MINI PROJECT

(2020-21)

Building and Deployment of Android application

Remote Mobile Phone Access

PROJECT REPORT

Institute of Engineering & Technology

Submitted by: -

Kushal Sharma

(181500348)

Pradhum Bansal

(181500460)

Supervised By: -

Ms. Harvinder Kaur

Technical Trainer

Department of Computer Engineering & Applications GLA University

Department of computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan,

DECLARATION

I/we hereby declare that the work which is being presented in the B.Tech. Project **"Remote Mobile Phone Access"**, in partial fulfilment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of Ms. Harvinder Kaur**.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign_____

Name: Kushal Sharma

University Roll No.: 181500348

Sign_____

Name: Pradhum Bansal

University Roll No.: 181500460

Date:

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan,

CERTIFICATE

This is to certify that the project entitled "**Remote Mobile Phone Access**", carried out in Mini Project – II Lab, is a bonafide work by Kushal Sharma (181500348) and Pradhum Bansal (181500460) and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Ms. Harvinder Kaur

Date:

ACKNOWLEDGEMENT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout our mini project to complete the project successfully.

I/we would like to express our deep and sincere gratitude to our college faculties for giving us this opportunity to do a mini project. I/We am extremely grateful to my mentor, Ms. Harvinder Kaur, for her invaluable guidance throughout this mini project. Her dynamism, vision, sincerity, and motivation have deeply inspired us. She has guided us so well. It was a great privilege and honor to work and study under her guidance. I/we are extremely grateful for what she has offered us. I would also like to thank her for her empathy. I/we am extremely thankful to our friends and family for their acceptance and patience during this mini project.

I/We are extremely grateful to our parents for their love, prayers, caring and sacrifices for educating and preparing us for my future.

Sign_____

Name: Kushal Sharma

University Roll No.: 181500348

Sign _____

Name: Pradhum Bansal

University Roll No.: 181500460

ABSTRACT

This project is about making a web application which has multiple uses when you have misplaced your phone. We will make an application with the main purpose of having a phone contact reach your phone even when you don't have your phone with yourself. The application will provide a platform which will help you to send a text message on your phone where you will write the name of your contact as saved in the database of your phone which should be followed by "FIND#" which is because we do not aim for the app to read all the messages. We only want the app to read the messages which have this string "FIND#" written before it. Then the application will revert with the phone number of that person and thus you will be able to access the contact without even your phone being there.

The application is made on android studio and we have used Java programming language for building it. We have successfully run the application already on our mobile phones and have received the messages which contain the contact details of the contact name we had specified in the message we had sent before.

We also aim to further provide extra security to the web application in which there will be a passcode to access your application from some other device which would ensure that there is no one who would interfere in your security and as we all know that any application is not good enough when you don't have proper security in it.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1. GENERAL INTRODUCTION

Here, we have just built an offline android application where people can have access to the contact feature on their mobile phone without even having it with them at that moment. That feature includes their contact list being accessed by the application and finding out the contact the user wants to have at that very moment by traversing through the whole contact list.

It is an offline android application where just by having the application on your phone and some other phone you can have access to the contacts present in the contact list of your phone's features. As we know that sometimes when we do not have our phone with us, there are a load of things which we want to do with it and some of them are important sometimes. Thus, this application was necessary to build up and the icing on the cake is that it is completely offline. This application firstly focuses on the main aim which was to access the phone's contact list which works as when you message a contact's name through this application to your phone, there will be a password which you will have to enter and then the contact's details will automatically be sent to you without any use of internet. There are many more uses which we aim for the application to fulfil in future times like changing the mobile audio profile and even lock the phone when it isn't close enough. The only requirement will be to have the application downloaded on our phone and boom it is done!

1.2. OBJECTIVE

The main objective of this project is to create an android application through which people can have access to their contacts present in their phone's contact list by accessing someone else's mobile phone. It will be an android application developed using android studio platform and Java programming language.

1.3. MOTIVATION

"The beautiful thing about learning is that nobody can take it away from you."

The motivation behind the building of this android application is that we all know how important it is for all of the people to have access to their mobile phones at all times presently in today's era. So, when someone has forgotten his/her phone at some place and needs some contact details

immediately then this application will come into play and will completely make him/her a relaxed person and will easily return the number to the concerned person just after the person will send a text to the person by adding “FIND#” before the contact’s name.

1.4. FEATURES

a) The application designed is interactive and user friendly.

b) Simple and easy to access.

c) Easy to modify and add new features

1.5. TECHONOLOGIES USED

1.5.1. Android Studio

Android Studio is the official integrated development environment (IDE) for Google’s Android operating system; built on JetBrains’s IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating system or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for the native Android application development.
Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first build was released in December 2014, starting from version 1.0.
On May 7, 2019, Kotlin replaced Java as the preferred language of Google for Android development. Java is still supported, so is C++.

Features of Android Studio:

Following are some of the important features that make Android Studio so alluring for all the computer science enthusiasts:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities

- Template-based wizards to create common Android designs and

components

- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

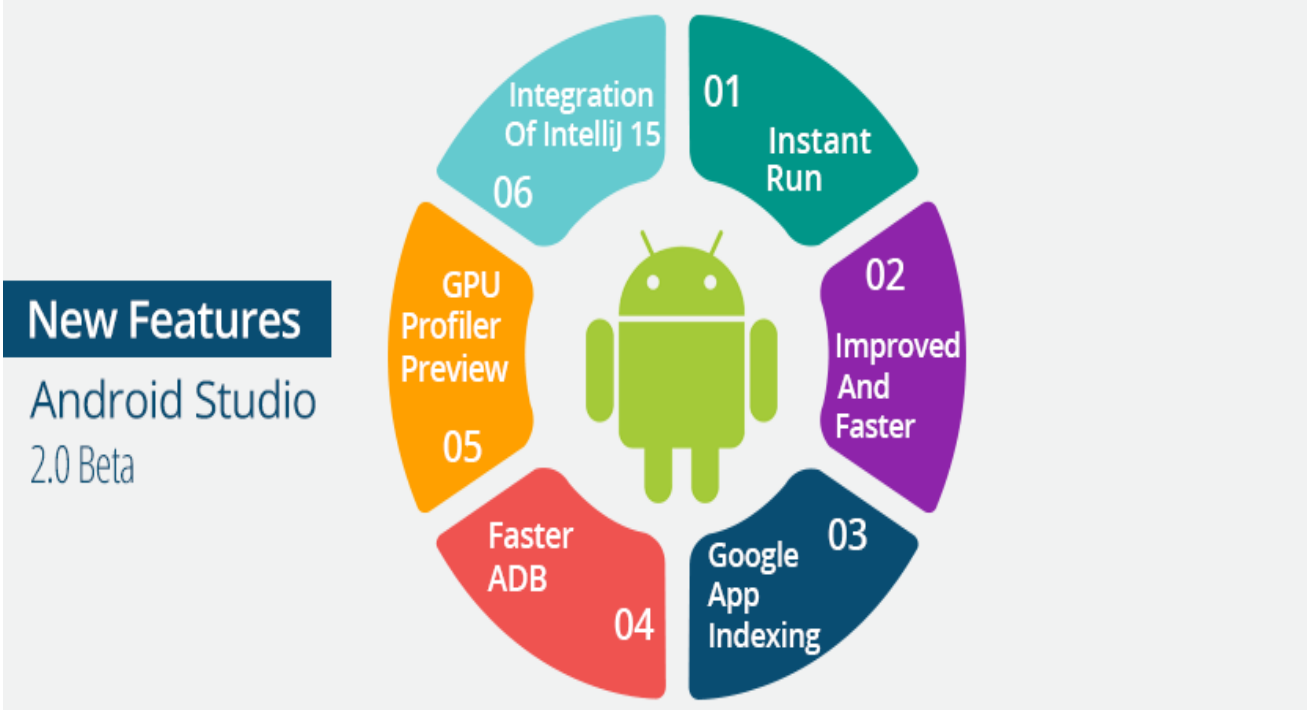


Figure 1 Features: Android Studio

1.5.2. Java

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is like C and C++ but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.
Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Oracle offers its own Hotspot Java Virtual Machine; however, the official reference implementation is the OpenJDK JVM which is free open source software and used by most developers and is the default JVM for almost all Linux distributions.
As of March 2021, the latest version is Java 16, with Java 11, a currently supported long-term support (LTS) version, released on September 25, 2018. Oracle released the last zero-cost public update for the legacy version Java 8 LTS in January 2019 for commercial use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Other vendors have begun to offer zero-cost builds of OpenJDK 8 and 11 that are still receiving security and other upgrades.
Oracle (and others) highly recommend uninstalling outdated versions of Java because of serious risks due to unresolved security issues. Since Java 9, 10, 12, 13, 14, and 15 are no longer supported, Oracle advises its users to immediately transition to the latest version (currently Java 16) or an LTS release.

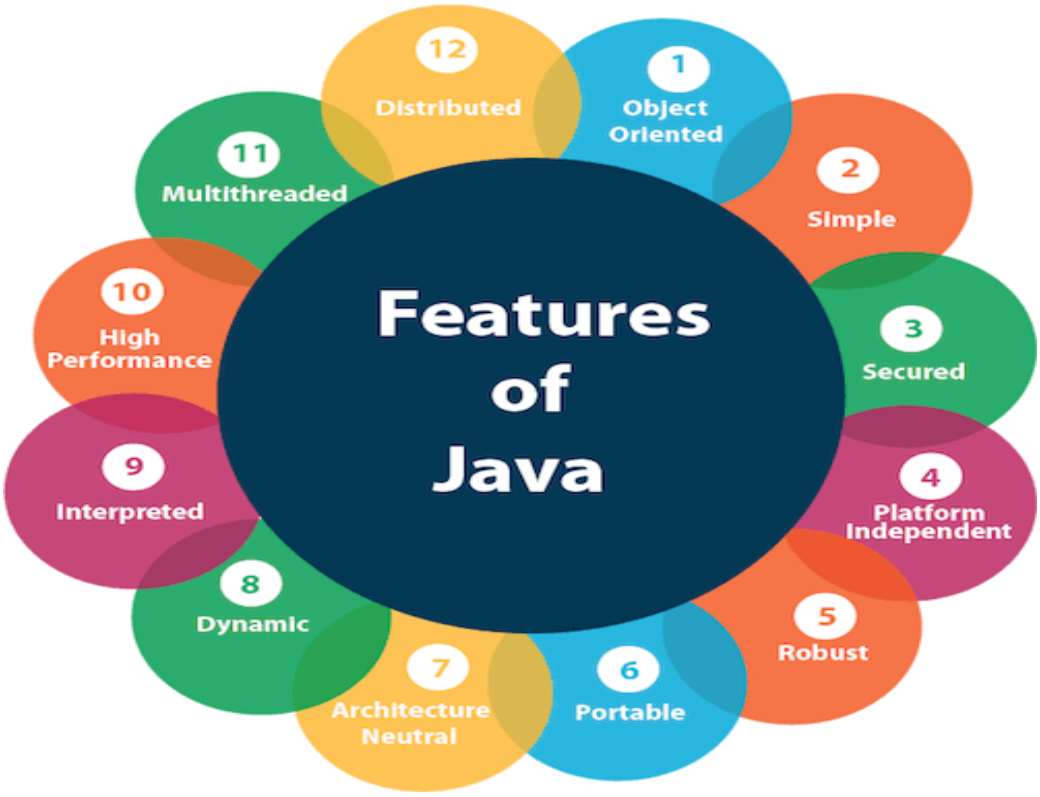


Figure 2 Features: Java

1.6. SCOPE

This project has been created after keeping the present living state of human beings in mind. We all know that living without our mobile phones isn't possible these days and there are times when we desperately need our phone to have some contact details but we have forgotten our phone somewhere and now are in need of it desperately. This is where this app comes into play. We can use this application to retrieve the contact details of that specific person just by adding a string "FIND#" before the contact's name. There are many more features we plan to add in this application in the future which also includes adding the security key so that it is much more useful to today's world.

2. SOFTWARE REQUIREMENT ANALYSIS

2.1. INTRODUCTION

The aim of this part is to gather and analyze and give an in-depth insight of the complete **REMOTE MOBILE PHONE ACCESS** by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of **REMOTE MOBILE PHONE ACCESS PROJECT** are provided in this document.

2.1.1. Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this report document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

2.1.2. Document Convention:

In this text, it will use font small 2 and overstriking for primary title, font small 3 for secondary title and font 4 for the content. And it will use the italic when mentions the name of the application **REMOTE MOBILE PHONE ACCESS**.

2.1.3. Intended Audience:

This SRS about **REMOTE MOBILE PHONE ACCESS** is for developers, mentors, users and testers. The article mainly introduces the overall description, external interface requirements, system features and other non-functional requirements. I suppose mentor to read the whole article carefully and user pay attention to overall description especially. Users and testers read the system features carefully.

2.1.4. PERSPECTIVE:

A web application preferably using the Android Studio platform, Java Programming language and SQLite for database.

2.2. PRODUCT FUNCTIONS:

Help the users to use this application to access their contacts from their phone's contact list without even their phone being with them.

This is simply an android application which will provide the user with an environment in which a login page will appear as soon as the user land on the application. It shows up with a register button for the users using the application for the first time.

After registering, the main messaging page would appear on the android phone's screen. There the user can message anyone as there are 2 input fields there out of which 1 is phone number and the other is the message you want to send. Even if the user doesn't want to find the contact detail of someone, still he/she can use this application as a simple messaging application as well.

2.3. USER CLASSES AND CHARACTERISCTICS:

Our android application mainly targets today's generation people and broadly speaking, everyone in today's world who uses android phones. There are people who are using phones at almost all times of the day and as we all know that there are instances where we forget our mobile phone at home or somewhere else and then sometimes we need some contact in an urgent manner and then is the time when this android application will come into use.

2.4. OPERATING ENIRNOMENT:

Our software is a multi-functional software system based on the android platform.

2.5. DESIGN CONSTRAINS:

Our application firstly prompts the user to give the permission to the application to use the contacts and also to send and read the messages. We must consider about the arrangement and beautification of the interface; Prioritization of processing operations and it deepens the difficulty of coding and testing. This application needs users to register in the application first and update their existence in the database of the application, only then will they be allowed to move forward and use the application. Every time, the user logs in he/she will be shown the messaging screen which will consist of 2 different text fields which will be mandatory to fill the details in. One of them is the phone number and the other one being the message, the user wants to send on that phone number.

2.6. ASSUMPTIONS AND DEPENDENCIES:

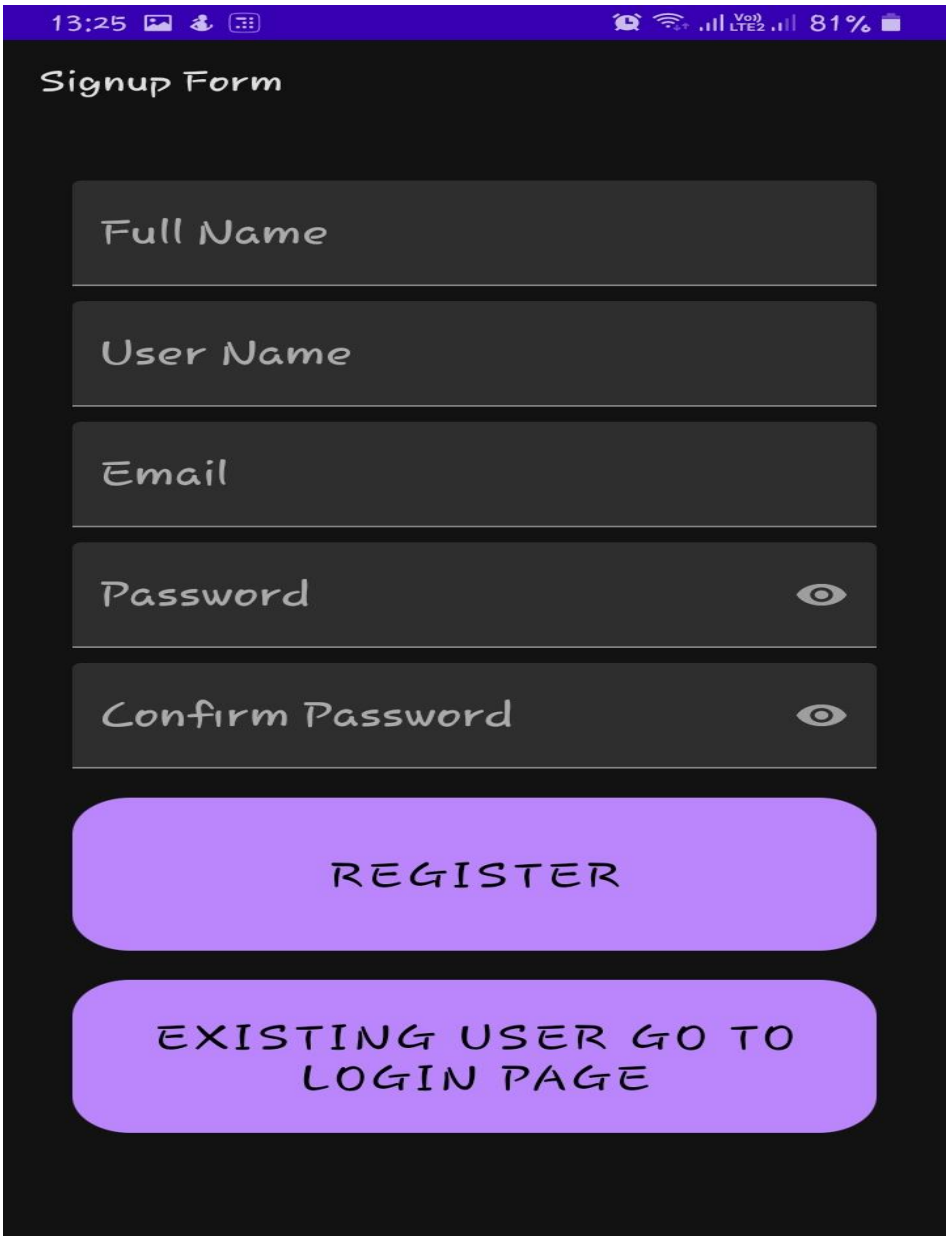
REMOTE MOBILE PHONE ACCESS is an android app which has been developed using android studio and Java has been used for the programming purpose.

2.7. FUNCTIONAL REQUIREMENT:

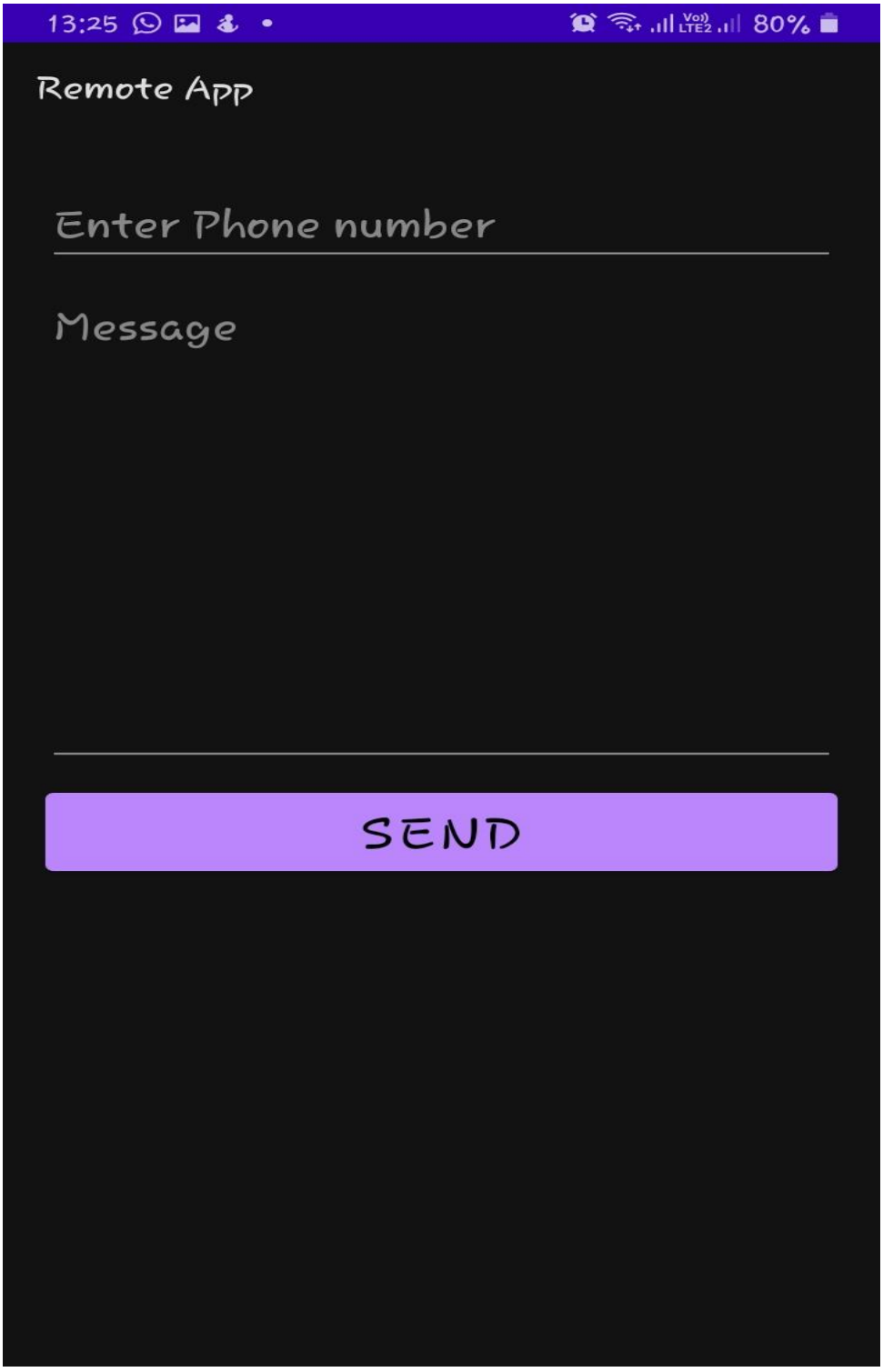
2.7.1. User Interfaces:

UI-1: As soon as the user lands on the android application, he/she will encounter a login page which will prompt the user to login to the app using his/her username and password which is already stored in the SQLite database. If the user has come first time, then there is also a button for registering the user and that button will take the user to another UI which will be the register page and will prompt user to enter the detail.

UI-2: This page is the register page which will prompt the user to enter all the information which would include the Full Name, User Name as the user wants, the Email ID of the user, the password and also there is another field for re-entering the password for confirming it. There are two buttons on this page out of which one is for confirming the registered details and another one is for those users who are already registered and have arrived at this page by mistake and that button states “EXISTING USERS GO TO LOGIN PAGE”.



UI-3: Once the user is done with all the process of registration and logging in, he/she lands on the main UI which has a couple of text input fields, out of which 1 is for the phone number on which you want to send the message and the other one is for the message you want to send to that specified phone number.



2.8. PERORMANCE REQUIREMENTS:

Since the application is completely offline, there is absolutely no dependency on the phone's internet connection. If the user wants to access the contact then there is just one need that is there should be a contact with that name there or a message as " The contact number of this name does not exist in this phone.", would return back to the android phone.

2.8.1. Hardware Requirements:

The requests of the hardware for the web application are as followed:

- Android Phone

2.8.2. Software Requirements:

To access this application, its only needs an Android Mobile with an android version above KitKat.

Mobile phone: RAM above 2GB and storage above 1GB

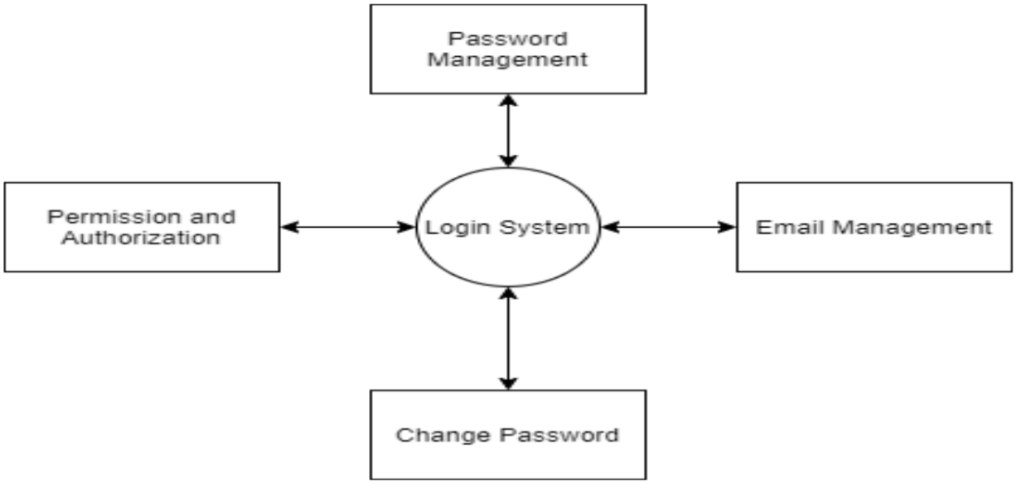
Free space: 30MB.

3. SOFTWARE DESIGN

Function Oriented Design for procedural approach and different diagram to show the designing of the application.

3.1. DATAFLOW DIAGRAM

3.1.1. DFD Level 0



4. TESTING

4.1. Android Testing

Android Studio is designed to make testing simple. With just a few clicks, you can set up a JUnit test that runs on the local JVM or an instrumented test that runs on a device. We can also extend our test capabilities by integrating test frameworks such as 'Mockito' to test Android API calls in your 'local unit tests', and 'Espresso' or 'UI Automator' to exercise user interaction in your 'instrumented tests'. You can generate Espresso tests automatically using 'EspressoTestRecorder

4.2.1 Local Test Units

Located at **module-name**/src/test/java/.

These are tests that run on your machine's local Java Virtual Machine (JVM). Use these tests to minimize execution time when your tests have no Android framework dependencies or when you can mock the Android framework dependencies.

At runtime, these tests are executed against a modified version of android.jar where all final modifiers have been stripped off. This lets you use popular mocking libraries, like Mockito.

4.5.3 Instrumented test

Located at **module-name**/src/androidTest/java/

These are tests that run on a hardware device or emulator. These tests have access to Instrumentation APIs, give you access to information such

as the Context of the app you are testing, and let you control the app under test from your test code. Use these tests when writing integration and functional UI tests to automate user interaction, or when your tests have Android dependencies that mock objects cannot satisfy.

Because instrumented tests are built into an APK (separate from your app APK), they must have their own AndroidManifest.xml file. However, Gradle automatically generates this file during the build so it is not visible in your project source set. You can add your own manifest file if necessary, such as to specify a different value for `minSdkVersion` or register run listeners just for your tests. When building your app, Gradle merges multiple manifest files into one manifest.

5.1. DEPLOYMENT

The flask app is deployed on Heroku. Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages.

To deploy this, react app on Heroku. We had used the website of Heroku, after logging in to the website mentioned below, we proceeded with creating a new app and connecting it with our GitHub repository.

https://dashboard.heroku.com/apps

After this we proceeded with the user interface of Heroku platform and after its execution and connection process was finished, we deployed our application.

How to Run the app:

1. Open Android Studio.
2. Clone the repository by entering

```
$ git clone https://github.com/KushS20/Remote-Mobile-Phone.git.
```

3. Ensure that Java is installed on the system.
4. Connect your android phone using a USB cable.

5. Open the developer settings of your Android phone and turn on USB debugging.
6. Now your phone name will be displayed on the Android Studio.

7. Now enter the run button and the app will get installed on your phone.

6. CONCLUSION AND FUTURE WORK

6.1. CONCLUSION

The main purpose of our project is to develop an application that offers new aspects of development in technical area. Most of the available apps are entertainment-based, which mostly do not contribute in helping people in daily life. The theme of our application is to provide user an app which can help people in emergency. This application is useful for people whose contacts are important in their daily life.

This application allows the user to register and login depending on the situation that you are a new user or an existing one. Then the app will ask you to red your contacts. This app will help you in emergency, like if you have forgotten your phone at your home and you are in need of a contact. You just have to borrow your friend's phone and send a simple text message with the keyword "FIND#name" and you will get the number of the contact within no time.

6.2. FUTURE WORK

We are planning to keep managing the project and improving it based on user feedback. Here is our to do list for future

- We will add some more categories in our app.
- We'll try to implement more features.
- We'll try to improve its security quality by adding a security key to it as well.
- We'll work on another feature in our app if you have kept your somewhere so you can ring your phone if even it is in silent mode.

7. REFERENCES

1. https://github.com/
2. Geeksforgeeks.com
3. Developer.android.com
4. W3schools.com

APPENDIX

Source Code –

- Online-Examination-System app: https://github.com/KushS20/Remote-Mobile-Phone