list, we can connect easily with just a single mouse-click!

Chrome Remote Desktop:

Chrome Remote Desktop allows users to remotely access another computer through Chrome browser or a Chromebook. Computers can be made available on a short-term basis for scenarios such as ad hoc remote support, or on a more long-term basis for remote access to our applications and files.

Splashtop:

It's perhaps most notable as a tool that allows us to stream audio and video across computers with minimal latency, so if we love watching movies on our tablet that are stored on our desktop but don't want to deal with apps or compatibility issues, Splashtop is a great tool.

Microsoft Remote Desktop (RDC)/Apple Remote Desktop:

Microsoft's RDC protocol and Apple's own Remote Desktop platform both use existing technologies within each respective operating system to give remote administrators the ability to connect from anywhere they need to, access their files, troubleshoot problems, or work with files and applications as though they were using the remote device.

VNC:

VNC, or Virtual Network Computing, is less of a specific product and more of a platform. It uses existing protocols to send keyboard and mouse actions to a remote computer, and in turn it sends the screen from that remote system back to our viewer. Depending on the VNC client and server software we use, we get more features, like clipboard syncing, file sync and transfer, and more.

All these applications use desktop to desktop or mobile to desktop communication. We wanted to build something new and more user friendly application. So, we came up with the idea of accessing and controlling an android powered smartphone from another smartphone as with the advent of android technology, more and more people are switching to smartphones.

Next we continued our survey to find best possible ways by which we can remotely access a smartphone from another smartphone and we consulted some research based article named "Remote Control of Mobile Devices in Android Platform" written by Angel Gonzalez Villan and Joseph Jorba of Universitat Oberta de Catalunya, Barcelona, Spain.(Bibliography 1.c)

We can achieve remote access by three ways mainly-

1. HTTP Request and Response
2. Google Cloud Messaging (GCM)
3. Local Area Network (Bluetooth or wireless)

We have tried out all the above ways and ultimately implemented our app using Internet via HTTP request and response method.

System Specification

• Hardware requirement

➤ Client side

- Ram : Minimum 20Mb.
- Platform : ARM architecture
- Camera
- Flashlight

➤ Server side

- Processor : Minimum of 2.5 Ghz or Intel Pentium Dual Core
- System type : 32 or 64 bit Operating System
- RAM : Minimum 1GB.
- Hard Drive : 60 GB or higher.

• Software Requirement

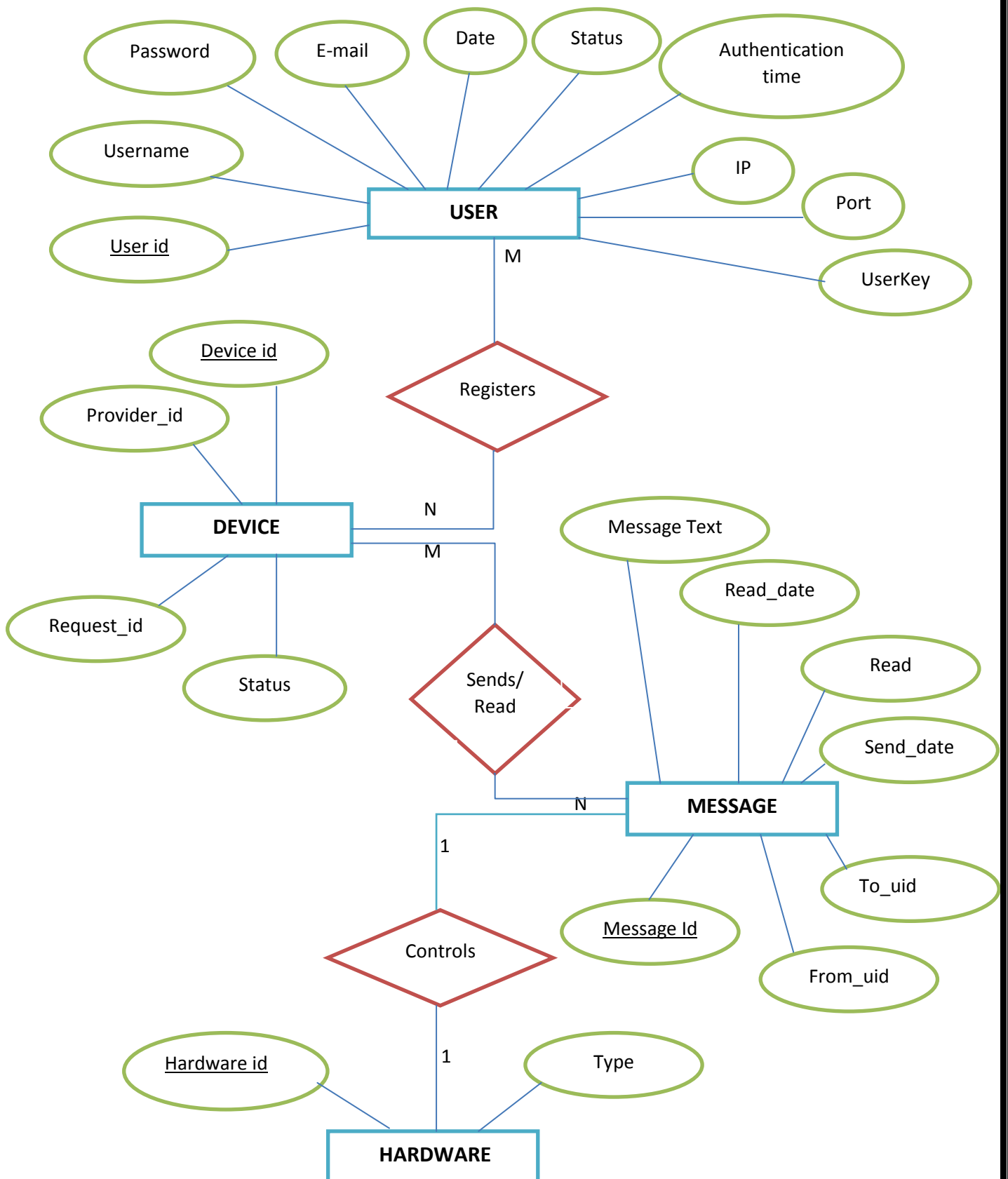
➤ Client side

- Android 10 – Gingerbread.

➤ Server side

- Mysql
- Php
- Jdk 1.7.0_13
- Jre7

Entity Relationship Diagram



Database Table Structure

Description of table structure:

Table name: 'users'

Field	Type	Null	Key	Default	Extra
user_id	Int(11)	NO	PRIMARY	NULL	auto_increment
Username	Varchar(45)	NO		NULL	
Password	Varchar(32)	NO		NULL	
Email	Varchar(45)	NO		NULL	
Date	Datetime	NO		0000-00-00 00:00:00	
Status	tinyint(3)	NO		0	
authenticationTime	Datetime	NO		0000-00-00 00:00:00	
userKey	Varchar(20)	NO		NULL	
Port	Int(10)	NO		0	
IP	Varchar(45)	NO		NULL	

Table name: 'messages'

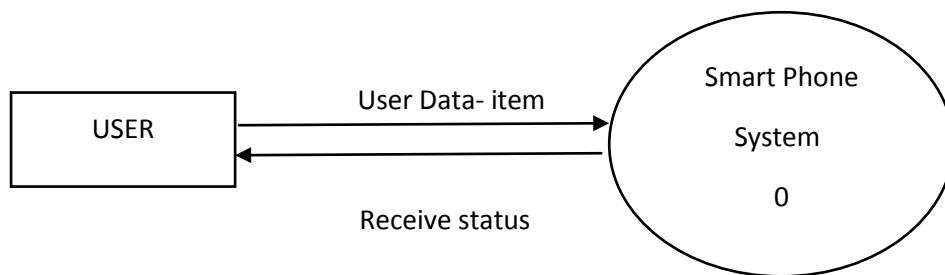
Field	Type	Null	Key	Default	Extra
message_id	int(255)	NO	PRIMARY	NULL	auto_increment
from_uid	Int(255)	NO		NULL	
to_uid	Int(255)	NO		NULL	
sent_date	datetime	NO		NULL	
Read	tinyint(1)	NO		0	
read_date	datetime	NO		NULL	
messagetext	longtext	NO		NULL	

Table name: 'devices'

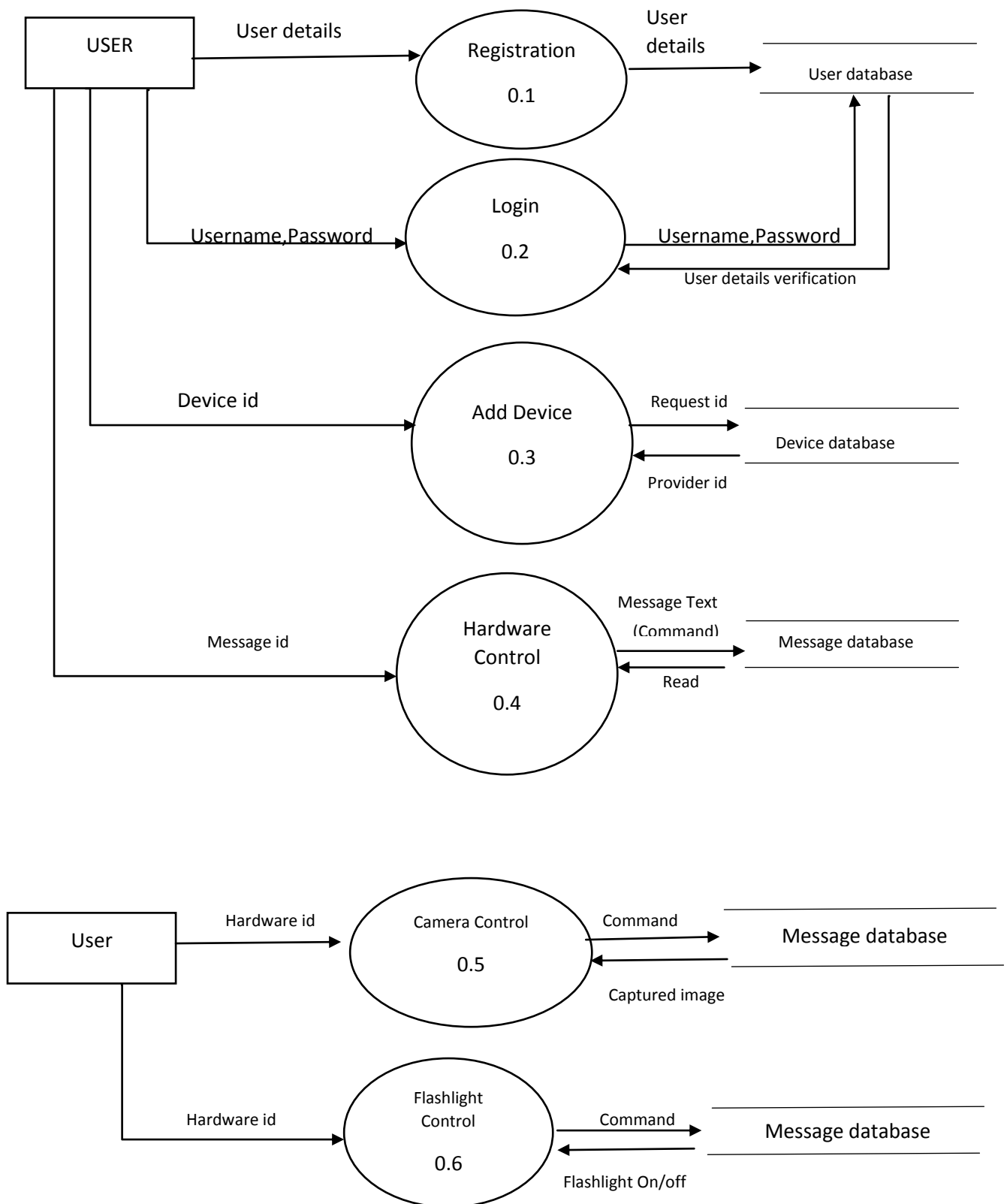
Field	Type	Null	Key	Default	Extra
device_id	int(11)	NO	PRIMARY	NULL	auto_increment
provider_id	Int(10)	NO		0	
request_id	Int(10)	NO		0	
Status	binary(1)	NO		0	

Data Flow Diagram

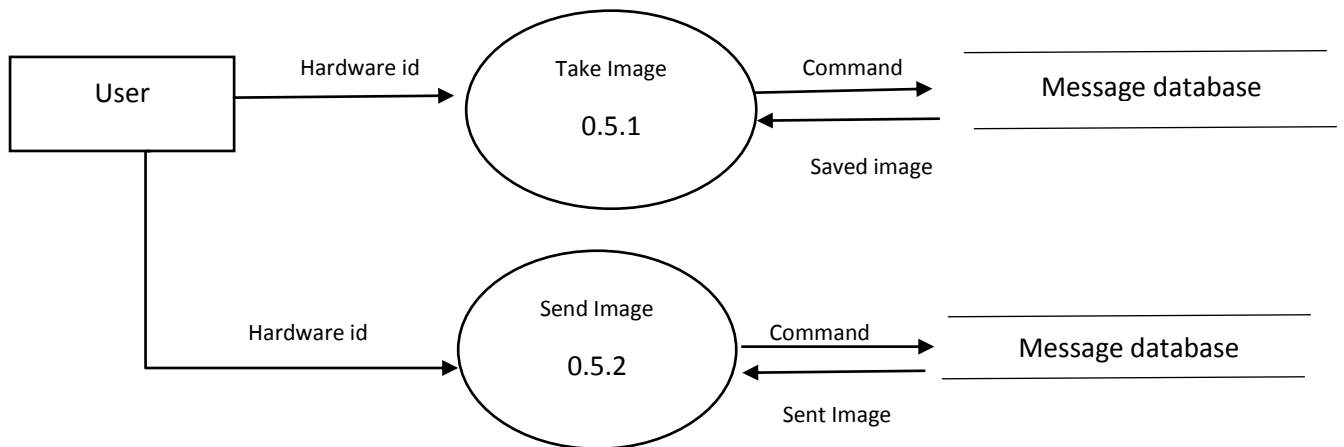
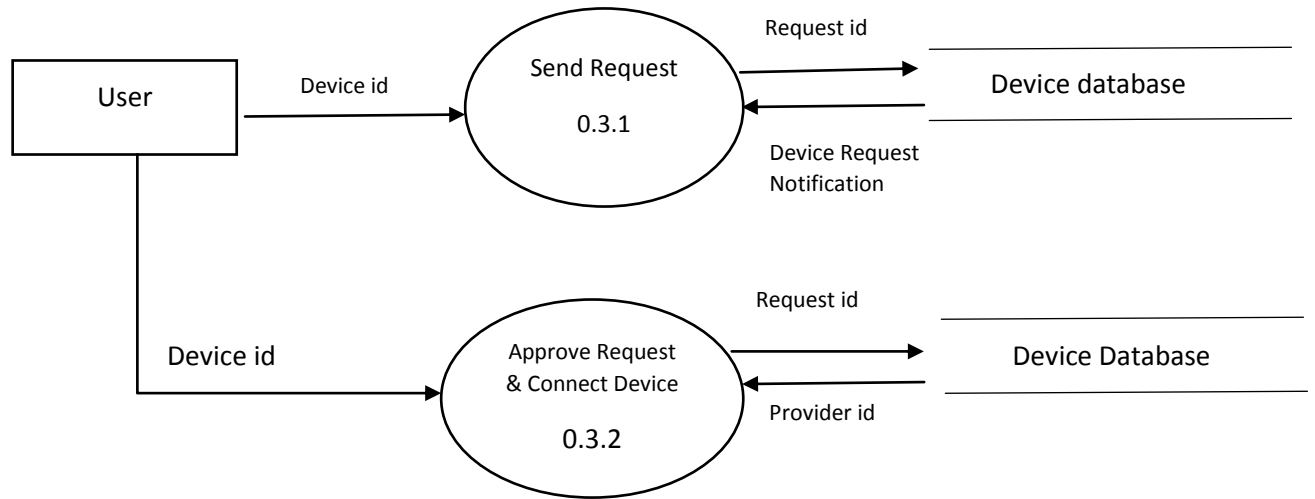
Level 0:



Level 1:

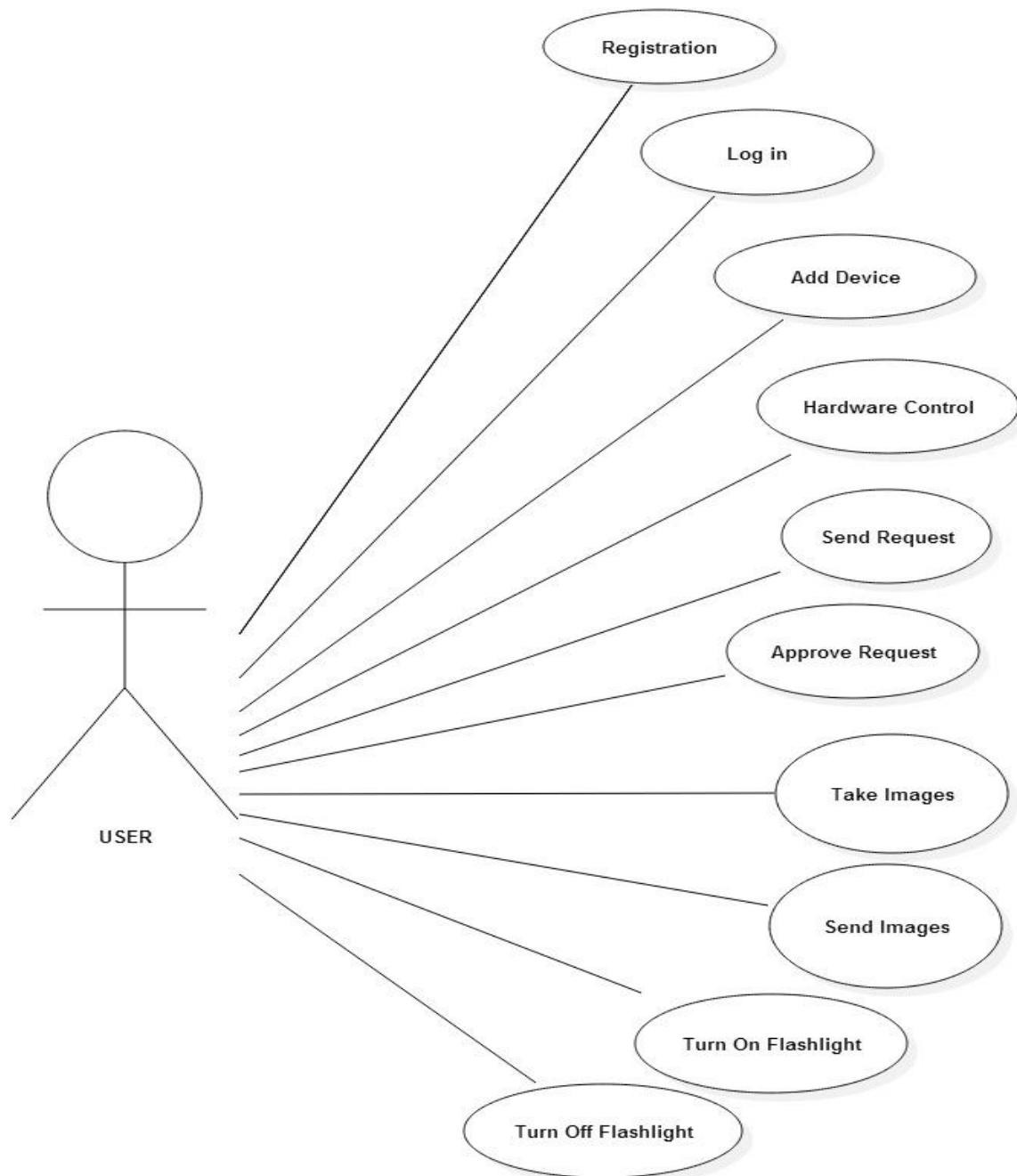


Level 2:

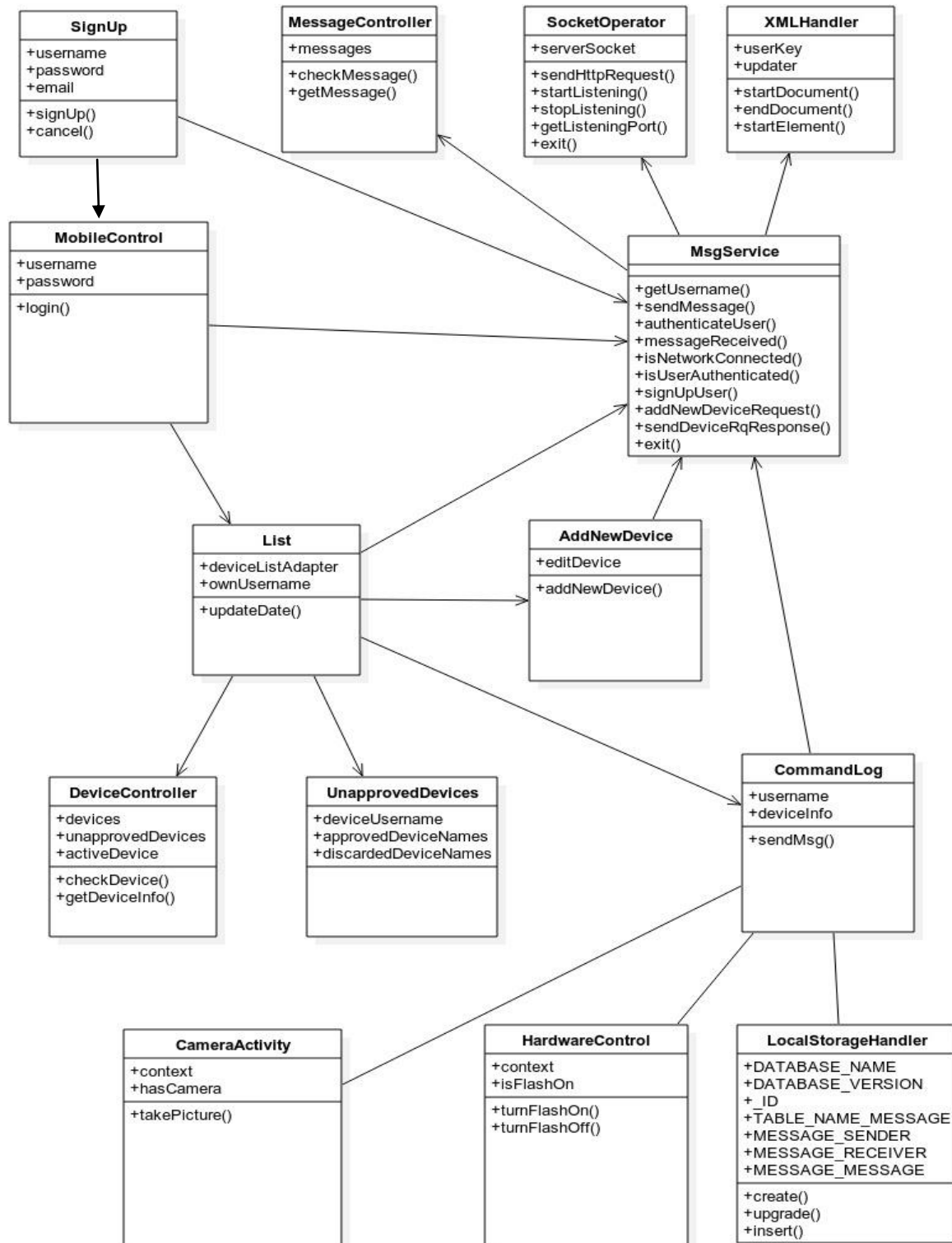


UML Diagrams

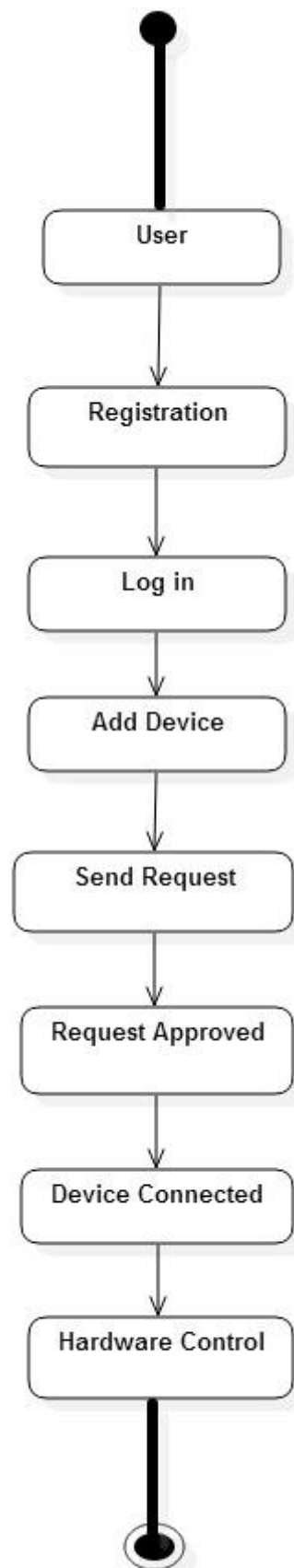
1. Use Case Diagram:



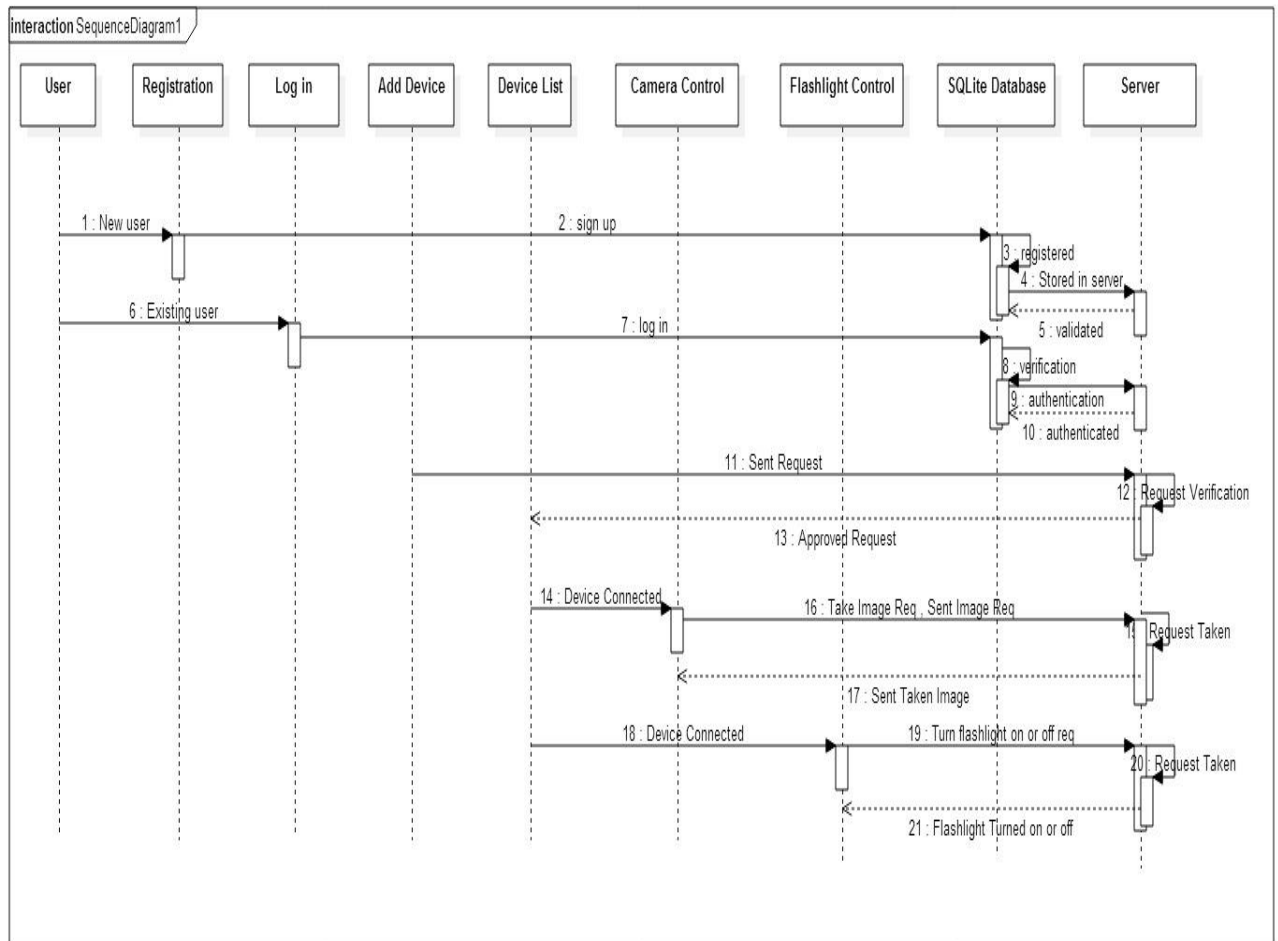
2. Class Diagram:



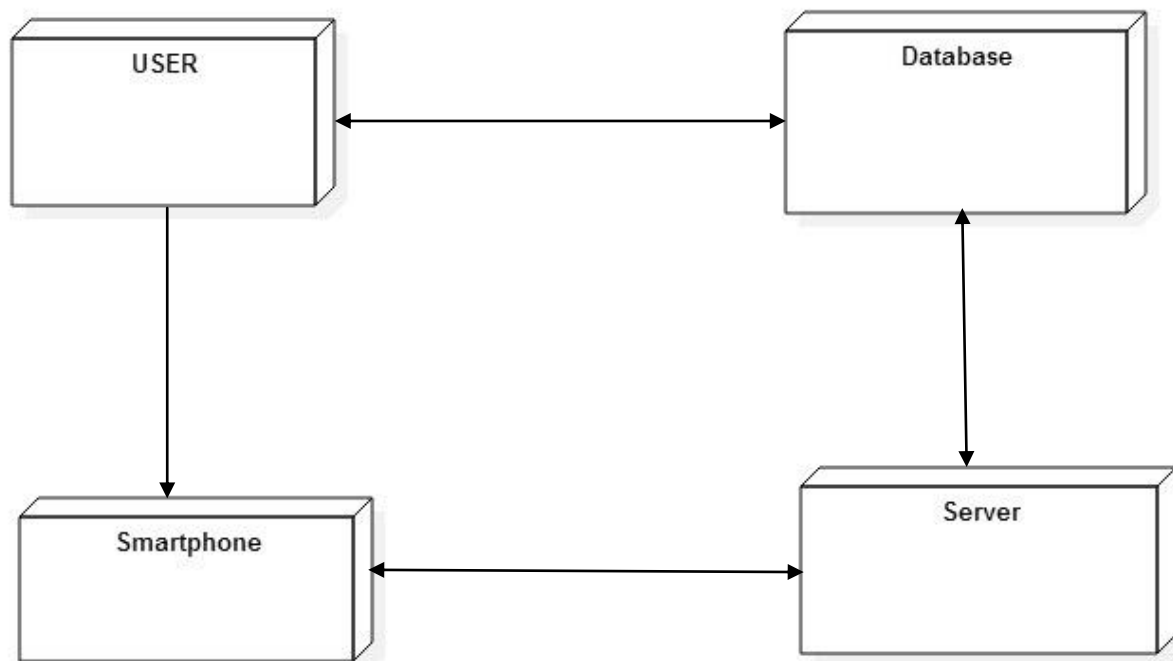
3. Statechart Diagram:



4. Sequence Diagram:



5. Deployment Diagram:

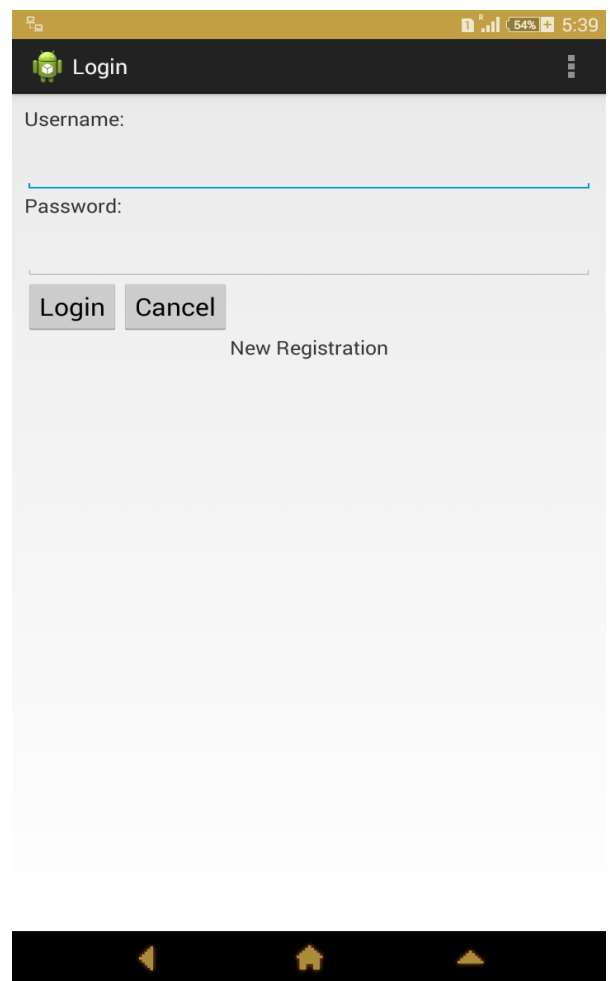


Screenshots

1.

LOGIN PAGE

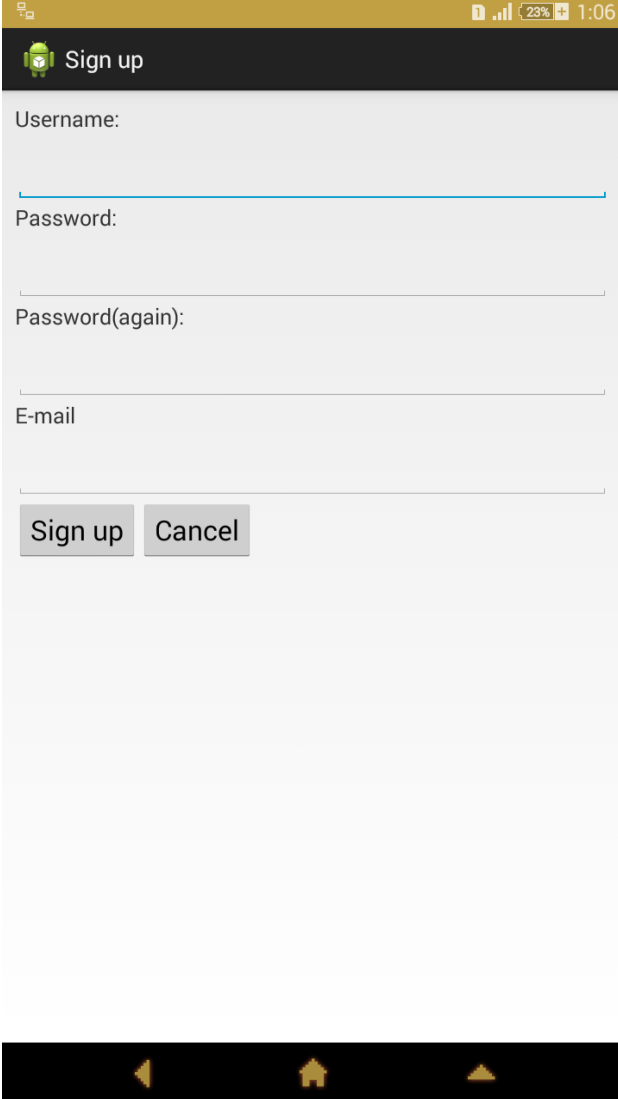
Here the user needs to give the username and password to gain access to the app. If the user is not registered then he/she will have to click on the 'New Registration' link to register himself/herself first.



2.

REGISTRATION PAGE

If the user is new, then he/she has to fill the following fields i.e. give necessary details (username, password and E-mail id) to register himself/herself in this app. After this registration with the given username and password the user can login to the app via login page.



The screenshot shows an Android application interface for a 'Sign up' page. At the top, there is a status bar with a gold background showing signal strength, 23% battery, and the time 1:06. Below this is a dark header bar with a green Android icon and the text 'Sign up'. The main area is light gray and contains four text input fields labeled 'Username:', 'Password:', 'Password(again):', and 'E-mail'. Each field has a blue underline. At the bottom of the form are two buttons: 'Sign up' and 'Cancel'. The very bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

3.

Sign Up

User completes his/her registration successfully.

The screenshot shows a mobile application interface for a 'Sign Up' screen. At the top, there is a status bar with icons for battery, signal, and time (8:17). Below the status bar is a dark header with the text 'SignUp' and a menu icon. The main content area is light gray and contains the following fields and buttons:

- Sign Up** (Title)
- Username:**
- Password:**
- Confirm Password:**
- Email:**
- Sign Up** (Button)
- Cancel** (Button)

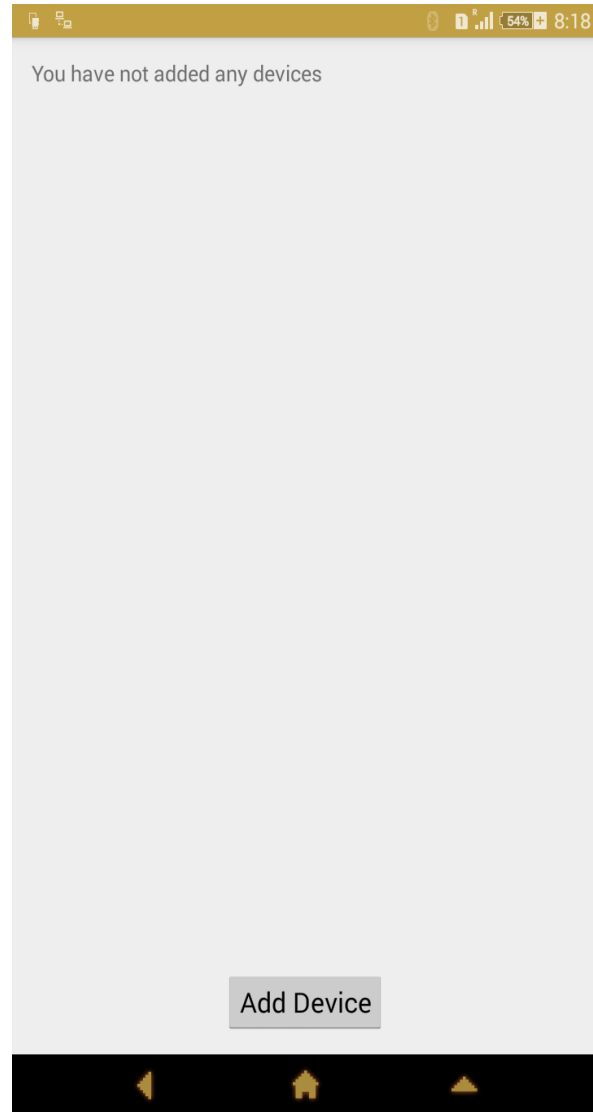
At the bottom of the screen, there is a dark navigation bar with three icons: a back arrow, a home icon, and a forward arrow.

A dark gray toast message is displayed at the bottom of the screen, stating: "Your registration is completed successfully, you can login now".

4.

Add Device menu

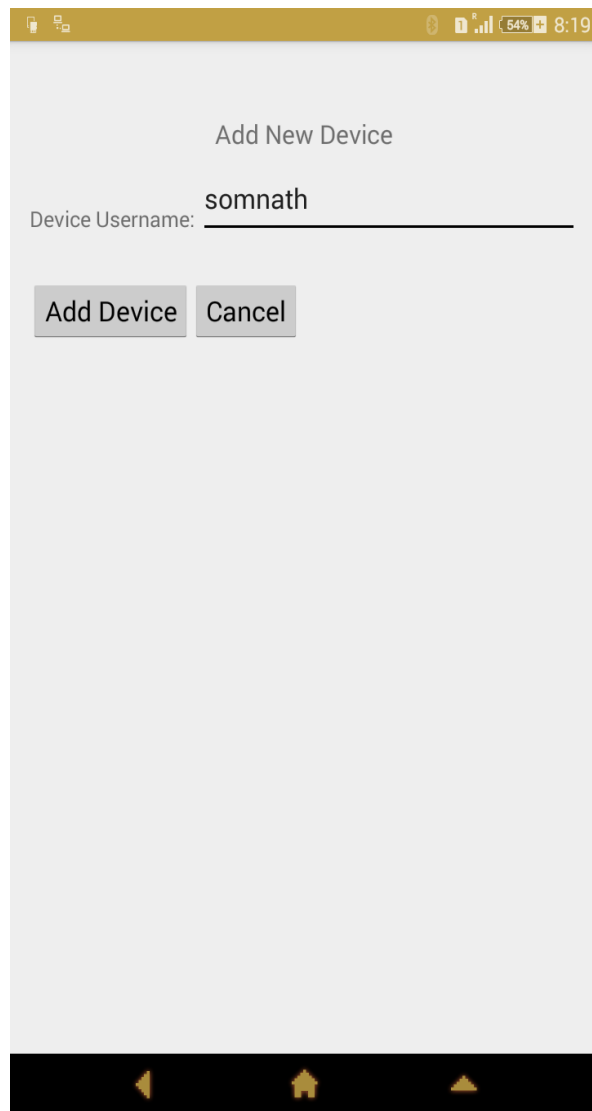
Device button which enables a user to configure a device.



5.

Adding a new device

In this screen user adds a new device to which he/she wishes to control.

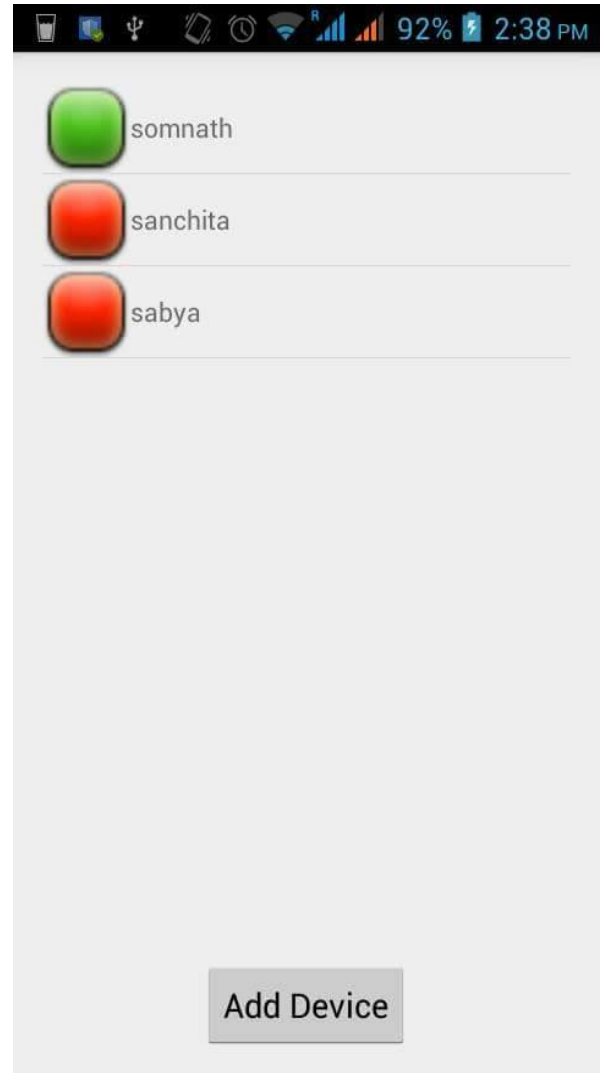


The screenshot shows a mobile application interface for adding a new device. At the top, there is a status bar with icons for battery, signal, and time (8:19). Below the status bar, the title "Add New Device" is centered. Underneath the title, there is a label "Device Username:" followed by a text input field containing the value "somnath". At the bottom of the form, there are two buttons: "Add Device" and "Cancel". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

6.

Device List

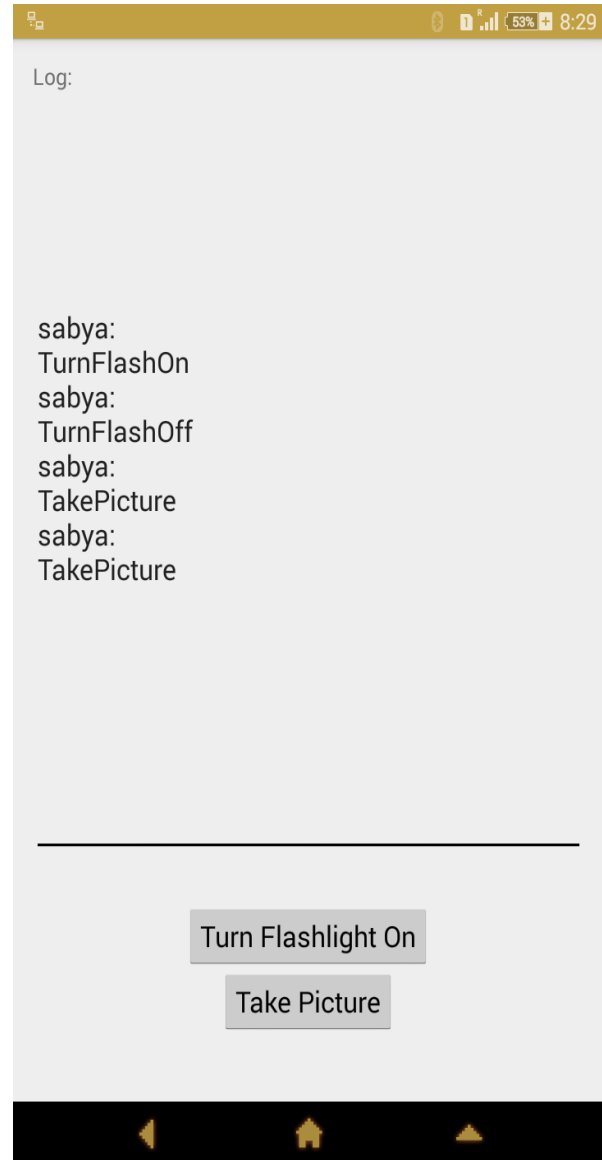
Configured devices are shown in this screen. Red ones show the devices which are offline. Green ones are those which are online.



7.

Hardware Control

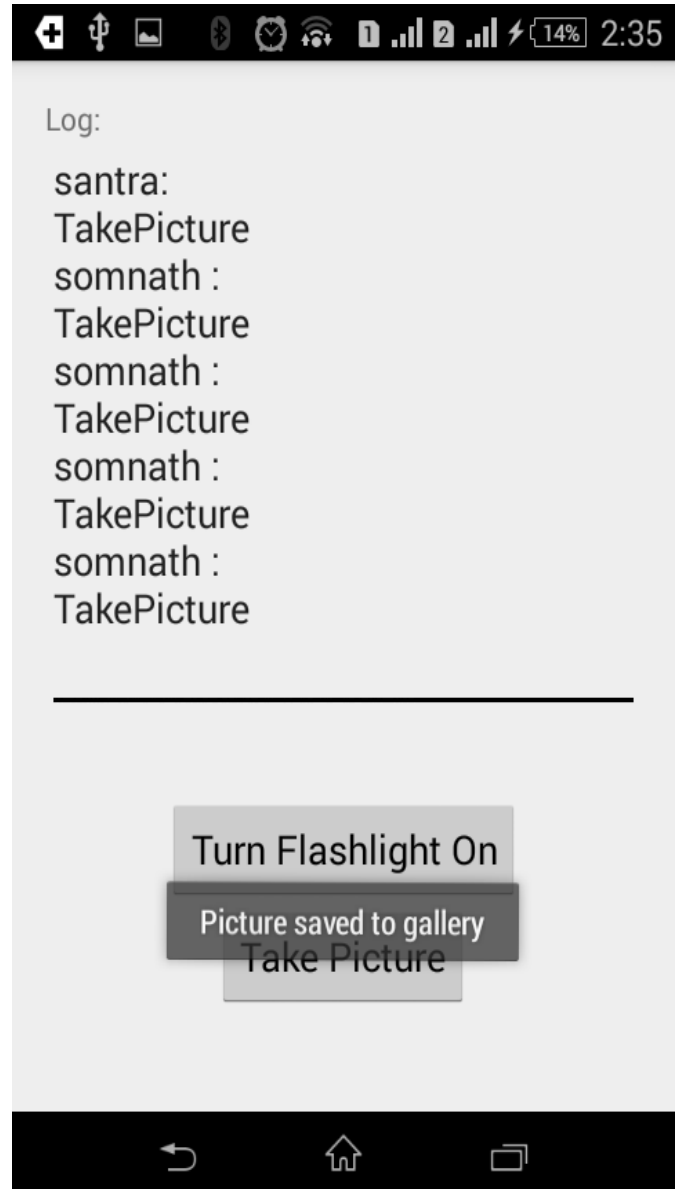
This screen provides user with two buttons - Turn Flashlight On/Off and take picture button.



8.

Taking Pictures

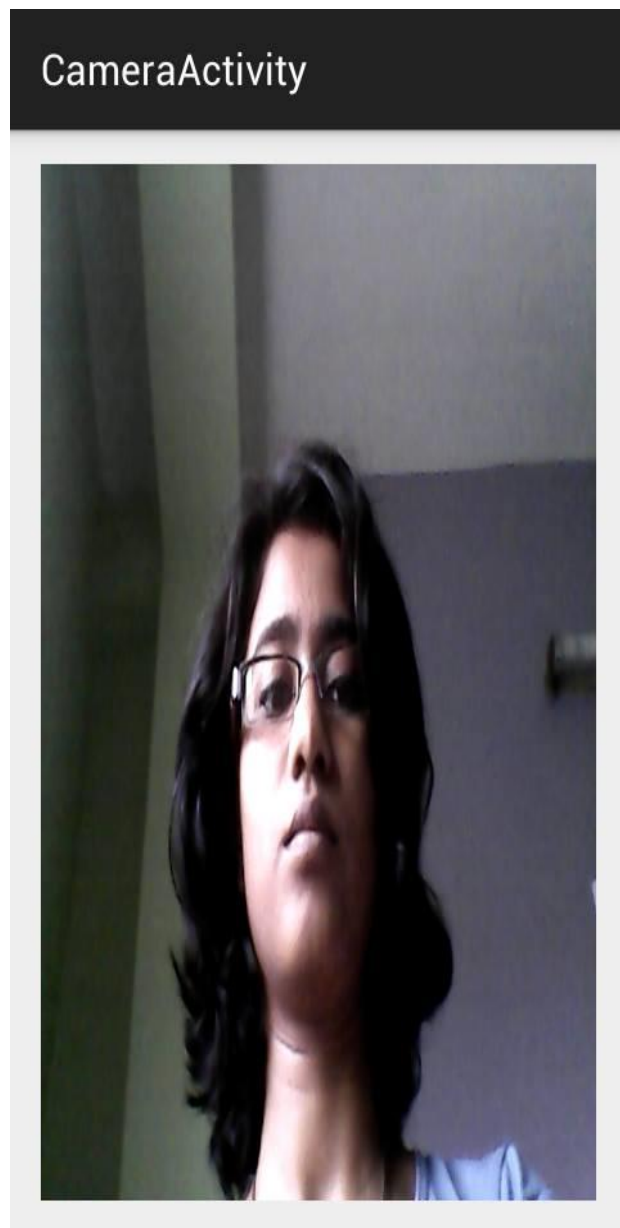
In this screen when one user clicks on the Take Picture button say for eg. There are two devices with usernames 'santra' and 'somnath'. When device 'somnath' triggers the Take picture button then in the log appropriate messages are displayed and a message 'Picture saved to gallery' is displayed.



9.

Image Preview

This screen shows up the image that gets stored up in both devices eg. device named 'somnath' and 'santra' .Image is actually saved up in the gallery.

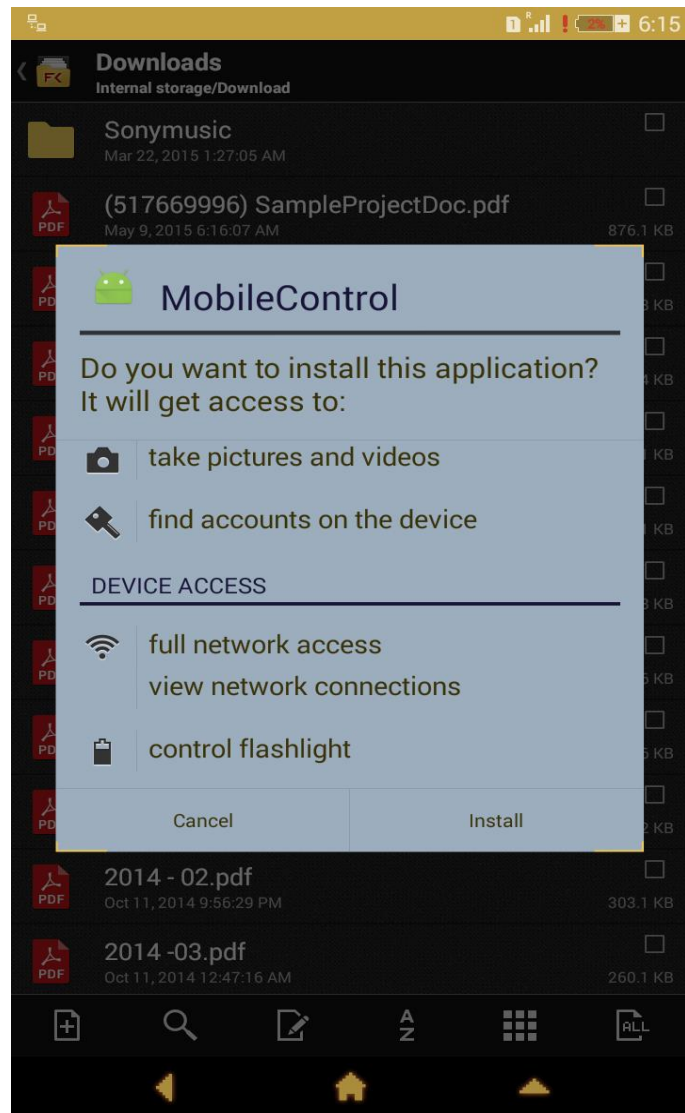


Smart Features

1.

Installation

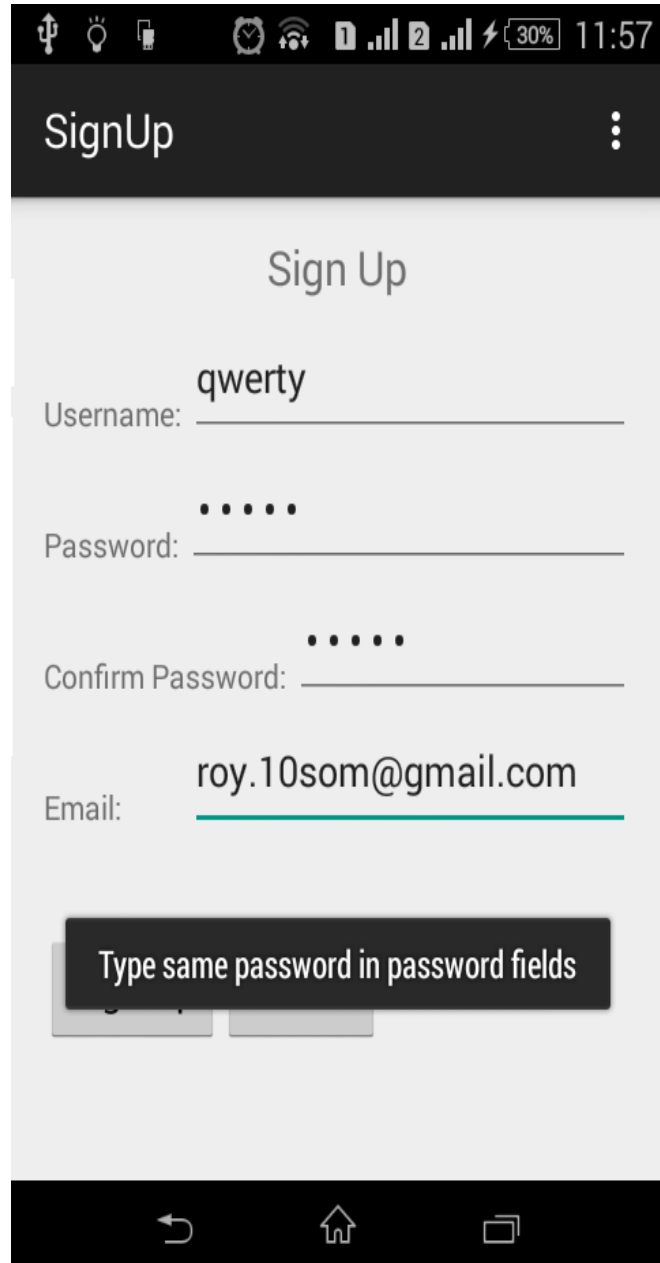
The app is compatible with leading android smartphones and is simple to install.



2.

Sign Up Validation

If both passwords don't match then registration doesn't take place and a message displaying 'Type same password in password fields' is displayed.



The screenshot shows a mobile application interface for a 'Sign Up' screen. At the top, there is a status bar with various icons and the time 11:57. Below the status bar is a dark header with the text 'SignUp' and a three-dot menu icon. The main content area has a light gray background with the title 'Sign Up' centered. Below the title, there are four input fields: 'Username:' with the value 'qwerty', 'Password:' with five dots, 'Confirm Password:' with five dots, and 'Email:' with the value 'roy.10som@gmail.com'. At the bottom of the form, there is a dark gray button with the text 'Type same password in password fields'. The bottom of the screen shows a dark navigation bar with three icons: a back arrow, a home icon, and a document icon.

3.

Username already exists!

If a user registers with an already registered name then also an error message is displayed.

The screenshot shows a mobile application interface for signing up. At the top, there is a status bar with various icons and the time 8:31. Below it is a dark header with the text "SignUp" and a menu icon. The main content area has a light gray background and is titled "Sign Up". It contains four input fields: "Username:" with the value "sabya", "Password:" with five dots, "Confirm Password:" with five dots, and "Email:" with the value "pahari.sabya23@gmail.com". Below the input fields are two buttons: "Sign Up" and "Cancel". At the bottom of the screen, there is a dark navigation bar with three icons. An error message is displayed in a dark gray box with a yellow border at the bottom of the form area: "Your username have been already taken, please try another username".

4.

**Length of username
and password must be
at least 5!!!**

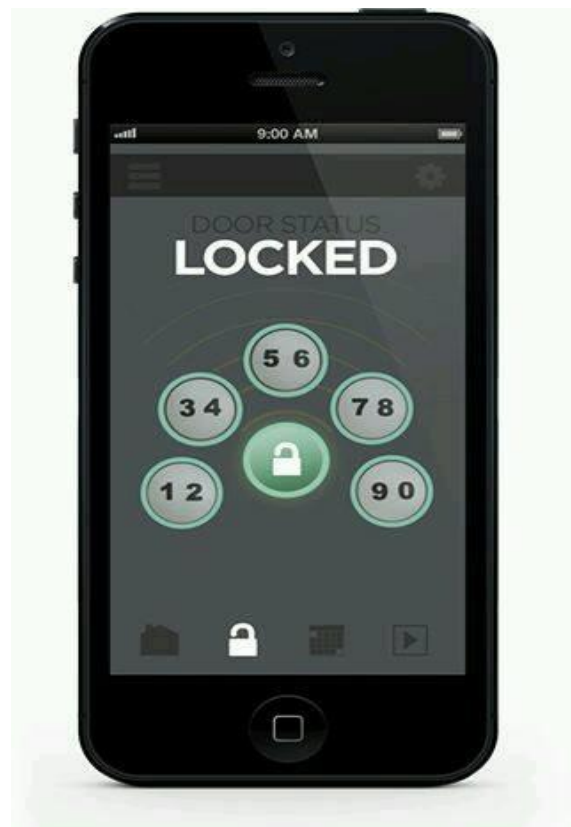
If user registers with a password of length less than five characters then a message will be displayed.

The screenshot shows a mobile application interface for signing up. At the top, there's a status bar with icons for alarm, Wi-Fi, cellular signal, and battery level (30%) at 4:45. Below this is a dark header with the text "SignUp" and a three-dot menu icon. The main content area is light gray and titled "Sign Up". It contains four input fields: "Username:" with the value "somnath12", "Password:" with three dots, "Confirm Password:" with three dots, and "Email:" with the value "roy.10som@gmail.com". At the bottom of the form, there's a dark gray error message box that says "Username and password length must be at least 5". Below this message are two buttons: "Sign Up" and "Cancel". The bottom of the screen features a black navigation bar with three icons: a back arrow, a home icon, and a recent apps icon.

Future Scope of Improvement

- The application can be improved in many ways and can be extended to support more devices like the tablets and iOS devices.
- Android applications at times are device specific. Most important is memory management. The images taken by the commander device and also the MobileControl application requires crisp and richness in its user interfaces which can take up a considerable amount of memory. Thus the biggest challenge is to develop the application giving a rich feel and at the same time having memory optimization. One of the ways in future we can attain this is by making a process of converting some PNG images into JPEG format which reduces the image size by nearly 70 to 80 percent.

- We can build remote locks and switches by which we can lock/unlock doors from anywhere in the world. We will also receive notification when someone rings the doorbell.



- We can add timer and set frequency of capturing image from other devices and also we can interact with the different Android sensors and camera and if someone tries to physically touch or unlock the client's mobile screen we will receive notification instantly.
- One of the most important things we can achieve with the help of this app is that we can get access to someone's current geo position via the Android Location API and display it on Commander's Google Maps.
- Eventually, we can add more hardware control features like sound and video recording, volume control, music player control, viewing contacts and many more.

Conclusion

This is to conclude that the project that we undertook was worked upon with a sincere effort. Our team has worked hard day and night with only one thing in mind i.e. to implement an idea which is new and have not been implemented before. Earlier we saw that the future scope in this project is immense. The requirements which are yet to be fulfilled can be completed within a short extension.

The goals that we have tried to achieve through this project are-

Once two smartphone devices get connected then two devices will be able to remotely control each other via Internet. After successful login into our app users will be able to control flashlight and camera i.e. he/she can turn on the flashlight or can turn it off remotely. He/she can also take images of the person remotely and can view the specified image in his/her gallery.

The minimum requirement for this app to function properly is that both the devices need proper Internet connections. This app is compatible with leading android smartphones and is simple to install.

From the point of view of security we can say that our app is secure and convenient. Although we know that information security plays a vital role in the system work model for ensuring confidentiality and integrity, smooth functioning and reduced misuse of the system. Advanced security features can be employed for enhanced functioning of the application.

We hope that users will find our app desirable and efficient.

Bibliography

1. Websites:

- a. *Introduction to Android:* <http://developer.android.com/guide/index.html>.
- b. *The Java Tutorials:* <http://docs.oracle.com/javase/tutorial/>
- c. http://www.researchgate.net/publication/257947770_Remote_Control_of_Mobile_Devices_in_Android_Platform
- d. <https://chrome.google.com/webstore/detail/chrome-remote-desktop/gbchcmhahfdphkhkmpfmihenigjmpp?hl=en>
- e. <http://lifehacker.com/five-best-remote-desktop-tools-1508597379>
- f. <https://github.com/Pirngruber/AndroidIM>
- g. <http://www.vogella.com/tutorials/android.html>

2. Books:

- a. Android Recipes: A Problem-Solution Approach by D Smith & J Friesen
- b. Hello,Android: Introducing google's Mobile Development Platform(Pragmatic Programmers) by Ed Burnette
- c. Programming Android by Zigurd Mednieks, Laird Dornin, G. Blake Meike & Masumi Nakamura
- d. Java Web Services: Up and Running by Martin Kalin