

**An Industry Oriented Project Report on
“Vehicle Tracking Using Driver’s Mobile”**

Submitted to the

**JAWAHARLAL NEHRUTECHNOLOGICALUNIVERSITY
HYDERABAD**

In partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

Ravi Sai Alekhya

19WJ1A05Q6

Under the Esteemed Guidance Of

Mr. K. Raveendra Kumar

Assistant Professor, CSE Dept.



GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (AUTONOMOUS)

School of Engineering and Technology

Ibrahimpattanam, R.R District-501506

2022-2023



GURU NANAK INSTITUTIONS TECHNICAL CAMPUS



Approved by
AICTE - New Delhi



Affiliated to
JNTU - Hyderabad



Accredited by
National Assessment and
Accreditation Council



Accredited by
National Board of
Accreditation

AUTONOMOUS
under Section 2 (f) & 12 (b) of
University Grants Commission Act

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that this project report entitled “**VEHICLE TRACKING USING DRIVER’S MOBILE**” by **Ravi Sai Alekhya (19WJ1A05Q6)** submitted in partial fulfillment of the requirements for the degree of **Bachelor of Technology in Computer Science and Engineering** of the **Jawaharlal Nehru Technological University Hyderabad** during the academic year 2022-2023, is Bonafide record of work carried out under our guidance and supervision.

INTERNAL GUIDE

Mr. K. Raveendra Kumar

PROJECT COORDINATOR

Mrs.V.Swathi

HOD CSE

Mr.Devasekhar

EXTERNAL EXAMINER



RAM Innovative Infotech

M : +91 9581 012 012

E : raminnovativeinfotech@gmail.com

Flat No.#309, Amrutha Ville,
Opp: Yashoda Hospital, Somajiguda,
Hyderabad-82, Telangana, India

www.raminnovativeinfotech.webs.com

PROJECT COMPLETION CERTIFICATE

This is to certify that the following student of final year B.Tech, Department of “**COMPUTER SCIENCE AND ENGINEERING**”- Guru Nanak Institutions Technical Campus (GNITC) has completed his training and project at GNITC successfully.

STUDENT NAME:

Roll No:


Ravi Sai Alekhya

19WJ1A05Q6

The training was conducted on “**MOBILE APPLICATION DEVELOPMENT**” Technology for the completion the project titled “**VEHICLE TRACKING USING DRIVER’S MOBILE**” In JANUARY, 2023 The project has completed in all aspects.



Mini & Major IEEE Live Projects For
M.E /M.Tech, B.E / B.Tech, MCA, M.S,



ACKNOWLEDGEMENT

I wish to express my sincere thanks to **Dr. Rishi Sayal**, Professor and Associate Director of CSE GNITC for providing us the conducive environment for carrying through our academic schedules and Project with ease.

We have been truly blessed to have a wonderful adviser **Mr.V. DEVASEKHAR**, Professor HOD of CSE, GNITC for guiding us to explore the ramification of our work and we express our sincere gratitude towards him for leading me through the completion of Project.

We would like to say our sincere thanks to **Mrs.V.Swathi**, Associate Professor, Department of CSE, Mini Project Coordinator, for providing seamless support and right suggestions are given in the development of the project.

We specially thank our internal guide **Mr. K. Raveendra Kumar**, Assistant Professor, Department of CSE, for his suggestions and constant guidance in every stage of the project. Finally, we would like to thank our family members for their moral support and encouragement to achieve goal.

Ravi Sai Alekhya (19WJ1A05Q6)

Vehicle Tracking Using Driver's Mobile

ABSTRACT

The Global Positioning System (GPS) is a space-based satellite navigation system that provides location information. This system uses GPS to track the location of the vehicle. This system will track location of the vehicle and will send details about the location to the admin. This system helps admin to find out the location of the driver driving the vehicle. Admin will know which driver is in which location. This system can be implemented in call taxi to find out the location of the driver driving the vehicle and will help the admin to allocate taxi to the customer. User login to the system using his user id and password. System will track the user's location with the help of GPS and will send this detail to admin. Admin will access the system using his admin ID and password and will view the location of the driver driving the vehicle based on these details he will perform various operations. This application helps the admin to find the location of various drivers. He can keep record of the driver's attendance that will help him to calculate salary of the driver very easily. Driver cannot do any type of cheating otherwise admin can easily find out using this system.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	TABLE OF CONTENTS	ii
	LIST OF FIGURES	v
	LIST OF SYMBOLS	vi
	LIST OF ABBREVIATIONS	ix
1.	CHAPTER 1: INTRODUCTION	
	1.1 GENERAL	1
	1.2 OBJECTIVE	3
	1.3 EXISTING SYSTEM	4
	1.4 LITERATURE SURVEY	5
	1.5 PROPOSED SYSTEM	11
2.	CHAPTER 2: PROJECT DESCRIPTION	
	2.1 GENERAL	12
	2.2 METHODOLOGIES	12
	2.2.1 MODULES NAME	12
	2.2.2 MODULES EXPLANATION	13
	2.3 TECHNIQUE OR ALGORITHM	18
	2.3.1 EXISTING TECHNIQUE	18
	2.3.1 PROPOSED TECHNIQUE	19
3.	CHAPTER 3: REQUIREMENTS ENGINEERING	
	3.1 GENERAL	20
	3.2 HARDWARE REQUIREMENTS	20
	3.3 SOFTWARE REQUIREMENTS	21
	3.4 FUNCTIONAL REQUIREMENTS	22
	3.5 NON-FUNCTIONAL REQUIREMENTS	23
4.	CHAPTER 4: DESIGN ENGINEERING	
	4.1 GENERAL	24
	4.2 UML DIAGRAMS	25

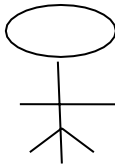
	4.2.1 USE CASE DIAGRAM	25
	4.2.2 CLASS DIAGRAM	26
	4.2.3 OBJECT DIAGRAM	27
	4.2.4 COMPONENT DIAGRAM	28
	4.2.5 DEPLOYMENT DIAGRAM	29
	4.2.6 SEQUENCE DIAGRAM	30
	4.2.7 COLLABORATION DIAGRAM	31
	4.2.8 STATE DIAGRAM	32
	4.2.9 ACTIVITY DIAGRAM	33
	4.2.10 DATA FLOW DIAGRAM	34
	4.3 SYSTEM ARCHITECTURE	36
5.	CHAPTER 5: DEVELOPMENT TOOLS	
	5.1 GENERAL	37
	5.1.1 PYTHON	37
	5.1.2 HISTORY OF PYTHON	37
	5.1.3 IMPORTANCE OF PYTHON	37
	5.1.4 FEATURES OF PYTHON	38
	5.1.5 LIBRARIES USED IN PYTHON	39
6.	CHAPTER 6: IMPLEMENTATION	
	6.1 GENERAL	40
7.	CHAPTER 7: SNAPSHOTS	
	7.1 GENERAL	46
8.	CHAPTER 8: SOFTWARE TESTING	
	8.1 GENERAL	52
	8.2 DEVELOPING METHODOLOGIES	52
	8.3 TYPES OF TESTING	52
	8.3.1 UNIT TESTING	52
	8.3.2 FUNCTIONAL TESTING	53
	8.3.3 SYSTEM TESTING	53

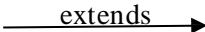

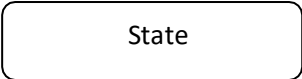
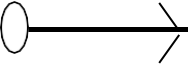
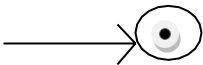
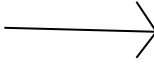
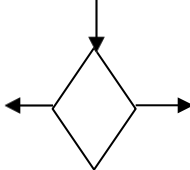
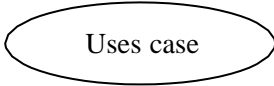
	8.3.4 PERFORMANCE TESTING	53
	8.3.5 INTEGRATION TESTING	53
	8.3.6 ACCEPTANCE TESTING	53
	8.3.7 BUILD THE TEST PLAN	54
9.	CHAPTER 9: FUTURE ENHANCEMENT 9.3 GENERAL	55
10.	CHAPTER 10: CONCLUSION AND REFERENCES 10.1 CONCLUSION 10.2 REFERENCES	56 57

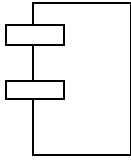
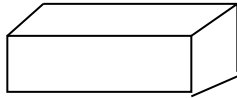
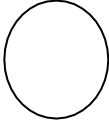


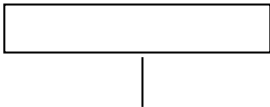
LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
2.3.2	Module Diagram	3
4.2	Activity Diagram	12
4.3	Use case Diagram	7
4.4	Data flow diagram	13
4.5	Sequence diagram	10
4.6	Collaboration diagram	11
4.7	Class diagram	8
4.8	Architecture Diagram	16
4.9	State Diagram	9
4.1	Component Diagram	8
4.12	E-R Diagram	15

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 20px;"> <i>+ public</i> <i>-private</i> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <i>Class Name</i> </div> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px; text-align: center;"> <i>-attribute</i> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px; text-align: center;"> <i>-attribute</i> </div>	Represents a collection of similar entities grouped together.
2.	Association	<div style="display: flex; align-items: center; justify-content: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin: 0 10px;">Class A</div> <div style="border-bottom: 2px solid black; width: 50px; margin: 0 10px; position: relative;"> NAME </div> <div style="border: 1px solid black; padding: 5px; margin: 0 10px;">Class B</div> </div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin: 0 10px;">Class A</div> <div style="border-bottom: 1px solid black; width: 50px; margin: 0 10px;"></div> <div style="border: 1px solid black; padding: 5px; margin: 0 10px;">Class B</div> </div>	Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation	<div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> Class A Class B </div> <div style="text-align: center;"> Class A Class B </div> </div>	Interaction between the system and external environment

5.	Relation (uses)	Uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processes.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.

14.	Component		Represents physical modules which are a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		i. Represents the vertical dimensions that the object communications.
20.	Message	Message	Represents the message exchanged.

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1.	DB	DataBase
2.	JVM	Java Virtual Machine
3.	JSP	Java Server Page
4.	PWS	Personalised Web Search
5.	UPS	User Personalised Search
6.	JRE	Java Runtime Environment

CHAPTER 1

INTRODUCTION

1.1 GENERAL

In this paper, an efficient vehicle tracking system is proposed that is used to track the location of any equipped vehicle with the help of an android application. The SIM808 module is used which incorporates GPS and GSM modules. The GPS module is used to get vehicle's location coordinates. These coordinates are sent to the Smartphone with the help of a GSM module. On a Smartphone, an android based application is developed to plot the location of a vehicle on a Google map. The application also has the capability to keep the previous route history of the vehicle. The proposed system is implemented and tested in a practical environment. Experimental results have proved that the proposed tracking system is accurate and feasible for the vehicle owners.

1.2 OBJECTIVE :

GPS is a system which is already implemented and everyone can access it without any restriction. Having the facility of GPS to develop this system we need a GPS device to calculate the location from the information taken from GPS. Hence, we have chosen Android device to perform this calculations because Android mobile phone is cost effective and offers multidimensional purposes having some special built-in features like GPS service.

1.3 Existing System:

GPSs are very quickly becoming a standard in most new automobiles, and are even finding their way onto a variety of new cell phones. The mapping devices can come in handy under a variety of circumstances. If we are thinking about purchasing a GPS or a device that has a GPS built-in, here are some of the benefits of having one. A GPS can help us to determine exactly where we are at any given moment. Not only can a GPS give us the name of the street we might be traveling on, we have to use a Gps Device in the vehicle .

Existing System Disadvantages:

- Cost of GPS Device
- Maintenance of GPS Device

Literature Survey:

Title: GPS-based Location Tracking System via Android Device

Author: Palash Uddin, Md. Zahidul Islam

Year: 2016

Description:

With recent technological advancement of modern science people are now expecting the information about the location of any object for tracking purposes. Presently, we want more location based services for being advanced and to save time and money also. GPS is a system which is already implemented and everyone can access it without any restriction. Having the

facility of GPS to develop this system we need a GPS device to calculate the location from the information taken from GPS. Hence, we have chosen Android device to perform this calculations because Android mobile phone is cost effective and offers multidimensional purposes having some special built-in features like GPS service. Thus, this system is developed for location tracking of a group of people with a proximity alert system using various latest demanding tools and technology like Jason, Java, AVD, LAMP etc

Title: Mobile Tracking System using Web Application and Android Apps

Authors: Shashika Lokuliyana , Anuradha Jayakody ,G.S.B.Dabarera , R.K.R.Ranaweera,

Year: 2018

Description:

Parents are very much concerned about their children now a days and are compelled to provide an Android phone to their children. They want to monitor children activities and movements, where they are going and what they are doing. But this process is difficult. To solve this problem a Mobile Tracking System has been designed, using this system parents can track children's location. An Android application (Mobile Tracking System) needs to install in mobile phone of the children and parents have to register this application with a password. After completing the registration, parents can monitor mobile phone's location of their children through a web site (www.mobiletrackerbd.com) from anywhere. Parents can track children's mobile phone's location by two methods, one is manually and another is automatically. In manual system children have to click a button of the apps for sending mobile phone's location. In automatic system parents have to set a time period into the apps and then apps automatically send mobile phone's location periodically as setting time. Parents can track last and previous location of their children's mobile phone.

1.2 Proposed System

A user using the application owns his own user name and pass-word so that no other person can enter on the application and create confusion to the admin about the location of the user. The user enjoys safety in a dangerous place against being lost as he/she is leaving his/her footprints on a website that is being viewed by the admin. So if there any mis-communication occurs, the admin can get there soon to help him/her. Additionally, if the user wants to set an alarm system based on location, then the proximity alert system can help him/her. It allows a user to manage his/her tasks better. Every user who will be tracked must

Advantages:-

- This system helps admin to keep track of the driver so that driver cannot do any type of cheating.
- This system helps admin to keep record of attendance of the driver which helps in calculating salary of the driver.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL:

The Global Positioning System (GPS) is a utility that provides users with positioning, navigation, and timing services. This system consists of three segments: the space segment, the control segment, and the user segment. The space segment consists of a nominal constellation of 24 operating satellites that transmit one-way signals that give the current GPS satellite position and time. The Control Segment tracks the GPS satellites, uploads updated navigational data, and maintains health and status of the satellite constellation. The user segment consists of the GPS receiver equipment and uses the transmitted information to calculate the user's three dimensional position and time

2.2 METHODOLOGIES

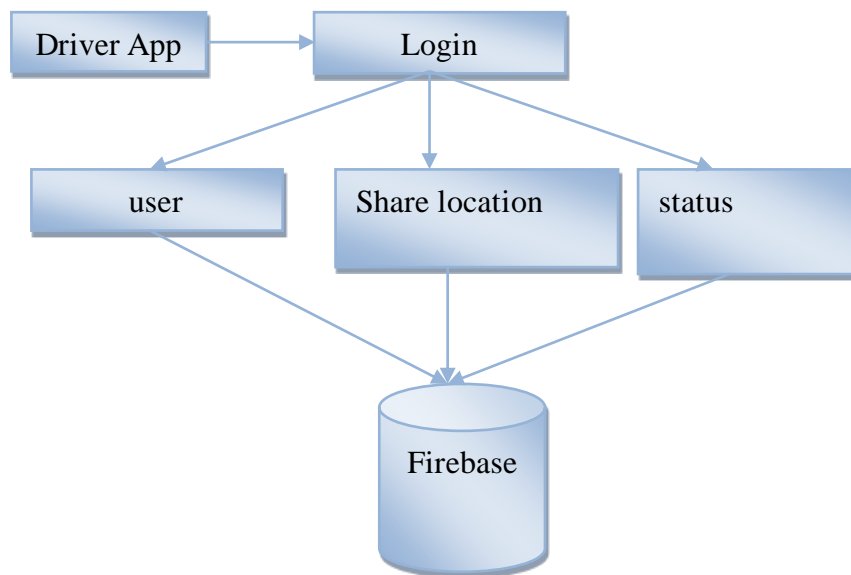
Driver

Admin

Geolocation

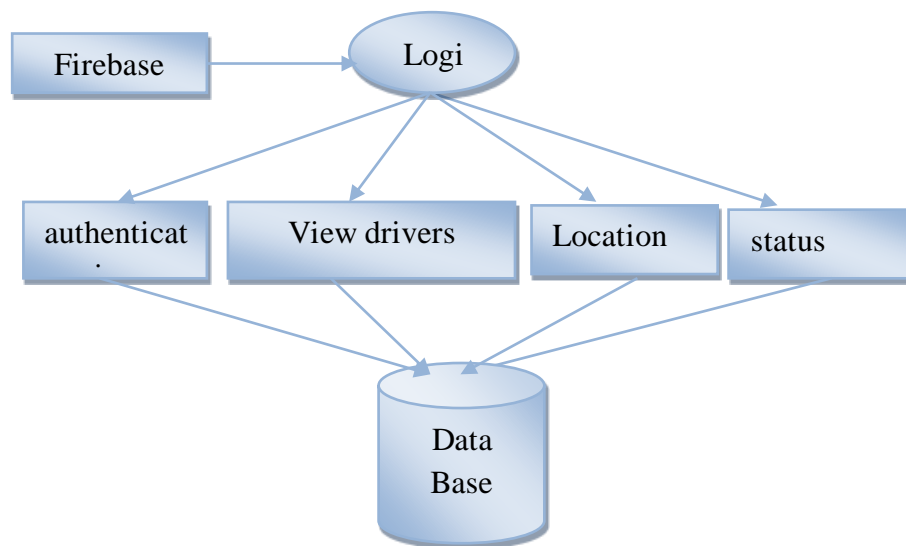
2.2.1 MODULES NAME:

Driver: User can login with user ID and password and share is geo location to the admin

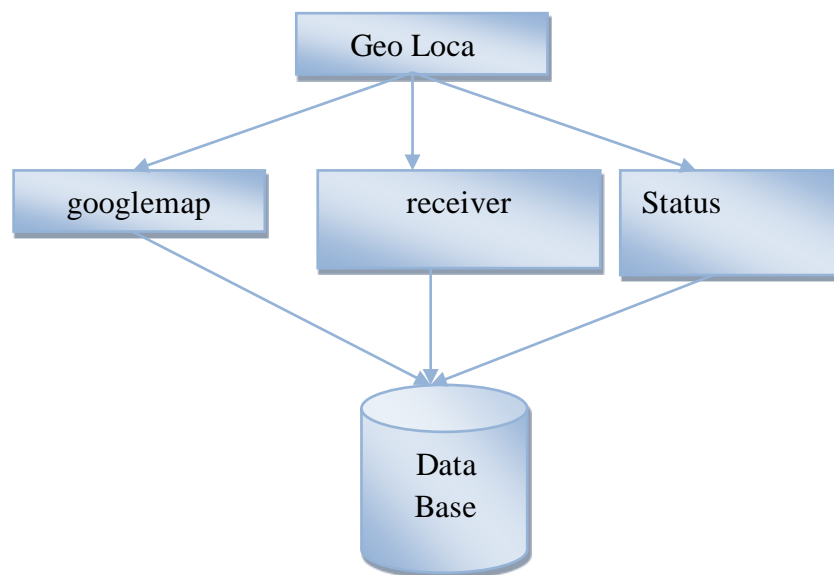


Admin:

- Admin can view the location of driver and alert the driver . we can view route map of driver. Admin can view vehicle and driver location.



GeoLocation: User use Geomap and google libraries to share the location of user



CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

The Global Positioning System (GPS) is a utility that provides users with positioning, navigation, and timing services. This system consists of three segments: the space segment, the control segment, and the user segment. The space segment consists of a nominal constellation of 24 operating satellites that transmit one-way signals that give the current GPS satellite position and time. The Control Segment tracks the GPS satellites, uploads updated navigational data, and maintains health and status of the satellite constellation. The user segment consists of the GPS receiver equipment and uses the transmitted information to calculate the user's three dimensional position and time

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should state what the system does and not how it should be implemented.

- Processor – i3
- Hard Disk – 5 GB
- Memory – 1GB RAM
- Android Phone with Kitkat and higher

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

SOFTWARE REQUIREMENTS

- Windows XP, Windows 7(ultimate, enterprise)
- Android Studio
- Java

3.4 FUNCTIONAL REQUIREMENTS

- A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, and outputs.
- The system has an admin login that has overall control over it. Admin feeds the issues or arguments in the system along with desired options.

- It provides authentication to individuals who wish to get connected. The registration details are stored in the database and whenever the Employees logs in, the employees credentials are retrieved to check whether the employee is an authorized employee or not.
- If the details entered do not match with any of the existing data then the system displays a warning.
- These questions can then be visible to all the employees through android devices. Employees have to first create an account into the system for casting their votes.
- At the end of the voting process the system counts all the votes casted and generates a brief report of the total votes accounted for yes, no and neutral. Eventually, the report is made available to admin and he may view the maximum votes casted for.
- Hence the system helps admin to receive appropriate response from employees for the matters in question.

3.5 NON-FUNCTIONAL REQUIREMENTS

- Portability: It should run on specified platforms successfully. To achieve this we should test the product on all platforms before launching the product. If our project runs successfully on different platforms then our system is portable in nature.
- Reliability: The system should perform its intended functions under specified conditions. If our system satisfies all the specified conditions then it is Reliable in nature.
- Reusability: The system should be extremely reusable as a whole or part. Make the system modularize and make sure that modules are loosely coupled. This project is having reusability nature because we can reuse whole or part of this project on other systems.
- Robustness: The system on the whole should be robust enough to perform well under different circumstances without any inconsistencies.
- Testability: The product of a given development phase should satisfy the conditions imposed at the start of that phase.
- Usability: It should be perfect and comfortable for users to work.
- Security: The system is completely based on the security. This system will provide security base on the password.

CHAPTER 4

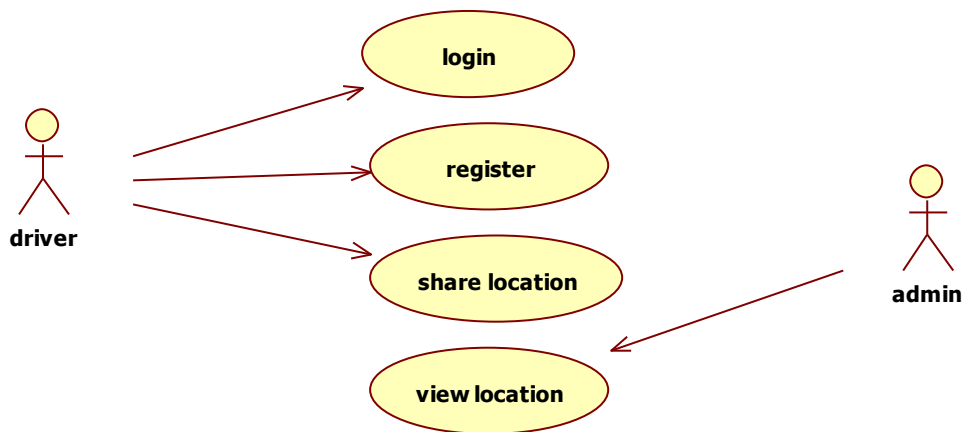
DESIGN ENGINEERING

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

UML Diagrams

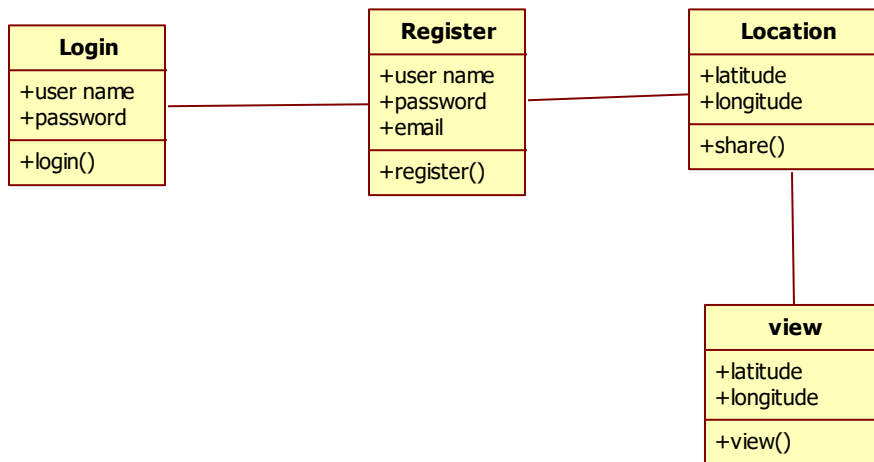
Use Case diagram



EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

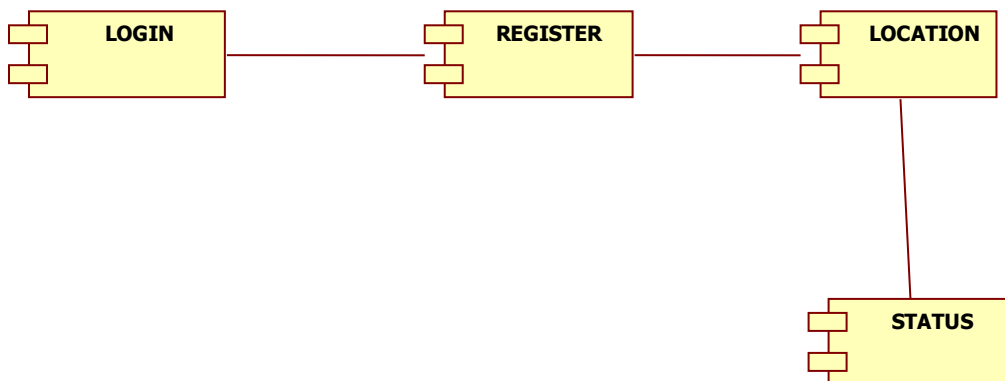
Class Diagram



EXPLANATION

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

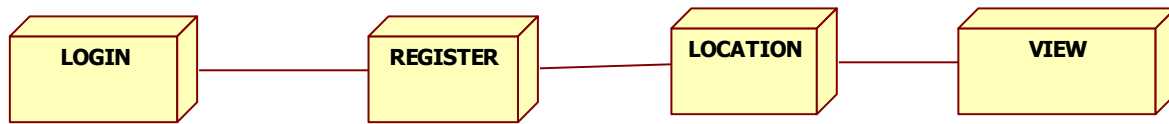
Component Diagram



EXPLANATION:

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

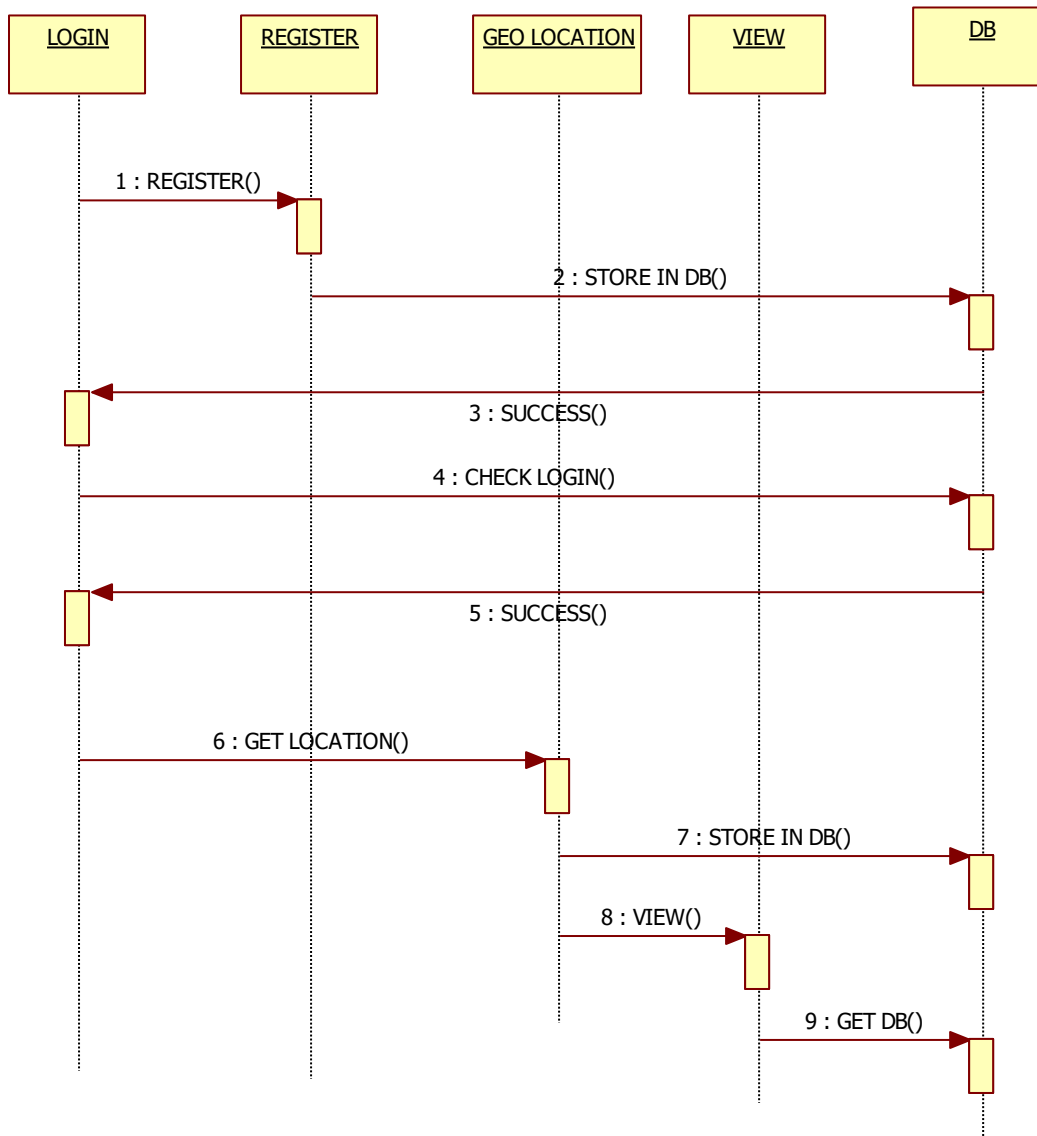
Deployment Diagram



EXPLANATION:

In the Unified Modeling Language, a deployment diagram depicts how deploys are wired together to form larger deployment and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

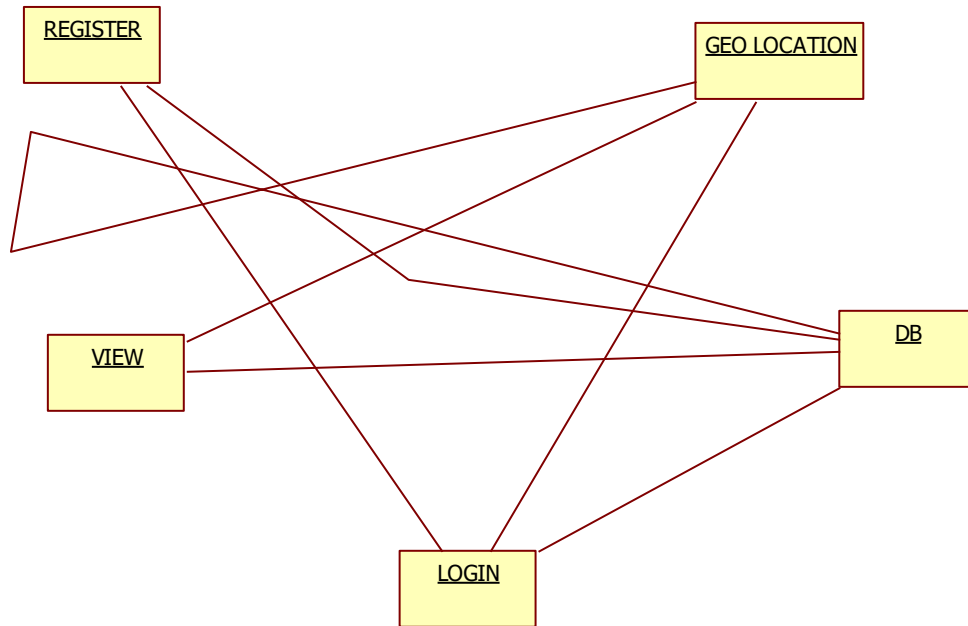
Sequence Diagram



EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

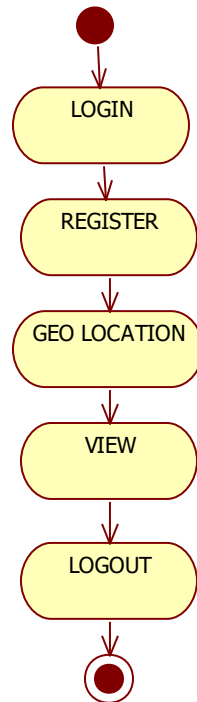
Collaboration Diagram



EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved

Activity Diagram

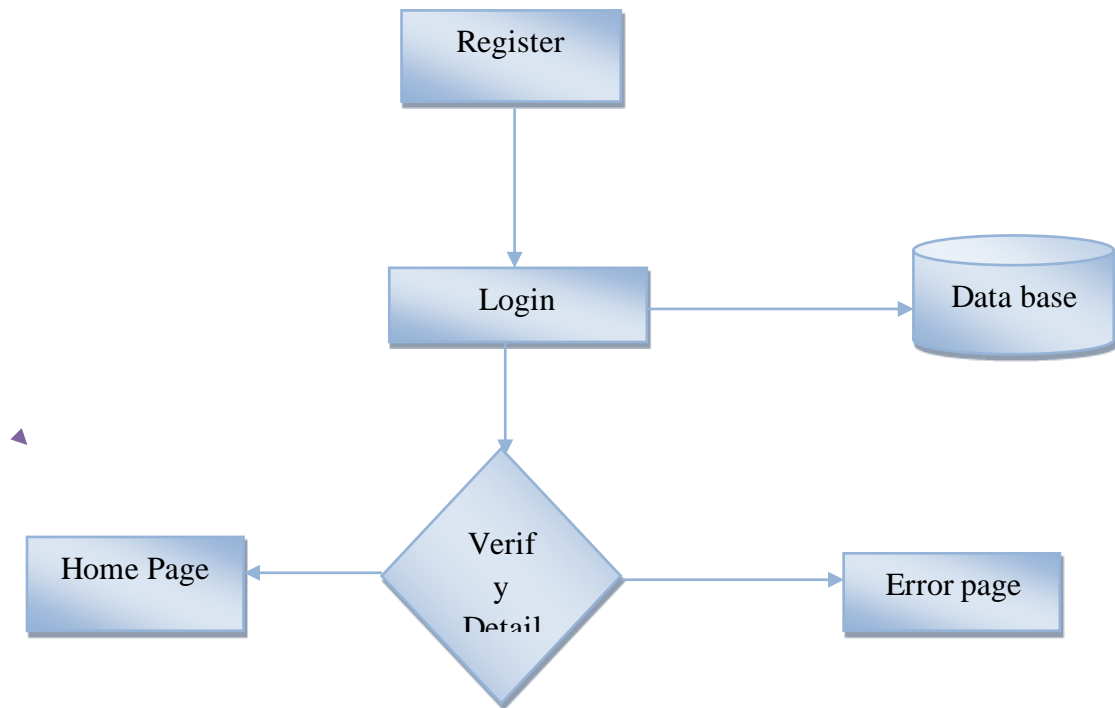


EXPLANATION:

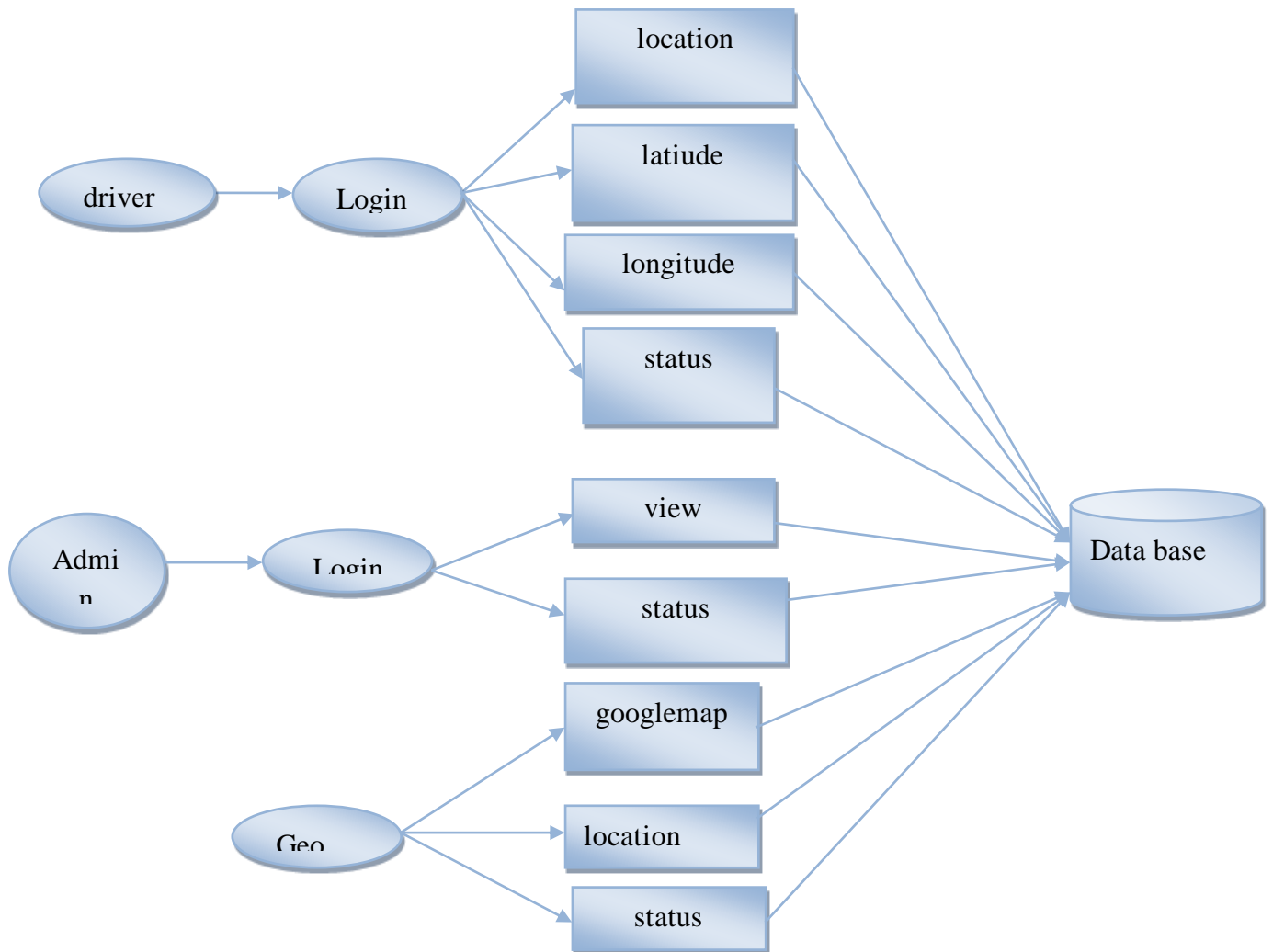
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Data Flow Diagram

Level 0



Level 1

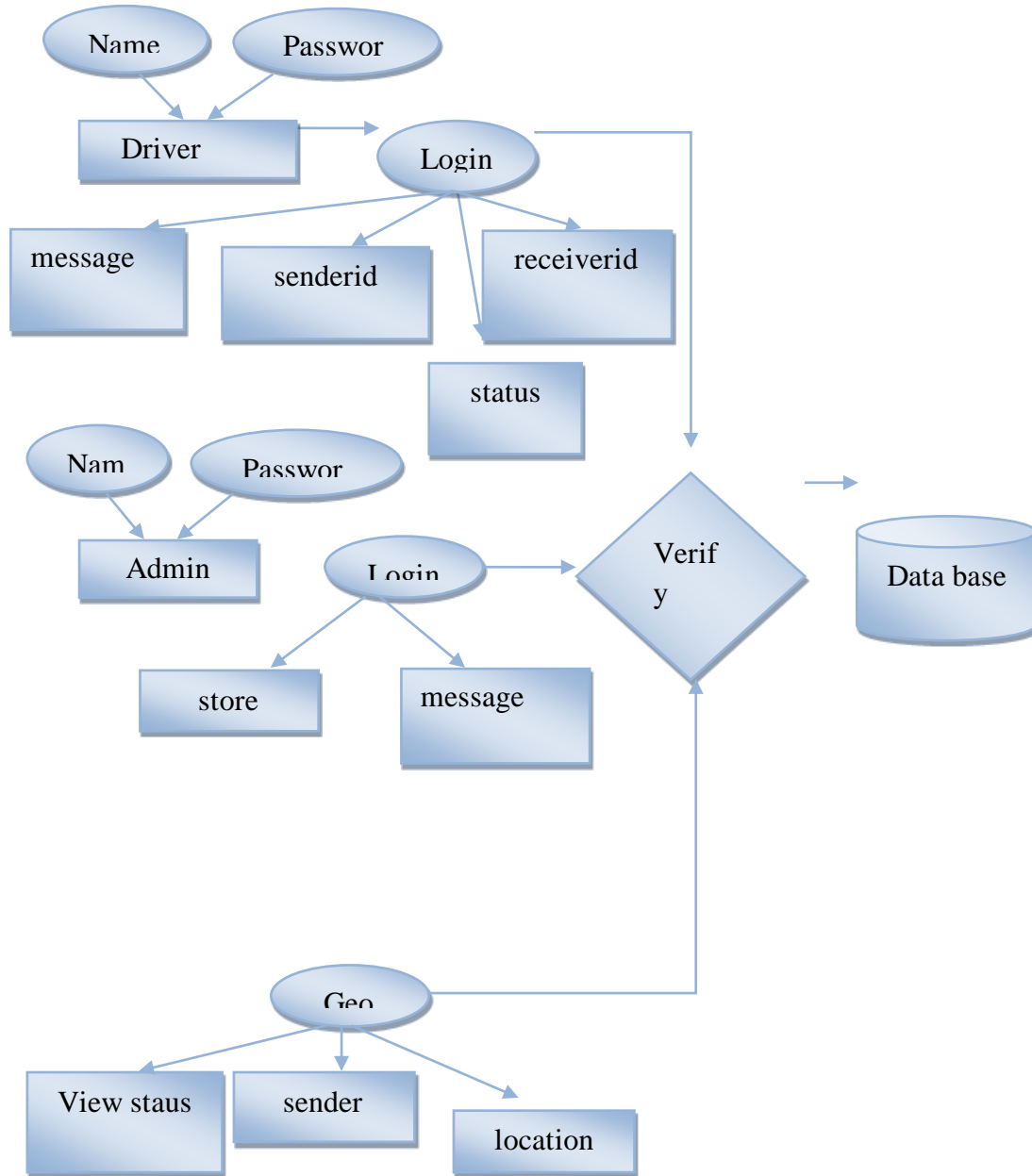


EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

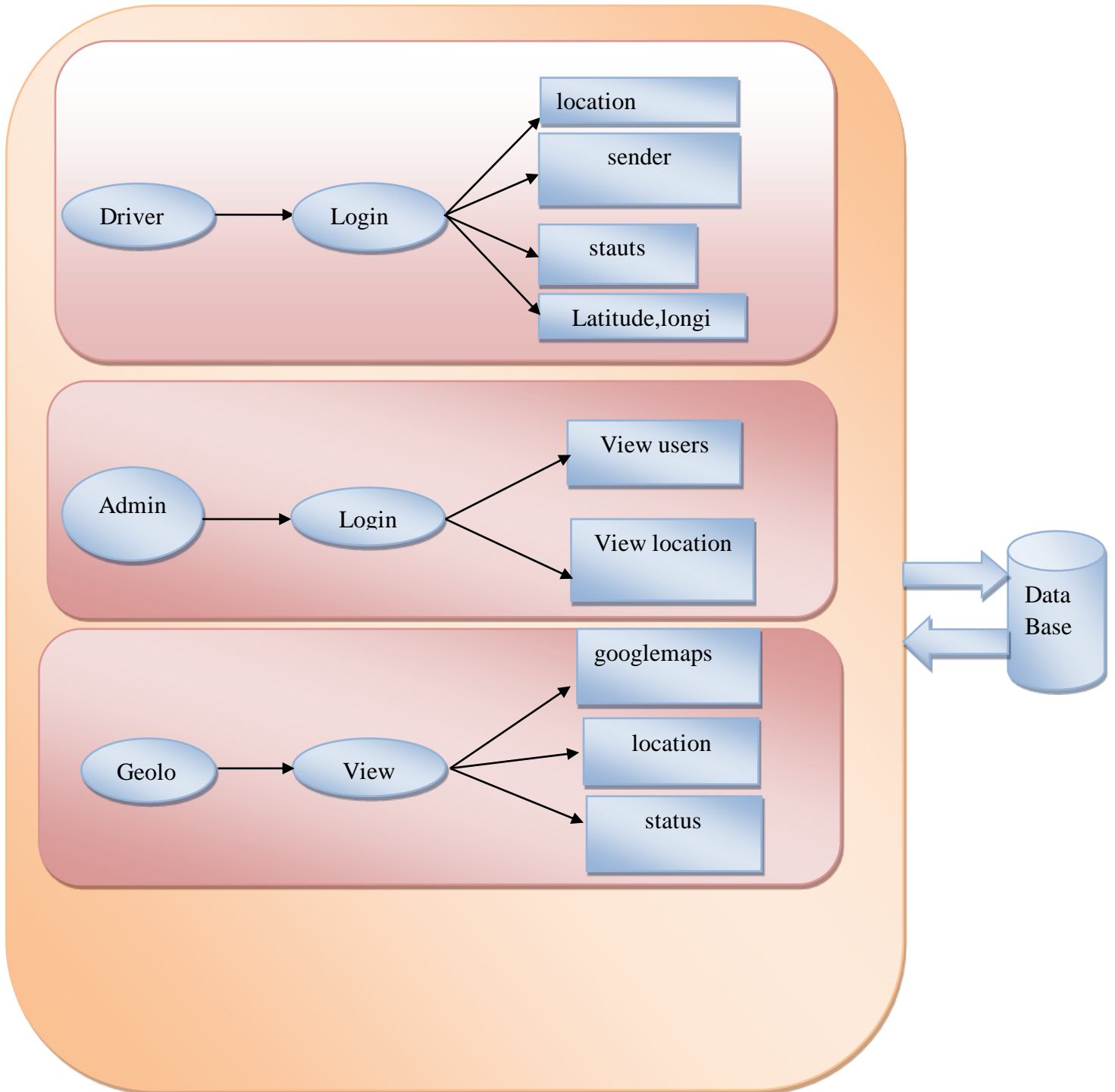
E-R Diagram



EXPLANATION:

Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database.

System Architecture



CHAPTER 5

DEVELOPMENT TOOLS

1.2 What is android?

Android is a software package and linux based operating system for mobile devices such as tablet computers and smart phones. It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used. The goal of android project is to create a successful real-world product that improves the mobile experience for end users. There are many code names of android such as Lollipop, Kitkat, Jelly Bean, Ice cream Sandwich, Froyo, Ecliar, Donut etc which is covered in next page.

1.3 About Open Handset Alliance (OHA)

It's a consortium of 84 companies such as Google, Samsung, AKM, synaptics, KDDI, Garmin, Teleca, Ebay, Intel etc. It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Plateform.

1.4 Features of Android

- 1) It is open-source.
- 2) Anyone can customize the Android Platform.
- 3) There are a lot of mobile applications that can be chosen by the consumer.
- 4) It provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.

It provides support for messaging services(SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Blue Tooth, Wi-Fi etc.), media, handset layout etc.

1.5 Categories of Android applications

- Entertainment
- Tools
- Communication
- Productivity
- Personalization
- Music and Audio
- Social
- Media and Video
- Travel and Local etc.

1.6 History of Android

- 1) Initially, **Andy Rubin** founded Android Incorporation in Palo Alto, California, United States in October, 2003.
- 2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.
- 3) The key employees of Android Incorporation are **Andy Rubin, Rich Miner, Chris White and Nick Sears**.
- 4) Originally intended for camera but shifted to smart phones later because of low market for camera only.
- 5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots.
- 6) In 2007, Google announces the development of android OS. In 2008, HTC launched the first android mobile.

1.7 Android Architecture

1. linux kernel: It is the heart of android architecture that exists at the root of android architecture. Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.

2. native libraries (middleware): On the top of linux kernel, there are **Native libraries** such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc. The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats.

3. Android Runtime: In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

4. Application Framework: On the top of Native libraries and android runtime, there is android framework. Android framework includes **Android API's** such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

5. Applications: On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using linux kernel.

1.8 Android Core Building Blocks

An android component is simply a piece of code that has a well defined life cycle e.g. Activity, Receiver, Service etc.

The core building blocks or fundamental components of android are activities, views, intents, services, content providers, fragments and AndroidManifest.xml.

Activity

An activity is a class that represents a single screen. It is like a Frame in AWT.

View

A view is the UI element such as button, label, text field etc. Anything that you see is a view.

Intent

Intent is used to invoke components. It is mainly used to:

- Start the service
- Launch an activity
- Display a web page
- Display a list of contacts
- Broadcast a message
- Dial a phone call etc.
- For example, you may write the following code to view the webpage.

Example :

```
Intent intent=new Intent(Intent.ACTION_VIEW);  
  
intent.setData(Uri.parse("http://www.gurunanak.com"));  
  
startActivity(intent);
```

Service

Service is a background process that can run for a long time. There are two types of services local and remote. Local service is accessed from within the application whereas remote service is accessed remotely from other applications running on the same device.

Content Provider

Content Providers are used to share data between the applications.

Fragment

Fragments are like parts of activity. An activity can display one or more fragments on the screen at the same time.

AndroidManifest.xml

It contains information about activities, content providers, permissions etc. It is like the web.xml file in Java EE.

Android Virtual Device (AVD)

It is used to test the android application without the need for mobile or tablet etc. It can be created in different configurations to emulate different types of real devices.

APK File

An apk file is created by the framework automatically. If you want to run the android application on the mobile, transfer and install it.

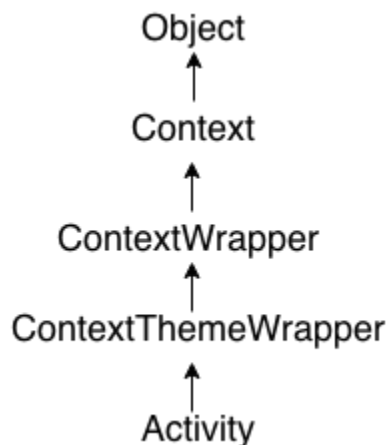
Resources

It contains resource files including activity_main, strings, styles etc.

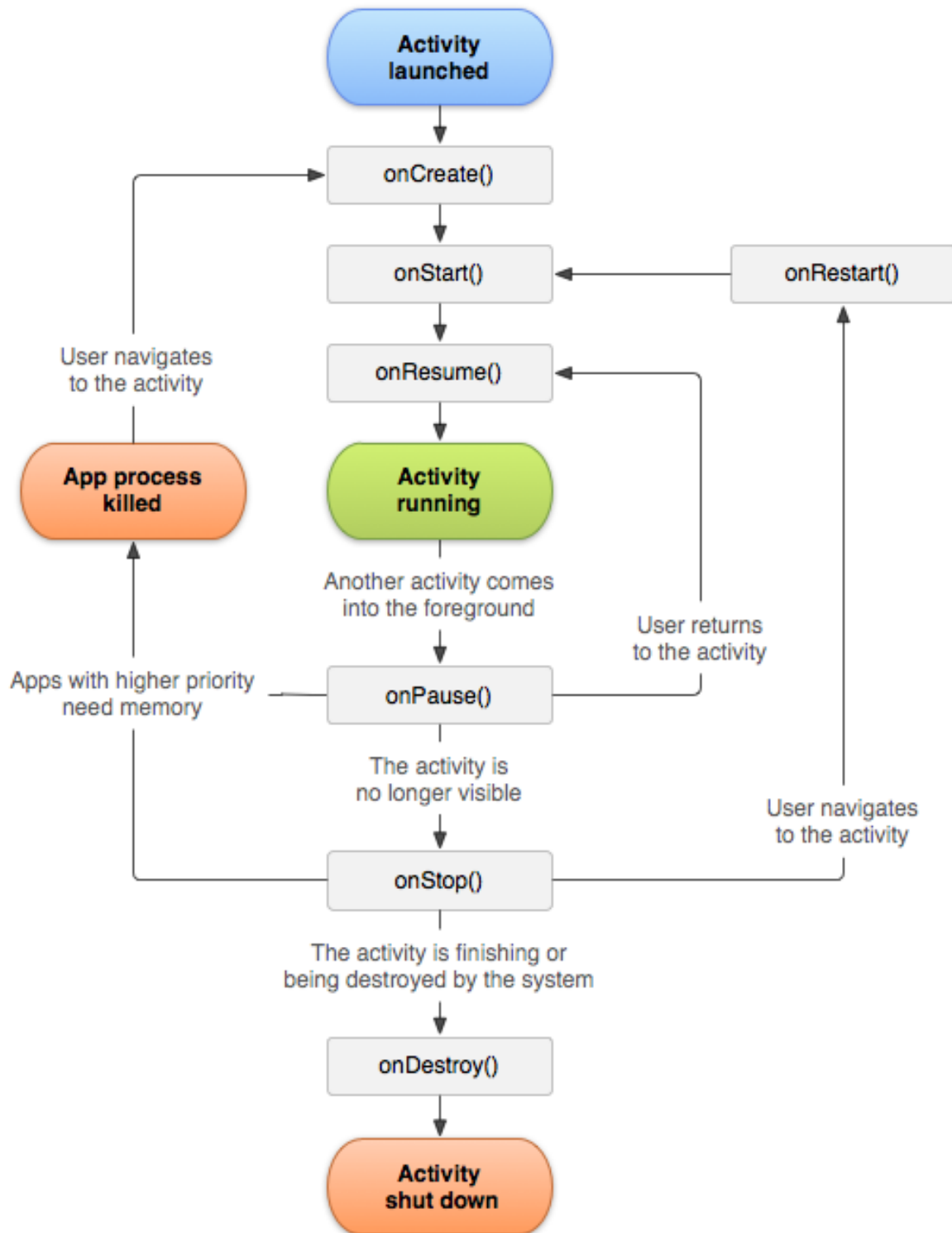
Manifest file

It contains information about package including components such as activities, services, content providers etc.

1.9 Android Activity Lifecycle



Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class. An activity is the single screen in android. It is like window or frame of Java. By the help of activity, you can place all your UI components or widgets in a single screen. The 7 lifecycle method of Activity describes how activity will behave at different states.



CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

Coding:

MainActivity.java

```
package com.us.onlineop;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button bthospital, btuser, btreghospital;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        bthospital=findViewById(R.id.btreghospital);

        btreghospital=findViewById(R.id.bthospital);

        btuser=findViewById(R.id.btuser);

        bthospital.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {
```

```

        startActivity(new Intent(getApplicationContext(),Hospital.class));
    }
});

btuser.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        startActivity(new Intent(getApplicationContext(),User.class));

    }

});

btreghospital.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        startActivity(new Intent(getApplicationContext(),RegisterHospital.class));

    }

});

}

```

AddOp.java

```

package com.us.onlineop;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.support.annotation.NonNull;

```

```

import android.support.annotation.Nullable;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.widget.AdapterView;

import android.widget.EditText;

import android.widget.ListView;

import android.widget.Spinner;

import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.Query;

import com.google.firebase.database.ValueEventListener;

import com.us.onlineop.Models.DoctorModel;

import com.us.onlineop.Models.Opusers;

import com.us.onlineop.Models.Store;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

public class AddOp extends AppCompatActivity {

    EditText etname,etsymptoms,etgender,etage,etdocid;

```

```

Spinner gender;

DatabaseReference mdatasref;

String docid;

String userid;

List<Opusers> opusersList;

ListView lvlist;

HashMap map;

List<DoctorModel> doctorModelList;

ArrayList arrayList;

Opusers op;

@Override

protected void onCreate( @Nullable Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_add_op);

    etname=findViewById(R.id.etname);

    etdocid=findViewById(R.id.etdocid);

    lvlist=findViewById(R.id.lvlist);

    opusersList=new ArrayList<>();

    map= Store.getUserDetails(this);

    etsymptoms=findViewById(R.id.etsymptoms);

    arrayList=new ArrayList();

    doctorModelList=new ArrayList<>();

    mdatasref=FirebaseDatabase.getInstance().getReference("opdoctors");

```

```

Query query=mdatabaseref.orderByChild("hospid").equalTo(map.get("docid").toString());
query.addValueEventListener(new ValueEventListener() {

    @Override

    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

        for(DataSnapshot dataSnapshot1:dataSnapshot.getChildren()){

            DoctorModel doctorModel=dataSnapshot1.getValue(DoctorModel.class);

            doctorModelList.add(doctorModel);

        }

        DoctorDisplay arrayAdapter=new DoctorDisplay(AddOp.this,doctorModelList);

        lvlist.setAdapter(arrayAdapter);

    }

    @Override

    public void onCancelled(@NonNull DatabaseError databaseError) {

    }

});

mdatabaseref= FirebaseDatabase.getInstance().getReference("opusers");

Bundle bundle=getIntent().getExtras();

// docid=bundle.getString("docid");

etage=findViewById(R.id.age);

gender=findViewById(R.id.spgender);

String s[]={ "male", "female" };

ArrayAdapter arrayAdapter=new ArrayAdapter(this,android.R.layout.simple_list_item_1,s);

```



```

gender.setAdapter(arrayAdapter);

findViewById(R.id.bsubmit).setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        final String name=etname.getText().toString();

        final String symptoms=etsymptoms.getText().toString();

        final String gender1=gender.getSelectedItem().toString();

        final String age=etage.getText().toString();

        final String docid=etdocid.getText().toString();

        final String[] userid = new String[1];

        final int[] i = new int[1];

        if(validate(name,symptoms,gender1,age,docid)){

            // Ousers opusers=new Ousers();

            final Ousers opusers=new Ousers();

            mdataserefer.addListnerForSingleValueEvent(new ValueEventListener() {

                @Override

                public void onDataChange( @NonNull DataSnapshot dataSnapshot) {

                    for(DataSnapshot dataSnapshot1:dataSnapshot.getChildren()){

                        userid[0]=dataSnapshot1.getKey();

                        Ousers opusers1=dataSnapshot1.getValue(Ousers.class);

                        opusersList.add(opusers1);

                        Log.e("opusers",String.valueOf(opusersList.size()));

```

```

        i[0]=opusersList.size()+1;
    }
    if(i[0]==0){
        i[0]=1;
    }

    opusers.setOpid(String.valueOf(i[0]));

    Log.e("i value is",String.valueOf(i[0]));

opusers.setName(name);opusers.setSymptoms(symptoms);opusers.setGender(gender1);

    opusers.setAge(age);opusers.setDocid(docid);opusers.setStatus("Waiting");

    opusers.setUserkey(userid[0]);

    userid[0]=mdatabaseref.push().getKey();

    // opusers.setUserkey(userid);

    opusers.setUserkey(userid[0]);

    mdatabaseref.child(userid[0]).setValue(opusers);

    addOpListener();

    Toast.makeText(getApplicationContext(),"addedd
op",Toast.LENGTH_SHORT).show();

    startActivity(new Intent(getApplicationContext(),Receptionist.class));

    finish();

}

@Override

    public void onCancelled(@NonNull DatabaseError databaseError) {

    Toast.makeText(getApplicationContext(),databaseError.toString(),Toast.LENGTH_SHORT).show();
}

```

```

        }

    });

    }else{

    }

    }

});

}

private void addOpListener() {

    final ProgressDialog progressDialog = new ProgressDialog(AddOp.this);

    progressDialog.setTitle("connecting...");

    progressDialog.show();

    final List<Opusers> opusersList = new ArrayList<>();

    mdataserefer.child(userid).addListenerForSingleValueEvent(new ValueEventListener() {

        @Override

        public void onDataChange(DataSnapshot dataSnapshot) {

            Opusers user = dataSnapshot.getValue(Opusers.class);

            // Check for null

            if (user == null) {

                Log.e("data", "User data is null!");

                ToastHelper.toastMsg(getApplicationContext(), "something went wrong");

                progressDialog.hide();

                return;

```

```

}else {

    Query query= mdataseref.orderByChild("docid").equalTo(user.getDocid());

    query.addListenerForSingleValueEvent(new ValueEventListener() {

        @Override

        public void onDataChange( @NonNull DataSnapshot dataSnapshot) {

            for(DataSnapshot dataSnapshot1:dataSnapshot.getChildren()){

                op=datasnapshot1.getValue(Opusers.class);

                opusersList.add(op);

            }

            int i=opusersList.size();

            op.setOpid(String.valueOf(i));

            mdataseref.child(userid).setValue(op);

            progressDialog.hide();

        }

        @Override

        public void onCancelled(@NonNull DatabaseError databaseError) {

            ToastHelper.toastMsg(getApplicationContext(),databaseError.toString());

        }

    });

    // ToastHelper.toastMsg(getApplicationContext(), "Registration success");

    // startActivity(new Intent(getApplicationContext(), UserSuccess.class));

```

```

        progressDialog.hide();

    }

}

@Override

public void onCancelled(DatabaseError databaseError) {

    Log.e("data","user data is cancelled");

    ToastHelper.toastMsg(getApplicationContext(),"something went wrong");

    progressDialog.hide();

}

});

}

public boolean validate(String name, String symptoms, String gender1, String age, String docid)
{

    if(name.isEmpty() || name==null){

        Toast.makeText(getApplicationContext(),"name cannot be
empty",Toast.LENGTH_SHORT).show();

        return false;

    }else if(symptoms.isEmpty() || symptoms==null){

        Toast.makeText(getApplicationContext(),"symptoms cannot be
empty",Toast.LENGTH_SHORT).show();

        return false;

    }else if(gender1.isEmpty() || gender1==null){

        Toast.makeText(getApplicationContext(),"select
gender",Toast.LENGTH_SHORT).show();

        return false;
    }
}

```

```

        }else if(age.isEmpty() || age==null){

            Toast.makeText(getApplicationContext(),"age
empty",Toast.LENGTH_SHORT).show();

            return false;

        }else if(docid.isEmpty() || docid==null){

            Toast.makeText(getApplicationContext(),"doc
empty",Toast.LENGTH_SHORT).show();

            return false;

        }else{

            return true;

        }

    }

}

```

DocotorsDisplay.java

```

package com.us.onlineop;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.LinearLayout;

import android.widget.TextView;

```

```

import com.us.onlineop.Models.DoctorModel;

import java.util.List;

class DoctorDisplay extends BaseAdapter {

    Context context;

    List<DoctorModel> doctorModelList;

    public DoctorDisplay(AddOp addOp, List<DoctorModel> doctorModelList) {

        this.context=addOp;

        this.doctorModelList=doctorModelList;

    }

    @Override

    public int getCount() {

        return doctorModelList.size();

    }

    @Override

    public Object getItem(int position) {

        return position;

    }

    @Override

    public long getItemId(int position) {

        return position;

    }

    @Override

    public View getView(int position, View convertView, ViewGroup parent) {

```

```

convertView= LayoutInflater.from(context).inflate(R.layout.activity_doc_row,null);

TextView tvdocid=convertView.findViewById(R.id.docid);

TextView tvdocname=convertView.findViewById(R.id.docname);

TextView tvdocspec=convertView.findViewById(R.id.docspec);

DoctorModel doctorModel=doctorModelList.get(position);

tvdocid.setText("Doctor id:\t\t"+doctorModel.getDocid());

tvdocname.setText("Doctor name:\t\t"+doctorModel.getDocname());

tvdocspec.setText("Doctor spec:\t\t"+doctorModel.getDocspec());

return convertView;

}

}

```

Activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:orientation="vertical"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:gravity="center"

    tools:context=".MainActivity">

    <Button

        android:id="@+id/bthospital"

```



```
    android:text="Register Your Hospital"

    android:layout_margin="25dp"

    android:background="@drawable/borderbutton_red"

    android:layout_width="match_parent"

    android:layout_height="wrap_content" />
```

```
<Button
```

```
    android:id="@+id/btreghospital"

    android:text="HOSPITAL"

    android:layout_margin="25dp"

    android:background="@drawable/borderbutton_red"

    android:layout_width="match_parent"

    android:layout_height="wrap_content" />
```

```
<Button
```

```
    android:id="@+id/btuser"

    android:text="User"

    android:background="@drawable/borderbutton_red"

    android:layout_margin="25dp"

    android:layout_width="match_parent"

    android:layout_height="wrap_content" />
```

```
</LinearLayout>
```

```
Addop.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<LinearLayout
```

```
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
<EditText
```

```
    android:id="@+id/etname"
    android:layout_margin="25dp"
    android:hint="Enter your name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```
<EditText
```

```
    android:id="@+id/etsymptoms"
    android:layout_margin="25dp"
    android:hint="Enter symptoms"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```
<Spinner
```

```
    android:layout_margin="25dp"
    android:id="@+id/spgender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"></Spinner>
```

```
<EditText  
    android:id="@+id/etdocid"  
    android:layout_margin="25dp"  
    android:hint="Enter Doctorid"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

```
<EditText  
    android:id="@+id/age"  
    android:layout_margin="25dp"  
    android:hint="Enter age"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

```
<Button  
    android:id="@+id/btsubmit"  
    android:layout_gravity="center"  
    android:text="get op"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

```
<ListView  
    android:id="@+id/lvlist"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"></ListView>
```

```
</LinearLayout>
```

</ScrollView>

Activity_doctors_list

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical" android:layout_width="match_parent"

android:layout_height="match_parent">

<ListView

android:id="@+id/lvlist"

android:layout_margin="15dp"

android:layout_width="match_parent"

android:layout_height="wrap_content"></ListView>

</LinearLayout>

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.us.onlineop">

<application

android:allowBackup="true"

android:icon="@mipmap/ic_launcher"

android:label="@string/app_name"

android:roundIcon="@mipmap/ic_launcher_round"

android:supportRtl="true"

android:theme="@style/AppTheme">

```

<activity android:name=".MainActivity">

    <intent-filter>

        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />

    </intent-filter>

</activity>

<activity android:name=".AddRecep"></activity>

<activity android:name=".AddDoctor"></activity>

<activity android:name=".AdminSuccess"></activity>

<activity android:name=".Hospital"></activity>

<activity android:name=".Login"></activity>

<activity android:name=".User"></activity>

<activity android:name=".ViewDoctors"></activity>

<activity android:name=".ViewRecepe"></activity>

<activity android:name=".Register"></activity>

<activity android:name=".AddOp"></activity>

<activity android:name=".Receptionist"></activity>

<activity android:name=".Get_Op_Doctor"></activity>

<activity android:name=".DoctorSuccess"></activity>

<activity android:name=".Availability"></activity>

<activity android:name=".RegisterHospital"></activity>

<activity android:name=".UserRegister"></activity>

<activity android:name=".UserSuccess"></activity>

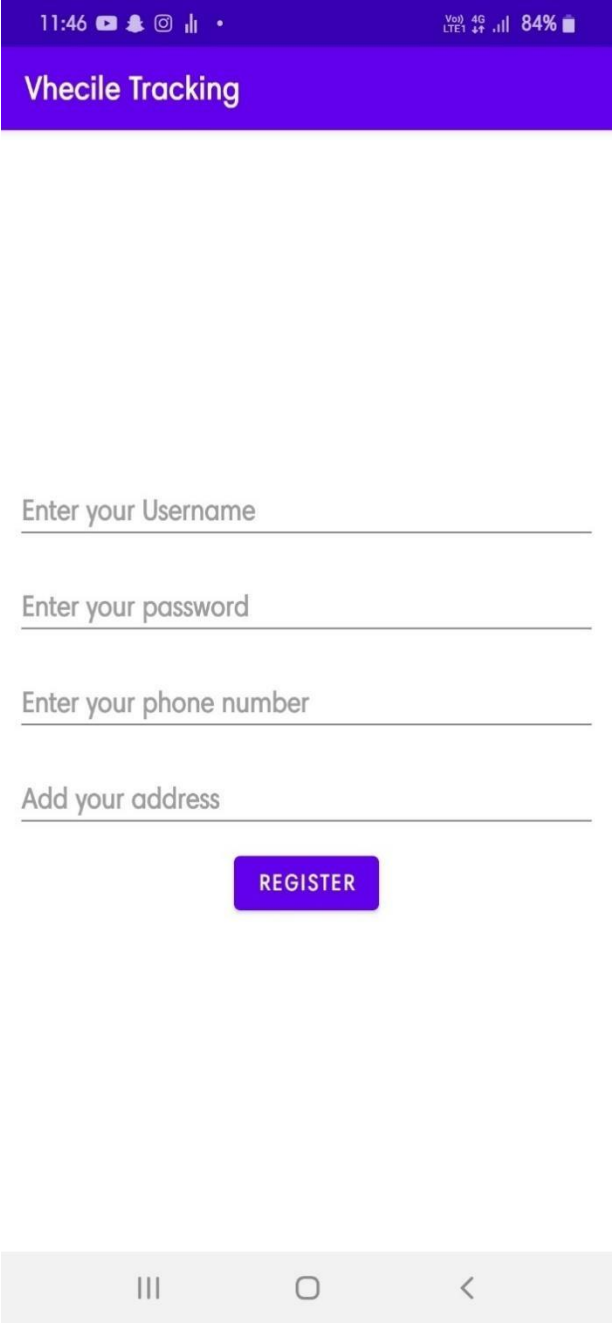
```

```
<activity android:name=".UserSearch"></activity>
<activity android:name=".Hospitalslist"></activity>
<activity android:name=".DoctorsList"></activity>
<activity android:name=".GetOpForm"></activity>
<activity android:name=".DashBoard"></activity>
</application>
</manifest>
```

CHAPTER 7

SNAPSHOTS

7.1 SNAPSHOTS



The image shows a mobile application interface for "Vhecile Tracking". At the top, there is a status bar with the time 11:46, social media icons, and battery level 84%. Below the status bar is a purple header with the text "Vhecile Tracking". The main content area is white and contains four text input fields with labels: "Enter your Username", "Enter your password", "Enter your phone number", and "Add your address". Below these fields is a purple button labeled "REGISTER". At the bottom of the screen is a grey navigation bar with three icons: a hamburger menu, a home icon, and a back arrow.

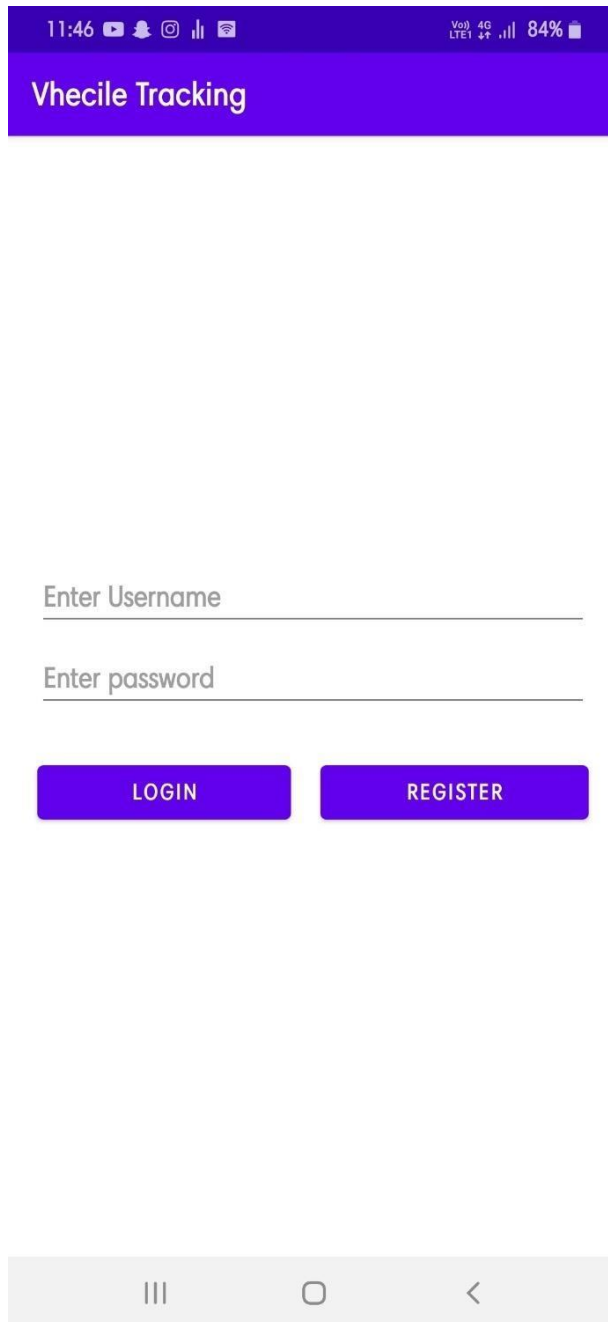
Fig7.1.1: User Registration

The user has to enter their desired username ID, password, phone number and their address to register an account for the application.



Fig 7.1.2: Admin login UI

The admin will login to their account. After they are logged in, they will be able to get the location of driver and the customer, so it will be easy to allocate the driver to the customer.



The image shows a mobile application interface for "Vhecile Tracking". At the top, there is a status bar with the time 11:46, social media icons, and battery status at 84%. Below the status bar is a blue header with the text "Vhecile Tracking". The main area is white and contains two input fields: "Enter Username" and "Enter password". Below these fields are two blue buttons: "LOGIN" and "REGISTER". At the bottom, there is a grey navigation bar with three icons: a hamburger menu, a home icon, and a back arrow.

Fig 7.1.3: User login UI

After the user has registered, they have created an account. Now, they have to enter their username ID and password to login to their created account.

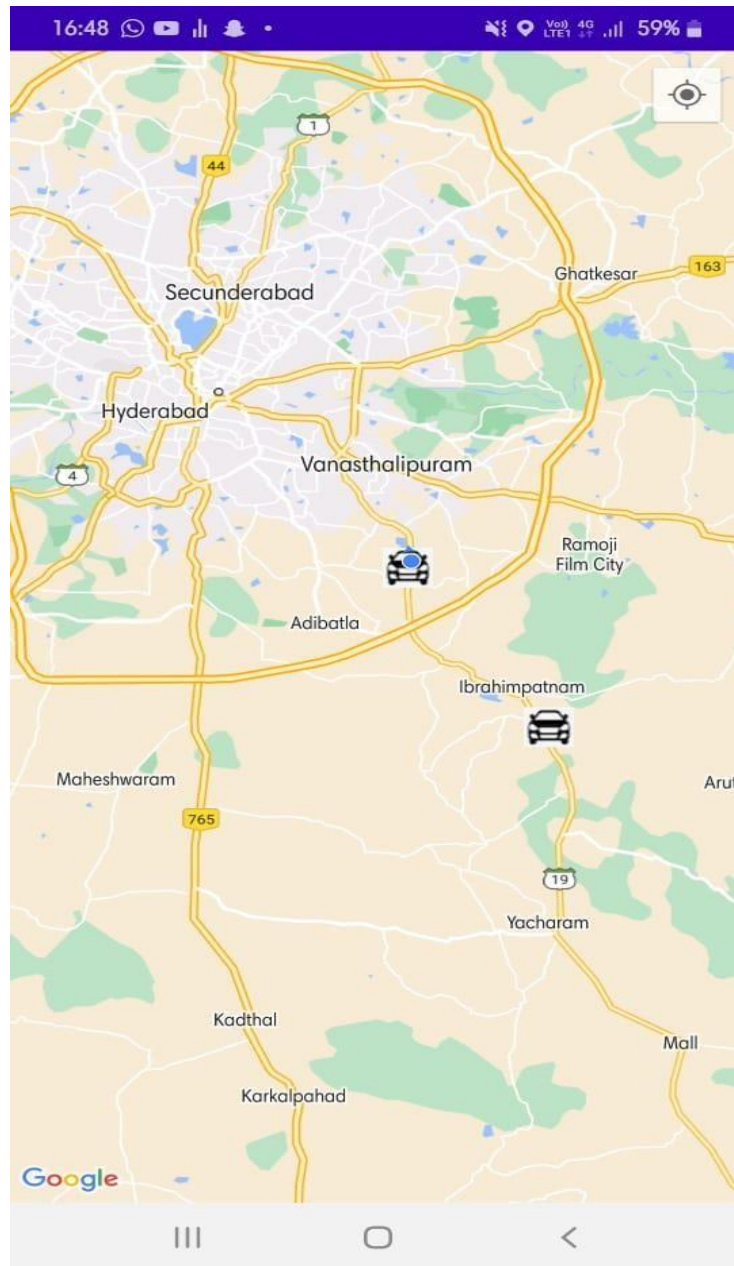


Fig 7.1.4: User Location on Google map

The admin is given access by the user about their location. The admin is able to track and reach the user's location.

CHAPTER 8

SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3 Types of Tests

8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

8.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process.

8.2.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 9

FUTURE ENHANCEMENT

As the dedicated GPS devices are costly we have chosen android mobile phone as GPS device. Because all the android mobile phones have this built-in feature. GPS device will find out the current location from satellite. Depending on certain condition we will find the location again. Might after certain distance of location change we will count the location again. We will plot this location to map. At the same time we will connect with an external web server to send this information there. The web server will store the visiting path as a summation of some co-coordinating points. Actually, the web server is a restricted area. So we will have to use username and password to login that area. We can see the visiting path from the desktop or any other portable device. Like other systems it is also not free from drawbacks. It will need the GPS service on from the time we want to track. It will

mostly use the battery power of the mobile phone. For this reason our mobile phone can be automatically switched off for lack of power. We need to depend on Internet connection to

store the information to the web server. If there is some problem with Internet connection or lack of necessary bandwidth, we may not be able to send the data correctly

CHAPTER 10

CONCLUSION & REFERENCE

10.1 CONCLUSION

This Mobile Tracking System has been designed and developed and works properly. It is very efficient, user can easily use this application. Any people can track any mobile location any time using this application. The application is free of cost and does not require any additional device. We have tested in different Android phones and different browsers it works smoothly. We think users will be benefited by using this application.

10.2 REFERENCES

- [1] <http://www8.garmin.com/aboutGPS/>
- [2] http://www.wild-heerbrugg.com/gps_timeline.htm
- [3] <http://inventors.about.com/od/gstartinventions/a/gps.htm>
- [4] <http://spectrum.library.concordia.ca/>
- [5] <http://www.collectionscanada.gc.ca/>
- [6] <https://play.google.com/store/apps/details?id=com.androidtips.android.tips&hl=en>
- [7] <http://en.wikipedia.org/wiki/Multilateration>
- [8] <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>
- [9] <http://developer.android.com/tools/devices/index.html>
- [10] <http://jumpstartcto.com/glossary/lamp>