

<u>INDEX</u>		
<u>No.</u>	<u>Title</u>	<u>Pg. No</u>
<u>1.</u>	<u>CHAPTER 1 : INTRODUCTION OF PROJECT</u>	<u>3</u>
	<u>1.1 Purpose and Scope</u>	<u>4</u>
	<u>1.2 Document Conventions</u>	<u>5</u>
	<u>1.3 Intended Audience</u>	<u>5</u>
	<u>1.4 Assumption and Dependencies</u>	<u>6</u>
<u>2.</u>	<u>CHAPTER 2: PROJECT PLANNING</u>	<u>7</u>
	<u>2.1 Scope of the system</u>	<u>8</u>
	<u>2.2 Project Team</u>	<u>8</u>
	<u>2.3 Project Schedule</u>	<u>9</u>
	<u>2.4 Frontend & Backend</u>	<u>10</u>
	<u>2.5 Software Process Model</u>	<u>10</u>
<u>3.</u>	<u>CHAPTER 3:SYSTEM ANALYSIS</u>	<u>13</u>
	<u>3.1 Literature Survey</u>	<u>13</u>
	<u>3.2 Fact Finding</u>	<u>23</u>
	<u>3.3 Feasibility study</u>	<u>24</u>
	<u>3.4 System Requirement Engineering</u>	<u>24</u>
<u>4.</u>	<u>CHAPTER 4 :SYSTEM DESIGN</u>	<u>30</u>
	<u>4.1 Use -Case Diagram</u>	<u>31</u>

	<u>4.2 Sequence Diagram</u>	<u>33</u>
	<u>4.3 E-R Diagram</u>	<u>34</u>
	<u>4.4 Activity Diagram</u>	<u>35</u>
	<u>4.5 User Design</u>	<u>36</u>
<u>5.</u>	<u>CHAPTER 5: USER MANUAL</u>	<u>46</u>
	<u>5.1 USER MANUAL</u>	<u>47</u>
<u>6.</u>	<u>CHAPTER 6: TOOLS AND TECHNOLOGIES</u>	<u>55</u>
	<u>6.1 TOOLS AND TECHNOLOGY</u>	<u>56</u>
<u>7.</u>	<u>CHAPTER7: TESTING</u>	<u>57</u>
	<u>7.1 Testing</u>	<u>58</u>
	<u>7.2 Testing plans</u>	<u>58</u>
	<u>7.3 Testing strategy</u>	<u>58</u>
	<u>7.4 Testing methods</u>	<u>59</u>
	<u>7.5 Test Cases</u>	<u>62</u>

- **Introduction**

CHAPTER-1-INTRODUCTION OF PROJECT

1.1 : Purpose and scope

- The world today we live in is dominated by the machines and technology and one of them is 4wheelers that we use in today's world on the daily basis. Most of the people uses cars for commuting from one place to another and they often triggers with some problems in their cars, they can search the garages and find better garages in the area and see reviews given by others based on that they can go to better garages and get their problem solved.
- The major problem that the car holder faces is that they don't have enough knowledge about garages especially for the people who are not the native livers of the city and they have a great advantage using this application and they can search and navigate the location of garage through google maps.
- The SCS is developed to solve the problem of the people who uses the car as a source of a medium to commute from one place to another place. The SCS aims to reduce the time of finding good garages in the city and around the city.
- Users have to download this app. Then, they have to register themselves on the application by providing some registration information. Both the mechanics and customers have to register themselves and then their account will get verified by email verification, if they get verified then they will be able to access the app features and do whatever they want.
- Mechanics will provide services to the users according to their problem and their choices. So the user will find user friendly environment with mechanics.
- Thus, with the help of SCS the car services will become rapid and easy-going for the users. With the help of SCS, more and more number of people will receive services according to their taste.

- The scope of the application is for garages and service stations by merging with them and provides service to the user with largely reduced efforts.

1.2:Abstract

- Smart car service is an android application for helping any individual who lives in city and does not have a knowledge of good garages plus also for mechanics too. Mechanics can create their own profile and customers can see it according to their area plus mechanics can also see customer list of their area. It helps customers too see and view garage list according to their area and they can see the garages available in those areas, plus they can also give feedback, find garage's location through navigation, call mechanic and see reviews given by other customers.

1.3:Document Convention

- We have followed the simple convention of a technical document while writing this report. The header represents various categories of the content mentioned in the document and is of the font size 24, the sub header represents any keyword and the particular section contains the detailed description in the context of the application. The font size of the sub heading and the body is 22 and 20 respectively.

1.4:Intended Audience

- The intended users of the SCS are the business organization people, company people, job based people, doctors, general public, etc. Apart from this the garage and service station holders are also the intended users who merged with the SCS. The garage and service station people can check their regular users, their feedback and comments by users, etc. So that it can help them to improve their efforts for better quality of service.

1.5: Assumption and Dependencies

- Here it is assumed that the users have an android enabled device. It is also assumed that the version of android should be above Ice-cream sandwich.
- It is also assumed that user and merged garage and service stations have their unique id allocated to them by us.
- Apart from that the user and garage and service station people must have android enabled device.
- And all those android enabled devices must have sufficient storage space to store the SCS.
- The network connectivity is assumed to allow feedback from the user and other network based activities.

II: SYSTEM **ANALYSIS AND** **DESIGN**

CHAPTER-2-PROJECT PLANNING

2.1:Scope of the system

- The Smart Car Service has its scope to the daily use of the car users. The Smart Car Service is intended for the users who use the car for their daily travel by the above mentioned purposes. And mostly those users who have less time to go to the mechanic and tell the problem & solve it at that time. Emergency services also are provided.
- This project is to develop an application to notify the car services. The following tips will show its scope :

1. It must give users convenient and effective ways to deal with one another.
2. Friendly interfaces are also necessary in this project.
3. Only server can visit entire database, and user can visit only his/her database.

2.2:Project Team

<u>Name</u>	<u>Department</u>	<u>Enrollment Number</u>
Darshit Rana	Information Technology	140410116012
Meet Patel	Information Technology	140410116061
Vishal Chauhan	Information Technology	150413116004

2.3:Project Schedule Chart

Phase 1: SEARCHING PROBLEM DEFINITION AND STUDY OF CURRENT SCENARIO

ID	Task Name	Start	Finish	Jun 2017																								
				6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	SEARCHING PROBLEM DEFINITION	06-Jun-17	08-Jun-17																									
2	STUDY OF CURRENT SCENARIO	12-Jun-17	29-Jun-17																									

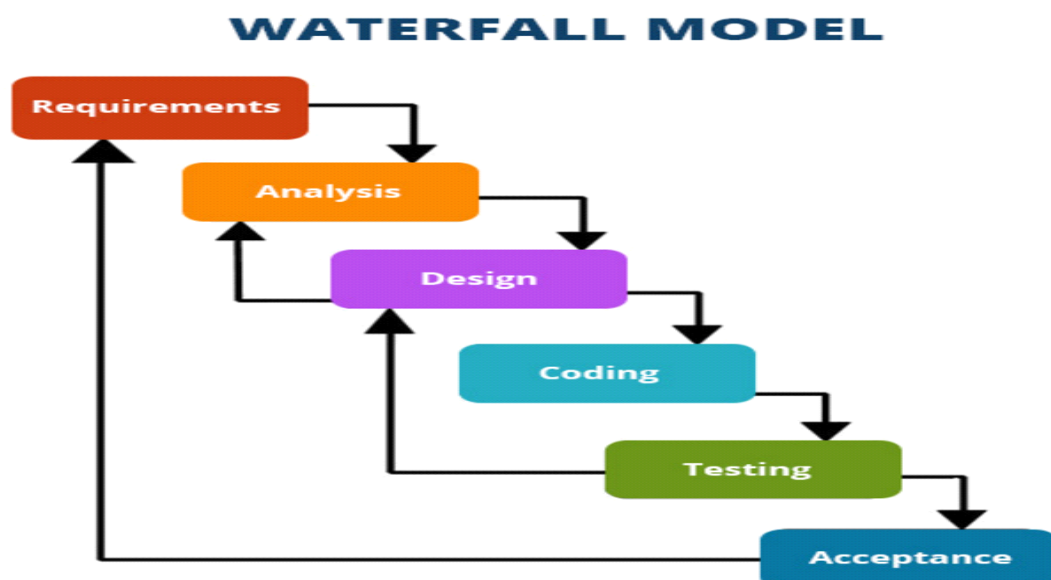
2.4:Frontend & Backend

Front End	Android
Back End	c#.net, My SQL server 2012
Diagram designing tools	Microsoft office Visio 2007, 2013
OS - Platform	Windows
Tools	Android Studio, Visual studio, sql studio manager

2.5:Software Process Model

- A software process model is a simplified description of a software process which is presented from a particular perspective. Models, by their very nature, are simplifications so a software process model is an abstraction of the actual process which is being described. Process models may include activities which are part of the software process, software products and the roles of people involved in software engineering. A process model is a development strategy that is used to achieve a goal that satisfies the requirements abiding by the constraints.

This software is adjusted into Waterfall Model with its parameters.



2.5.1:Parameters of the Waterfall model:

The phases in the Waterfall model are

- 1) **Requirements:** This defines the requirements of the user.
- 2) **Analysis:** This phase include the different analyzing processes. In analysis phase various aspects of the system check during the process.
- 3) **Design:** This phase includes all designing aspects. Designing include user interface of the system. As we know the user interface for any system is very important so it is consider being important phase of the system.
- 4) **Coding:** After designing phase you should try to get the implementation of the system. The coding part of the system is covered in the development phase. So, the specific knowledge of programming language is implemented in this part.
- 5) **Testing:** Test new components and all the interfaces.
- 6) **Acceptance:** This phase moves to accept the system.

2.5.2:Use of waterfall model :

1. This model is used only when the requirements are very well known, clear and fixed.
2. Product definition is stable.
3. Technology is understood.
4. There are no ambiguous requirements.
5. Ample resources with required expertise are available freely.

2.5.3: Why we use this model:

1. You don't realize any value until the end of the project (when you deploy) (See: Self-Funding Projects, a Benefit of Agile Software Development).
2. You leave the testing until the end, which means you're leaving issue discovery until late in the day.
3. You don't seek approval from the users until the day – their requirements might have changed.
4. You're heavily reliant upon a plan, which you will often follow to the detriment of the end result.
5. You're heavily reliant upon a project manager driving the way – the power of one.

CHAPTER-3-SYSTEM ANALYSIS

3.1:LITERATURE SURVEY

- In order to achieve success in project implementation, the first step is to research and find information already available. During research, we found many articles related to our topic. Main research questions are:
 - What is Android, and the evolution of Android?
 - What are the features of the Android technology?
 - Android architecture and working of the Android operating system?
 - Application development in Android system?
 - Security issues in Android system?
 - Comparisons with other mobile operating systems?
 - Why the Android is becoming the popular in market?

- **Introduction**

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android is a software platform and operating system for mobile devices based on the Linux operating system and developed by Google and the Open Handset Alliance. It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries, but does not support programs developed in native code. The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008, most of the Android platform will be made available under the Apache free-software and open-source license.

- **The birth of the Android**
- **Google Acquires Android Inc.**

<http://en.wikipedia.org/wiki/Image:Android-logo.svg>In July 2005, Google acquired Android Inc., a small start-up company based in Palo Alto, CA. Android's co-founders who went to work at Google included Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc), Nick Sears (once VP at T-Mobile), and Chris White (one of the first engineers at WebTV). At the time, little was known about the functions of Android Inc. other than they made software for mobile phones. At Google, the team, led by Rubin, developed a Linux-based mobile device OS which they marketed to handset makers and carriers on the premise of providing a flexible, upgradeable system. It was reported that Google had already lined up a series of hardware component and software partners and signalled to carriers that it was open to various degrees of cooperation on their part.

- **Open Handset Alliance Founded**

On 5 November 2007, the Open Handset Alliance, a consortium of several companies which include Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Sprint Nextel and NVIDIA, was unveiled with the goal to develop open standards for mobile devices. Along with the formation of the Open Handset Alliance, the OHA also unveiled their first product, Android, an open source mobile device platform based on the Linux operating system.

Exhibit 1.

Partial Listing of Open Handset Alliance Participants

Operators	OEMs	ISVs	Core Technology Vendors
<ul style="list-style-type: none"> • China Mobile • KDDI • NTT DoCoMo • T-Mobile USA • T-Mobile Germany • Telefonica (including O2) • Telecom Italia • Sprint 	<ul style="list-style-type: none"> • LG • Motorola • HTC • Samsung 	<ul style="list-style-type: none"> • Nuance • Packet Video • Sonic • eBay • The Astonishing Tribe 	<ul style="list-style-type: none"> • Wind River • QUALCOMM • Broadcom • Intel • Texas Instruments

Source: Yankee Group, 2007

Fig 3.1.1

• Hardware

Google has unveiled at least three prototypes for Android, at the Mobile World Congress on February 12, 2008. One prototype at the ARM booth displayed several basic Google applications. A 'd-pad' control zooming of items in the dock with a relatively quick response.

A prototype at the Google IO conference on May 28, 2008 had a 528 MHz Qualcomm processor and a Synaptic capacitive touch screen, and used the UMTS cellular standard. It had 128 MB of RAM and 256 MB of flash, showing that Android's memory requirements are reasonable. The demo was carried out using a 3.6 Mbit/s HSDPA connection.

- **Features**
- **Application Framework**

It is used to write applications for Android. Unlike other embedded mobile environments, Android applications are all equal, for instance, an applications which come with the phone are no different than those that any developer writes. The framework is supported by numerous open source libraries such as openssl, SQLite and libc. It is also supported by the Android core libraries. From the point of security, the framework is based on UNIX file system permissions that assure applications have only those abilities that mobile phone owner gave them at install time.

- **Dalvik Virtual Machine**

It is extremely low-memory based virtual machine, which was designed especially for Android to run on embedded systems and work well in low power situations. It is also tuned to the CPU attributes. The Dalvik VM creates a special file format (.DEX) that is created through build time post processing. Conversion between Java classes and .DEX format is done by included “dx” tool.

- **Integrated Browser**

Google made a right choice on choosing WebKit as open source web browser. They added a two pass layout and frame flattening. Two pass layout loads a page without waiting for blocking elements, such as external CSS or external JavaScript and after a while renders again with all resources downloaded to the device. Frame flattening converts founded frames into single one and loads into the browser. These features increase speed and usability browsing the internet via mobile phone.

- **Optimized Graphics**

As Android has 2D graphics library and 3D graphics based on OpenGL ES 1.0, possibly we will see great applications like Google Earth and spectacular games like Second Life, which come on Linux version. At this moment, the shooting legendary 3D game Doom was presented using Android on the mobile phone.

- **SQLite**

Extremely small (~500kb) relational database management system, which is integrated in Android. It is based on function calls and single file, where all definitions, tables and data are stored. This simple design is more than suitable for a platform such as Android.

- **Handset Layouts**

The platform is adaptable to both larger, VGA, 2D graphics library, 3D graphics library based on OpenGL ES 1.0 specifications, traditional smart phone layouts. An underlying 2D graphics engine is also included. Surface Manager manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.

- **Data Storage**

SQLite is used for structured data storage .SQLite is a powerful and lightweight relational database engine available to all applications.

- **Connectivity**

Android supports a wide variety of connectivity technologies including GSM, CDMA, Bluetooth, EDGE, EVDO, 3G and Wi-Fi.

- **Messaging**

SMS, MMS, and XMPP are available forms of messaging including threaded text messaging.

10. Web Browser

The web browser available in Android is based on the open-source WebKit application framework. It includes LibWebCore which is a modern web browser engine which powers both the Android browser and an embeddable web view.

11. Java Virtual Machine

Software written in Java can be compiled into Dalvik byte codes and executed in the Dalvik virtual machine, which is a specialized VM implementation designed for mobile device use, although not technically a standard Java Virtual Machine.

12. Media Support

Android will support advanced audio/video/still media formats such as MPEG-4, H.264, MP3, and AAC, AMR, JPEG, PNG, GIF.

13. Additional Hardware Support

Android is fully capable of utilizing video/still cameras, touch screens, GPS, compasses, accelerometers, and accelerated 3D graphics.

14. Development Environment

Includes a device emulator, tools for debugging, memory and performance profiling, a plug-in for the Eclipse IDE. There are a number of hardware

dependent features, for instance, a huge media and connections support, GPS, improved support for Camera and simply GSM telephony. A great work was done for the developers to start work with Android using device emulator, tools for debugging and plugin for Eclipse IDE.

- **Detailed Description of the topic**

Operation

- **Android Runtime**

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently.

The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

2. Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. It helps to manage security, memory management, process management, network stack and other important issues. Therefore, the user should bring Linux in his mobile device as the main operating system and

install all the drivers required in order to run it. Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user. Underlying all applications is a set of services and systems.

Architecture

The following diagram shows the major components of the Android operating system. Each section is described in more detail below.

- **Linux Kernel**

Android Architecture is based on Linux 2.6 kernel. It helps to manage security, memory management, process management, network stack and other important issues. Therefore, the user should bring Linux in his mobile device as the main operating system and install all the drivers required in order to run it. Android provides the support for the Qualcomm MSM7K chipset family. For instance, the current kernel tree supports Qualcomm MSM 7200A chipsets, but in the second half of 2008 we should see mobile devices with stable version Qualcomm MSM 7200, which includes major features:

- WCDMA/HSUPA and EGPRS network support
- Bluetooth 1.2 and Wi-Fi support
- Digital audio support for mp3 and other formats
- Support for Linux and other third-party operating systems
- Java hardware acceleration and support for Java applications
- Qcamera up to 6.0 megapixels
- gpsOne – solution for GPS
- and lots of other.

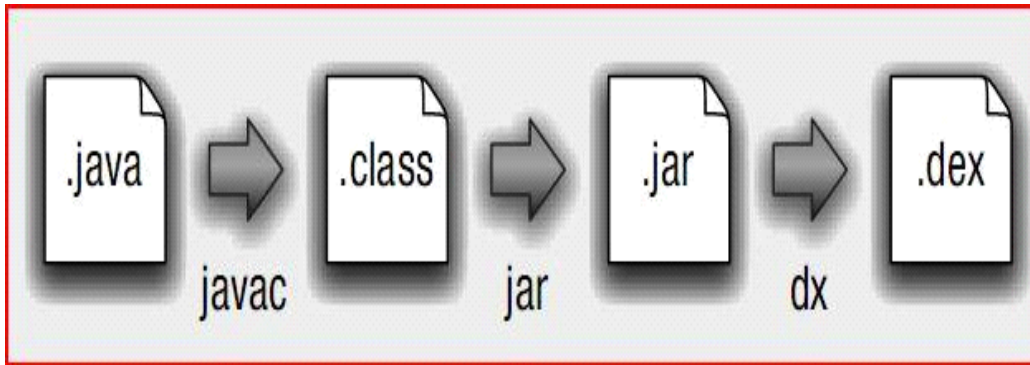
1. Library

In the next level there are a set of native libraries written in C/C++, which are responsible for stable performance of various components. For example, Surface Manager is responsible for composing different drawing surfaces on the mobile screen. It manages the access for different processes to compose 2D and 3D graphic layers. OpenGL ES and SGL make a core of graphic libraries and are used accordingly for 3D and 2D hardware acceleration. Moreover, it is possible to use 2D and 3D graphics in the same application in Android. The media framework was provided by PacketVideo, one of the members of OHA. It gives libraries for a playback and recording support for all the major media and static image files. FreeType libraries are used to render all the bitmap and vector fonts. For data storage, Android uses SQLite. As mentioned before, it is extra light rational management system, which locates a single file for all operations related to database. WebKit, the same browser used by Apples' Safari, was modified by Android in order to fit better in a small size screens.

2. Android Runtime

At the same level there is Android Runtime, where the main component Dalvik Virtual Machine is located. It was designed specifically for Android running in limited environment, where the limited battery, CPU, memory and data storage are the main issues.

Android gives an integrated tool “dx”, which converts generated byte code from .jar to .dex file, after this byte code becomes much more efficient to run on the small processors.



As the result, it is possible to have multiple instances of Dalvik virtual machine running on the single device at the same time. The Core libraries are written in Java language and contains of the collection classes, the utilities, IO and other tools.

3.Application Framework

After that, there is Application Framework, written in Java language. It is a toolkit that all applications use, ones which come with mobile device like Contacts or SMS box, or applications written by Google and any Android developer. It has several components. The Activity Manager manages the life circle of the applications and provides a common navigation back stack for applications, which are running in different processes. The Package Manager keeps track of the applications, which are installed in the device. The Windows Manager is Java programming language abstraction on the top of lower level services that are provided by the Surface Manager. The Telephony Manager contains of a set of API necessary for calling applications. Content Providers was built for Android to share a data with other applications, for instance, the contacts of people in the address book can be used in other applications too. The Resource Manager is used to store localized strings, bitmaps, layout file descriptions and other external parts of the application. The View System generates a set of

buttons and lists used in UI. Other components like Notification manager is used to customize display alerts and other functions.

4.Application Layer

At the top of Android Architecture we have all the applications, which are used by the final user. By installing different applications, the user can turn his mobile phone into the unique, optimized and smart mobile phone. All applications are written using the Java programming language.

3.2:Fact Finding

- The world today we live in is dominated by the machines and technology and one of them is 4wheelers that we use in today's world on the daily basis. Most of the people uses cars for commuting from one place to another and they often triggers with some problems in their cars, they can search the garages and find better garages in the area and see reviews given by others based on that they can go to better garages and get their problem solved. The app will be most useful for the outside people who can use the app and be happy by getting service done after getting their service done form mechanic.

3.3:Feasibility Study:

The current system of generating pass has the following drawbacks:

- 1) The current system of car service is based on manual and paperwork bases.
- 3) It is very hard to get the car to take to the service place or repairing when people have very busy scheduled routine in their life.

Hence looking at the above disadvantages the SCS has been developed. The SCS minimizes time requirements, manpower as well as the paper work. Users can get the range of categories and range of services under one application to get their car serviced. They can also have the option of online payment to pay the money. They can also navigate the nearest service stations to find immediate service to their 4 wheelers.

3.4:System Requirement Engineering

a) Requirement Specifications

I) Functional Requirements

There are mainly three modules in Application

1) Find Garage

- 1) Smart device require for application.
- 2) Internet connectivity requires.
- 3) Application has been installed at user device.

ii) Non-Functional Requirements

1) Security

The application has the primary requirement to login into the system to do the next task. Hence this username and password of the user should be maintained properly and should not allow any unauthorized user to enter into the system.

The SCS obtains the data for verification across the server. Hence the data transfer across the server should be secure. Also the data provided should be verified twice so that a user with incorrect detail should not be allowed to register into the system.

2) Compatibility

The SCS should be able to run on the android phones as well as on other android enabled devices like tablet etc. It should also be developed in such a way that it should run on all the older versions of android.

3) Portability

The SCS is an android based application running on mobile phones. Hence it should not take large amount of space such that it does not fit in the user's device. The app should display the results rapidly minimizing the delay.

b) Requirement Definition

Observations:

- The world today we live in is dominated by the machines and technology and one of them is 4wheelers that we use in today's world on the daily basis. Most of the people uses cars for commuting from one place to another and they often triggers with some problems in their cars, they can search the garages and find better garages in the area and see reviews given by others based on that they can go to better garages and get their problem solved. The app will be most useful for the outside people who can use the app and be happy by getting service done after getting their service done form mechanic.

Definition:

- The SCS is developed to solve the problem of the people who uses the car as a source of a medium to commute from one place to another place. The SCS aims to reduce the time of finding good garages in the city and around the city.
- The major problem that the car holder faces is that they don't have enough knowledge about garages especially for the people who are not the native livers of the city and they have a great advantage using this application and they can search and navigate the location of garage through google maps.
- Users have to download this app. Then, they have to register themselves on the application by providing some registration information. Both the mechanics and customers have to register themselves and then their account will get verified by email verification, if they get verified then they will be able to access the app features and do whatever they want.

c) Data Dictionary

CUSTOMER TABLE

COLUMN NAME	DATA TYPE	CONSTRAINT
PK_CUSTOMERID	Bigint	PRIMARY KEY
CUSTOMER_NAME	nvarchar(500)	NOT NULL
ADDRESS	nvarchar(MAX)	NOT NULL
FK_AREAID	Bigint	FOREIGL KEY – AREA
CITY	nvarchar(50)	NOT NULL
PINCODE	nvarchar(50)	
STATE	nvarchar(50)	
EMAIL	nvarchar(500)	NOT NULL
PHONE	nvarchar(50)	
MOBILE	nvarchar(50)	NOT NULL
BIRTHDAY_DATE	Date	
ANNIVERSARY_DATE	Date	
FK_CREATED_BY	Bigint	FOREIGL KEY – USER
FK_MODIFIED_BY	Bigint	FOREIGL KEY – USER
CREATED_DATE	Datetime	
MODIFIED_DATE	Datetime	
IS_DELETED	Bit	
IS_ACTIVE	Bit	
USERNAME	nvarchar(50)	NOT NULL
PASSWORD	nvarchar(50)	NOT NULL
IS_EMAIL_VERIFY	Bit	

AREA TABLE

COLUMN NAME	DATA TYPE	CONTRAIINT
PK_Area	Bigint	PRIMARY KEY
Area_Name	nvarchar(50)	
FK_CREATED_BY	Bigint	FOREIGL KEY – USER
FK_MODIFIED_BY	Bigint	FOREIGL KEY – USER
CREATED_DATE	Datetime	
MODIFIED_DATE	Datetime	
IS_DELETED	Bit	
IS_ACTIVE	Bit	

FEEDBACK TABLE

COLUMN NAME	DATA TYPE	CONSTRAINT
PK_FeedBackID	Bigint	PRIMARY KEY
FK_MechanicID	Bigint	FOREIGL KEY – MECHANIC
FK_CUSTOMERID	Bigint	FOREIGL KEY – CUSTOMER
FeedBack_Subject	nvarchar(200)	
FeedBack_Descriptions	nvarchar(1000)	
FK_CREATED_BY	Bigint	FOREIGL KEY – USER
FK_MODIFIED_BY	Bigint	FOREIGL KEY – USER
CREATED_DATE	Datetime	
MODIFIED_DATE	Datetime	
IS_DELETED	Bit	
IS_ACTIVE	Bit	

MECHANIC TABLE

COLUMN NAME	DATA TYPE	CONSTRAINT
PK_MechanicID	Bigint	PRIMARY KEY
Mechanic_NAME	nvarchar(50)	
USERNAME	nvarchar(50)	
PASSWORD	nvarchar(50)	
ADDRESS	nvarchar(1000)	
EMAIL	nvarchar(500)	
CITY	nvarchar(50)	
STATE	nvarchar(50)	
M_Garage_NAME	nvarchar(50)	
Fk_Area	Bigint	
IS_ACTIVE	Bit	
NOTES	nvarchar(MAX)	
FK_CREATED_BY	nvarchar(MAX)	FOREIGL KEY-USER
FK_MODIFIED_BY	Bigint	FOREIGL KEY-USER
CREATED_DATE	Datetime	
MODIFIED_DATE	Datetime	
IS_DELETED	Bit	
Longitude	nvarchar(50)	
Latitude	nvarchar(50)	

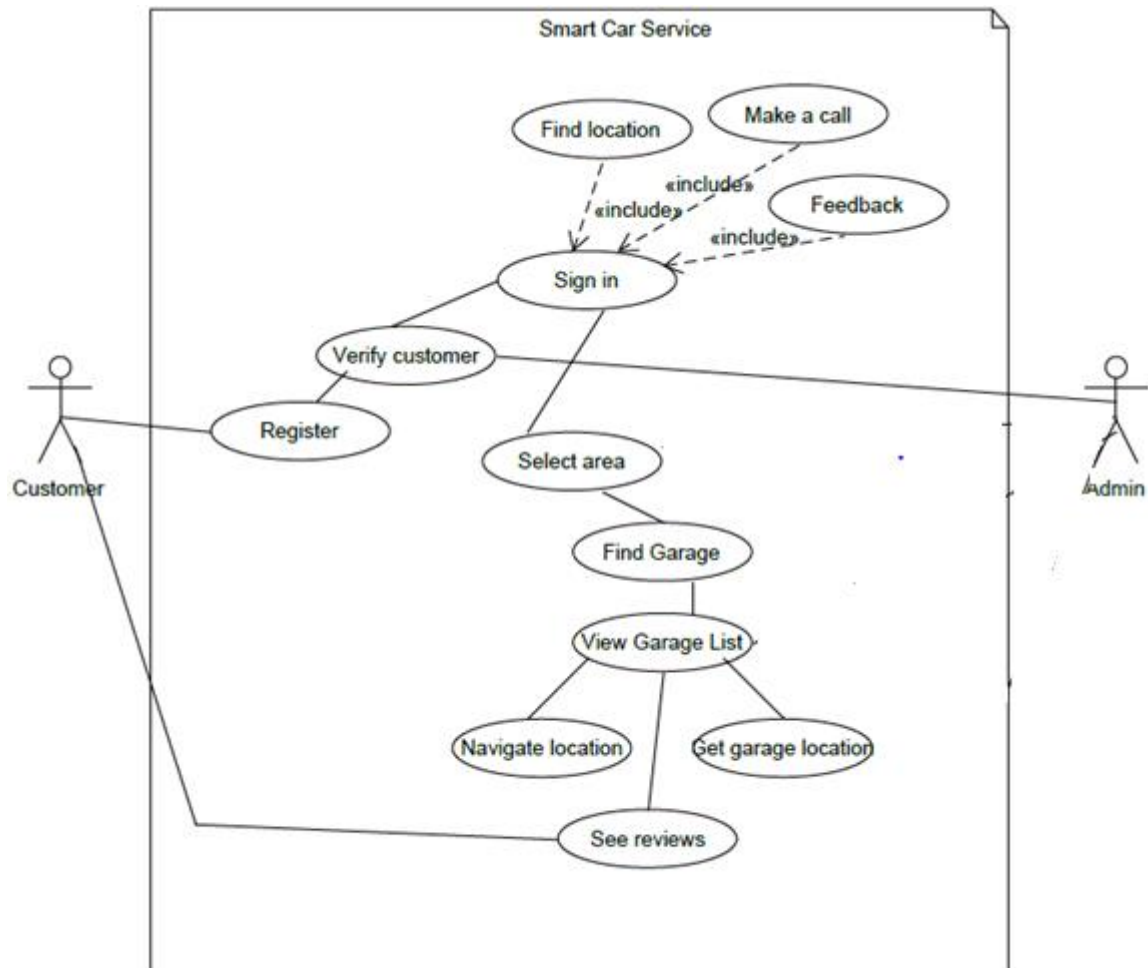
USER TABLE

COLUMN NAME	DATA TYPE	CONSTRAINT
PK_USERID	Bigint	PRIMARY KEY
NAME	nvarchar(500)	
USERNAME	nvarchar(50)	
PASSWORD	nvarchar(50)	
NOTES	nvarchar(MAX)	
FK_CREATED_BY	Bigint	FOREIGL KEY-USER
FK_MODIFIED_BY	Bigint	FOREIGL KEY-USER
CREATED_DATE	Datetime	
MODIFIED_DATE	Datetime	
IS_DELETED	Bit	
IS_ACTIVE	Bit	

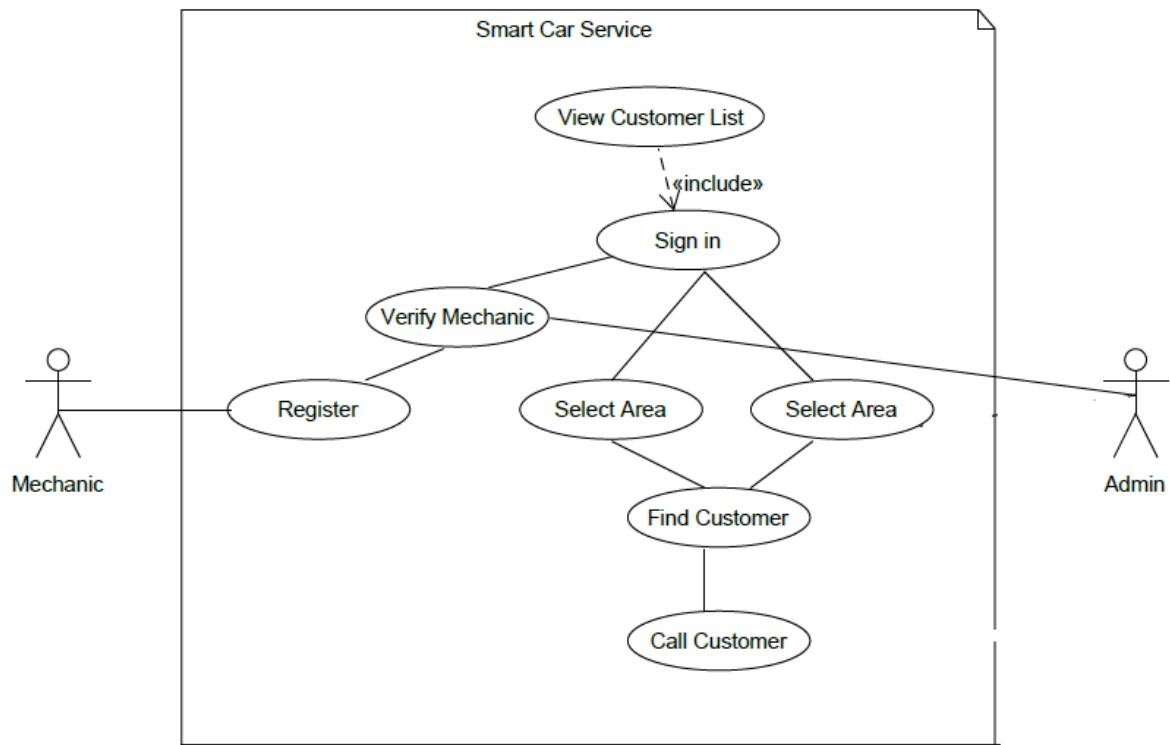
CHAPTER-4-SYSTEM DESIGN

4.1:UsecaseDiagram

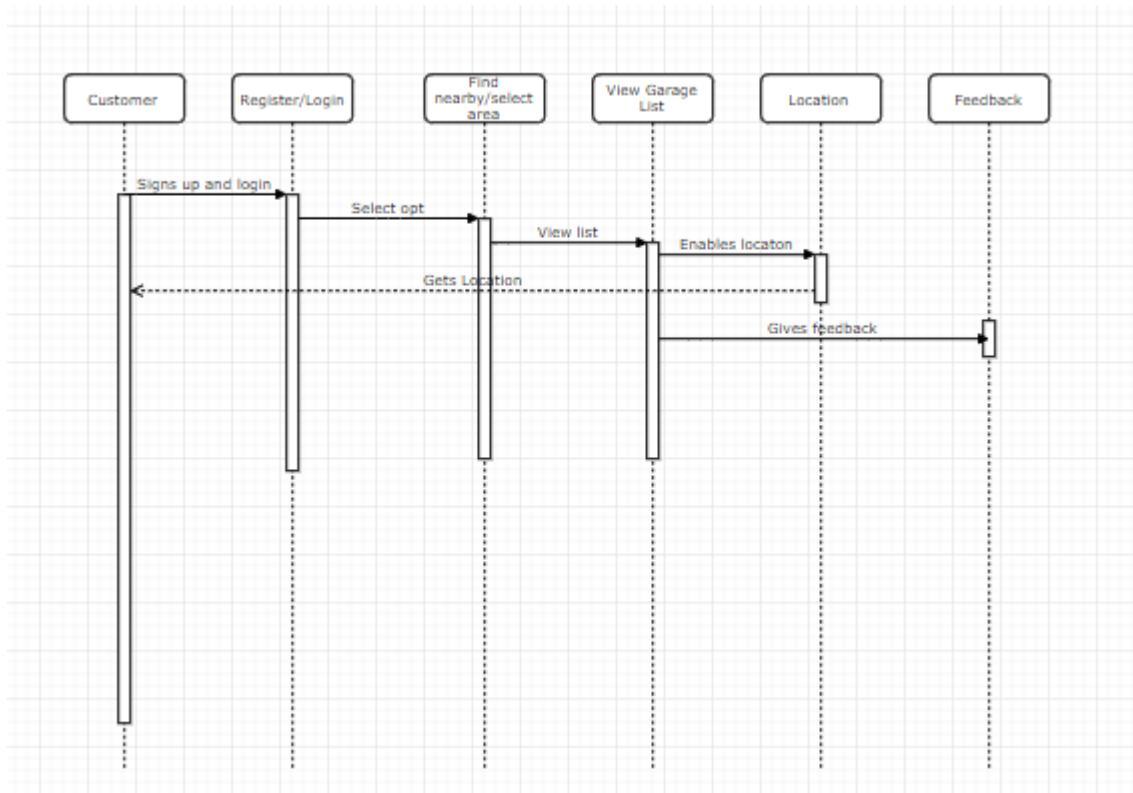
Customer Side



Mechanic Side

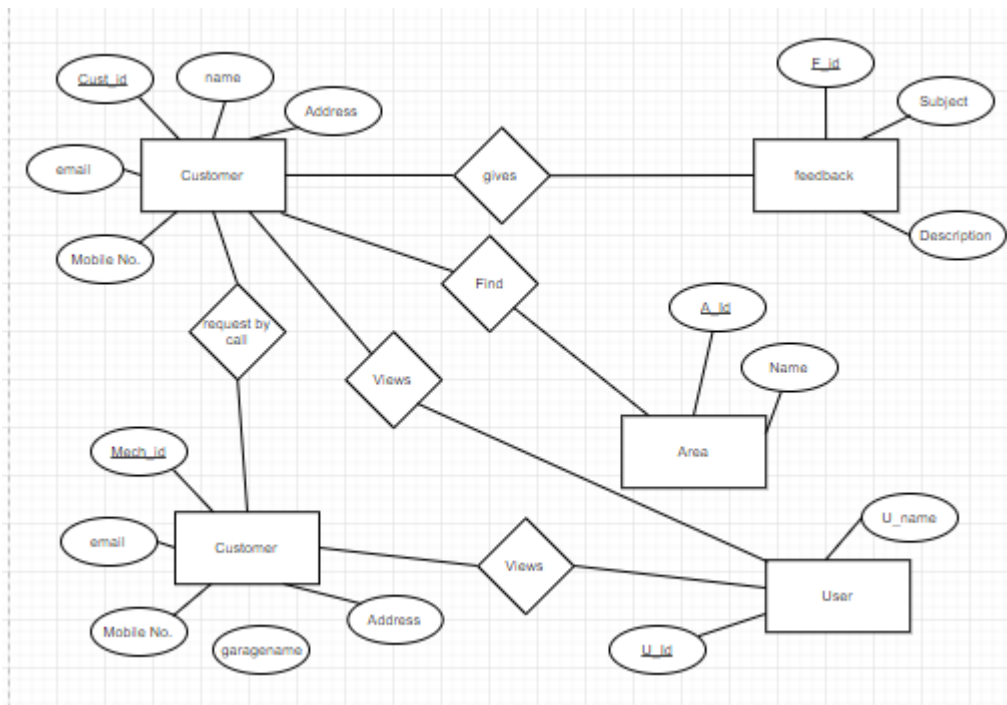


4.2:Sequence Diagram

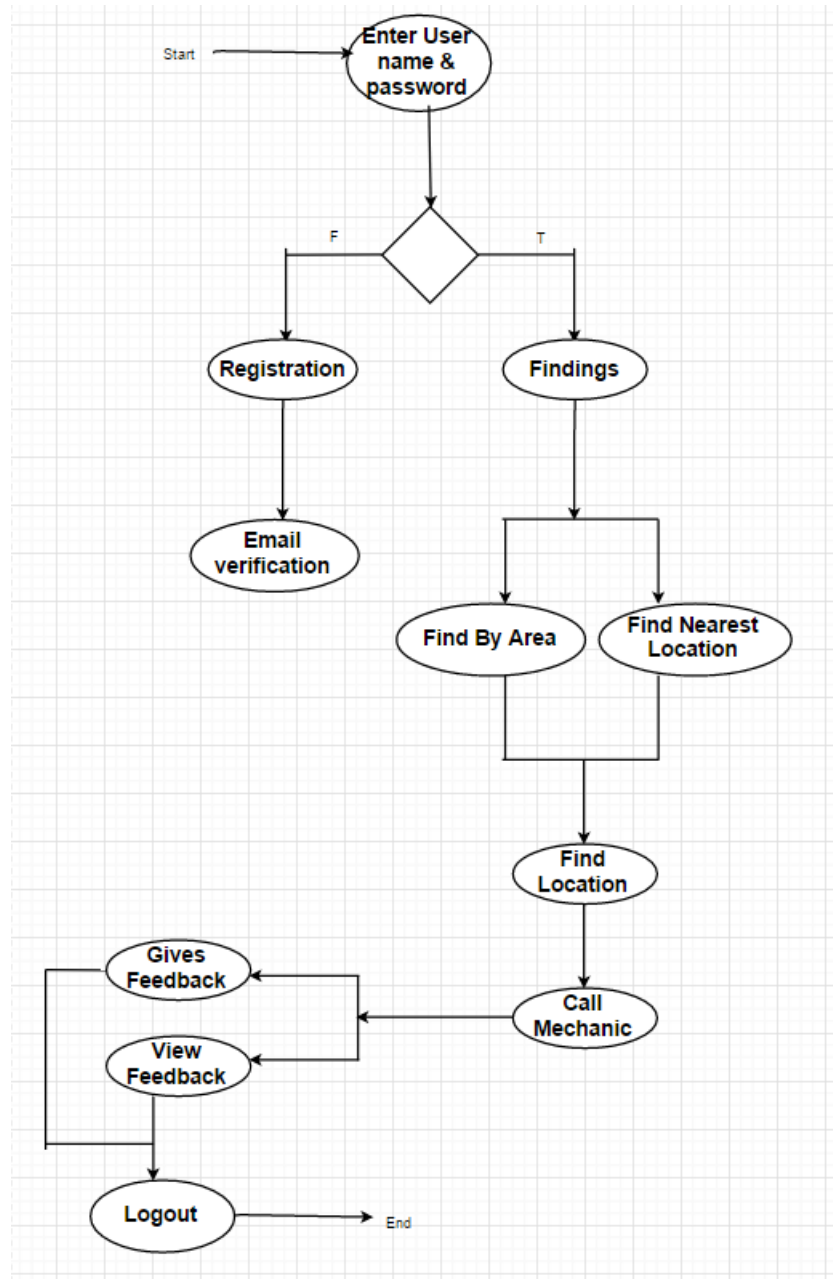


*

4.3:E-R Diagram

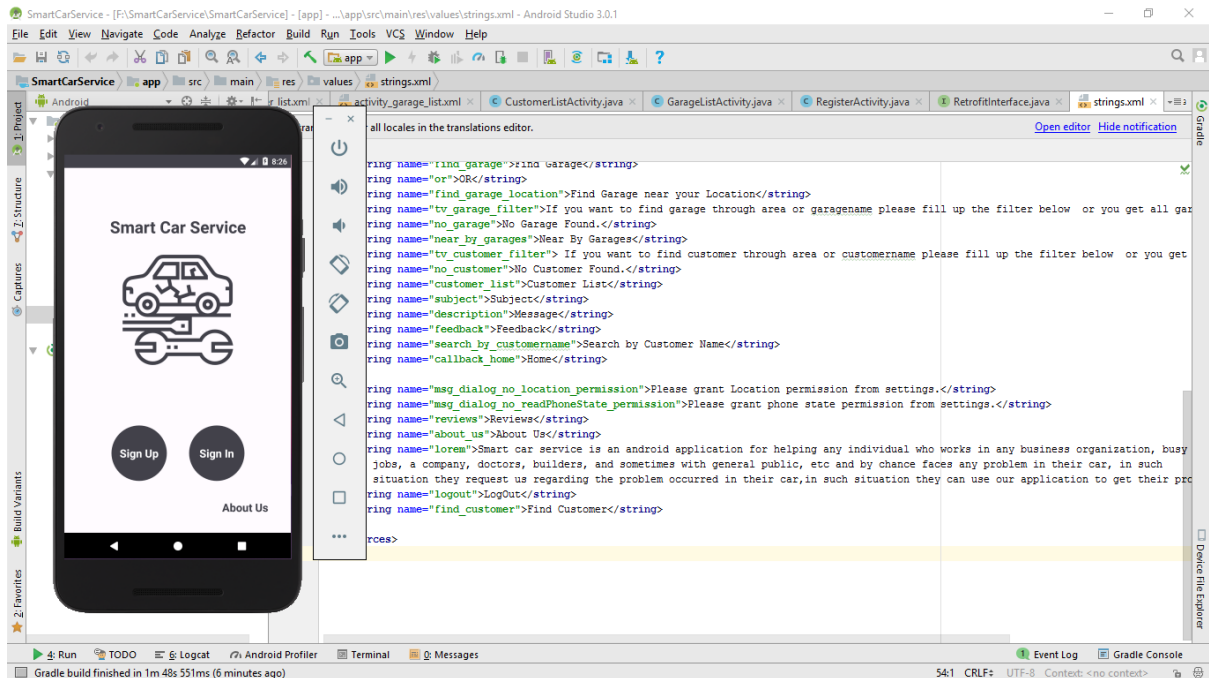


4,5:Activity Diagram



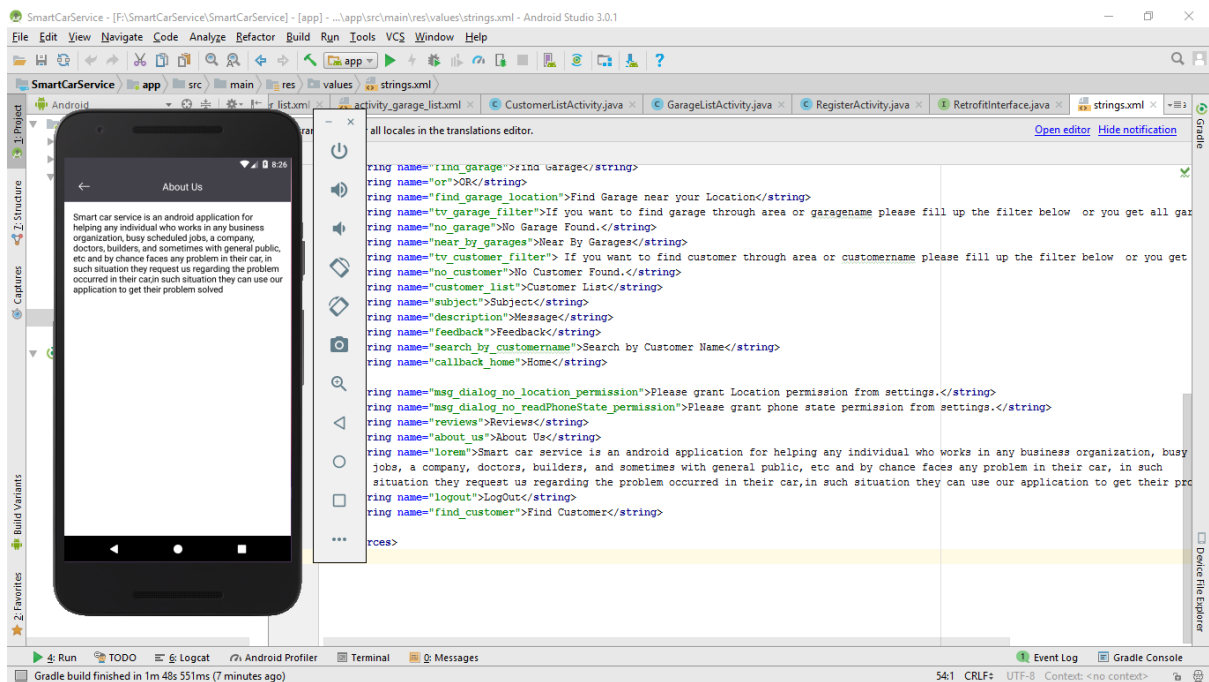
4.6: User Design

1.Home Page:



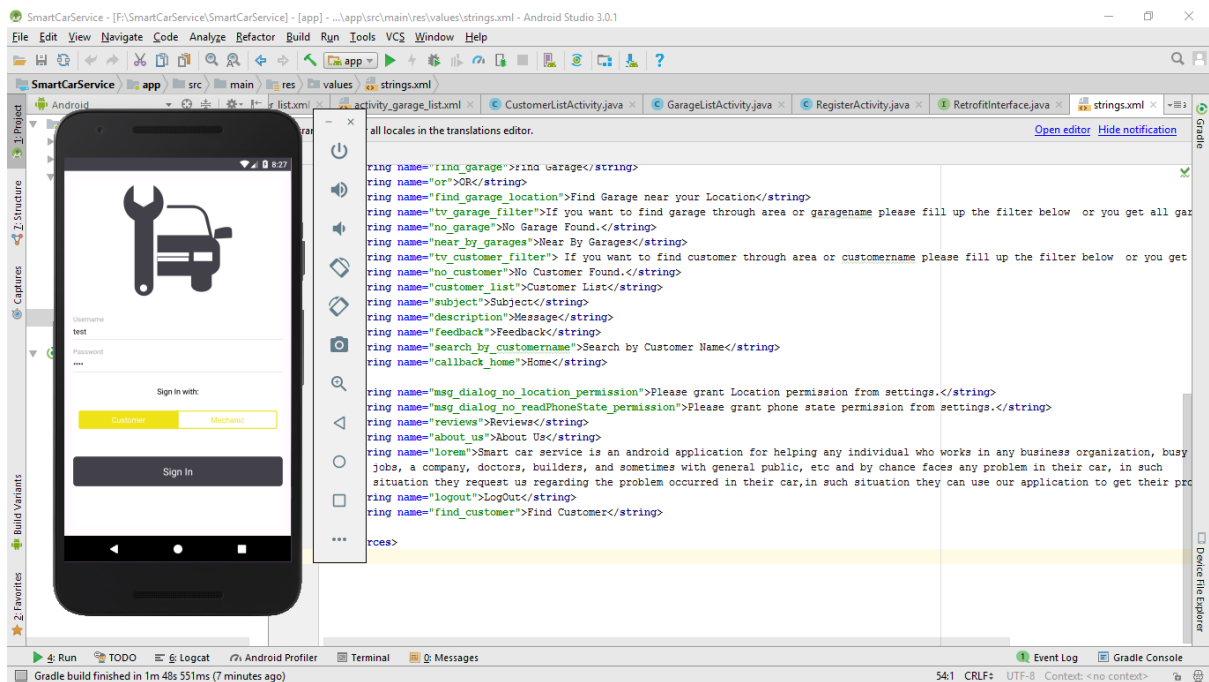
This is Home page of our application using this customer and mechanic register and login itself.

2. about Us:



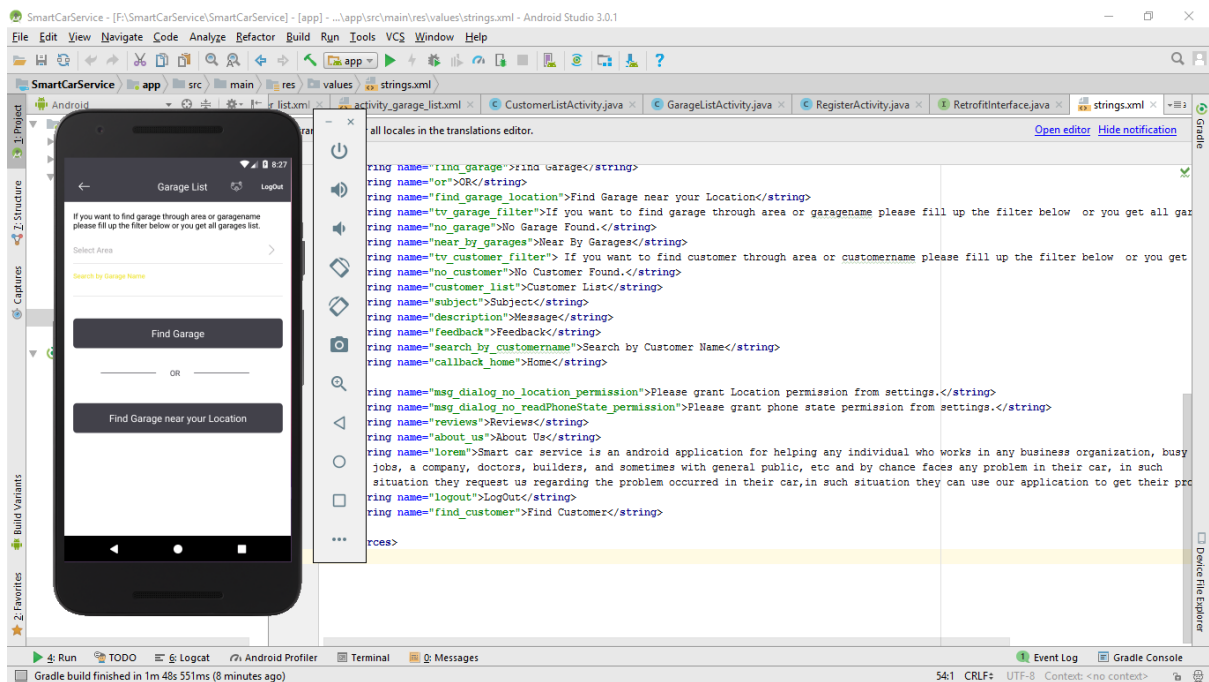
This is about us page of system using this user read our application details.

3. Sign In:



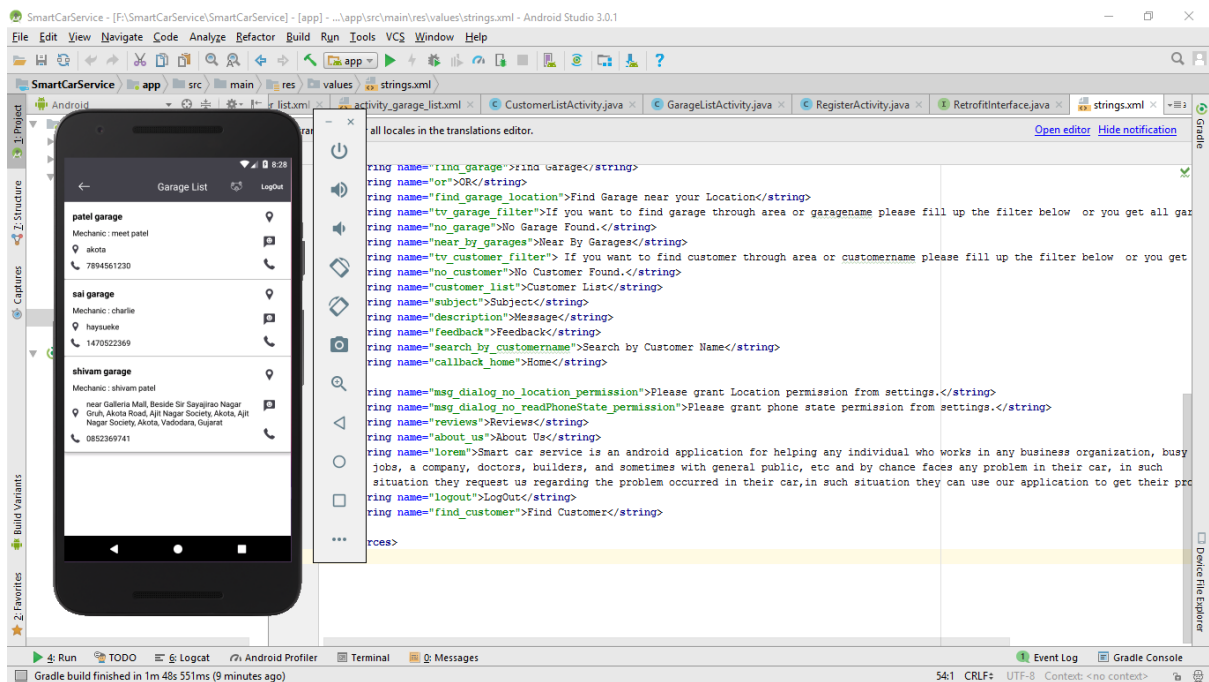
This is registration page of our application. Using this page customer and mechanic can sign-in/log-in in our app.

4.Find Garage:



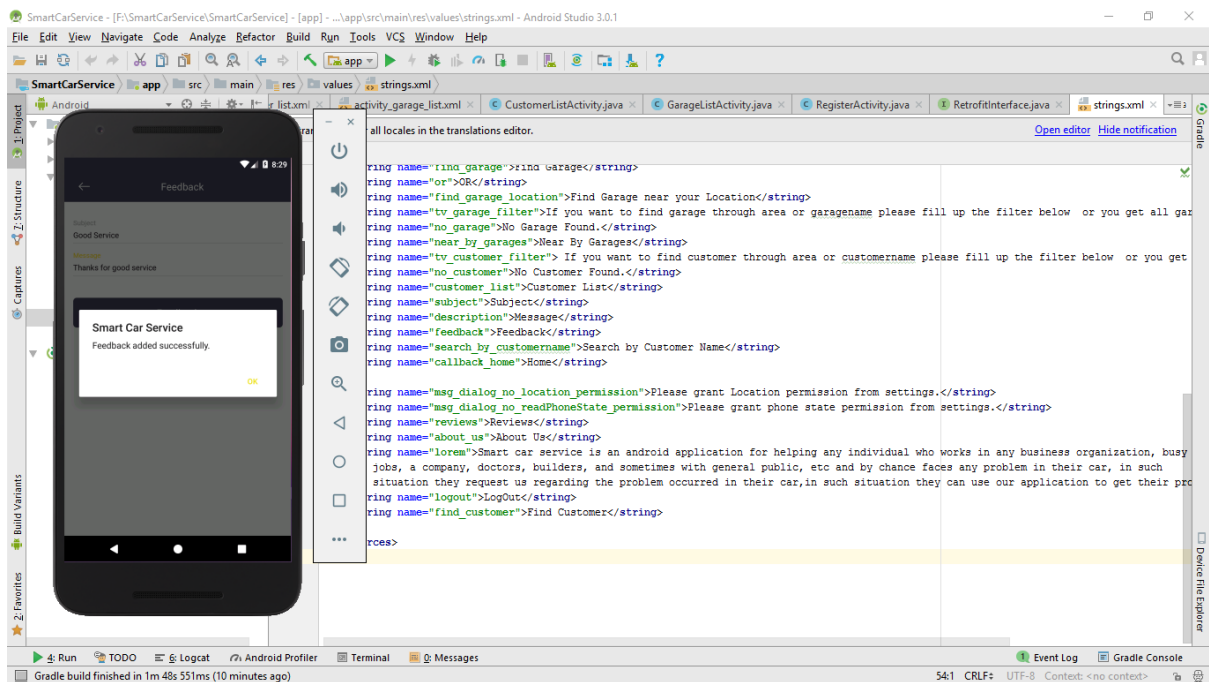
This is main activity of our app that gives option to customer that he/she find garage by clicking find garage button or find nearby garage using find garage near your location button.

5.View Garage list:



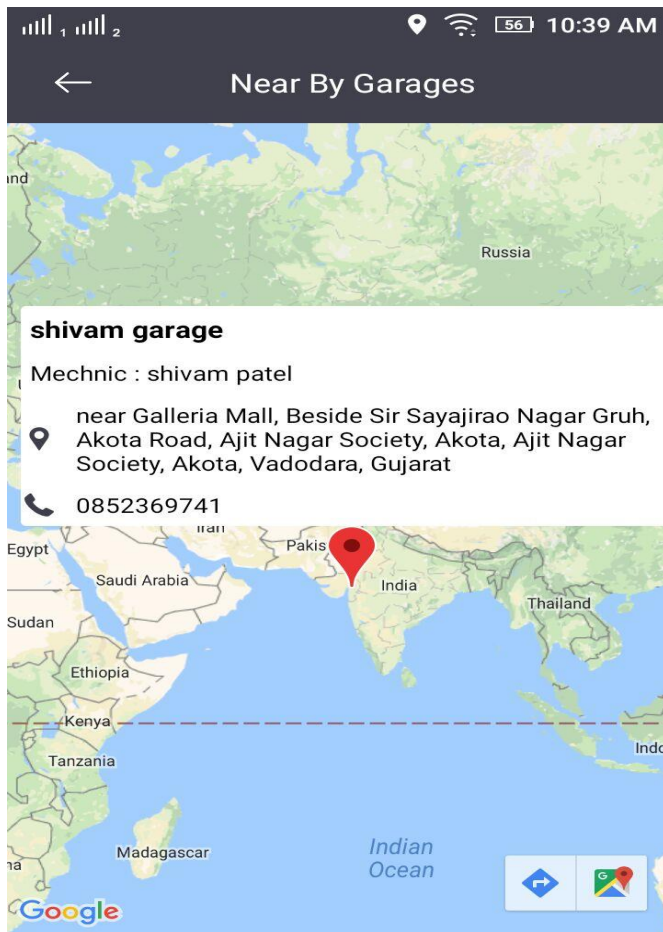
This is view garage list page of application that display garage list to the customer.

6.Feedback:



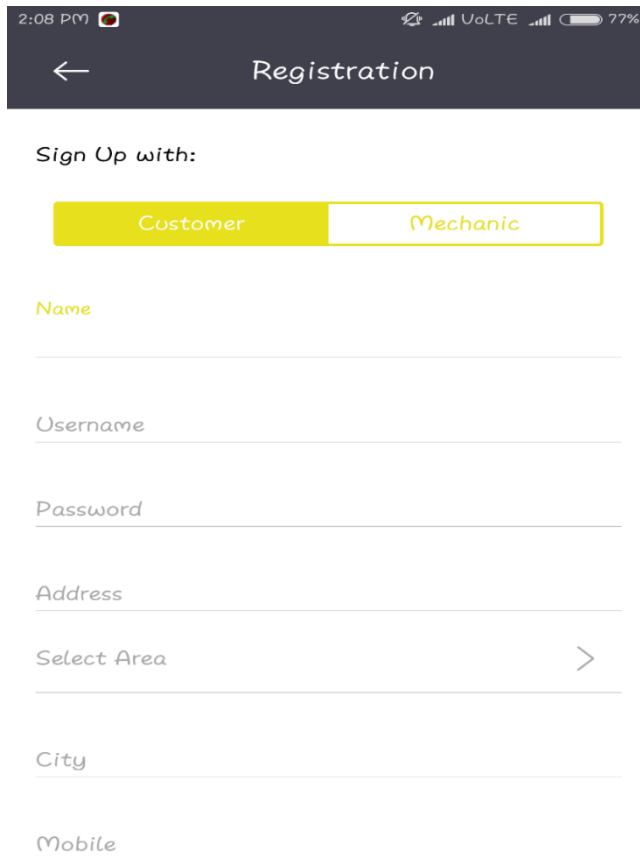
This is feedback page of our application using this page customer gives his/her opinion or suggestion on mechanic's facility of garage.

7.Nearby Garage:



This page shows mechanic list on map for customer understanding.
Customer show the detail of mechanic which he/she select on map.

8.Registration Page:



2:08 PM 77%

← Registration

Sign Up with:

Customer Mechanic

Name

Username

Password

Address

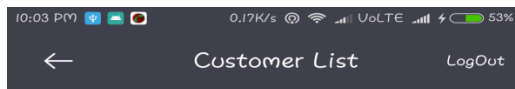
Select Area >

City

Mobile

This is the page of registration of customer and mechanic using this page customer and mechanic both can register itself.

9.Homepage for mechanic:



If you want to find customer through area or customername please fill up the filter below or you get all customers list.

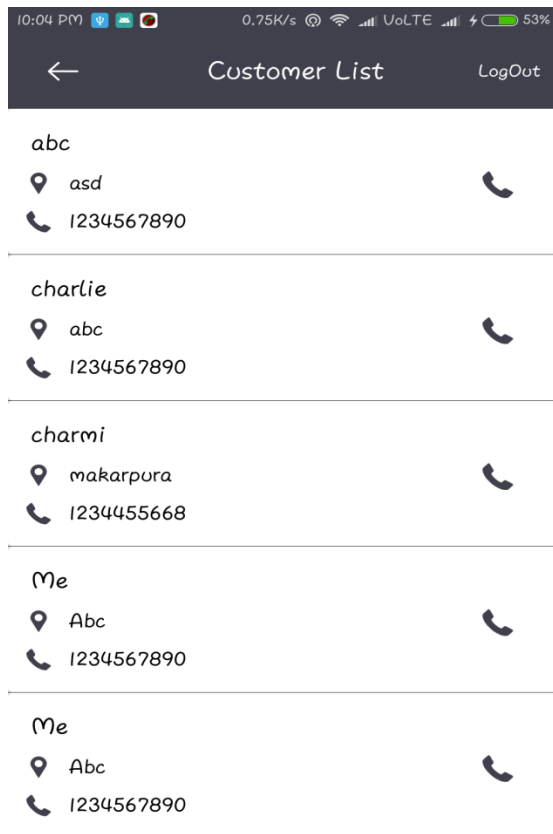
Akota >

Search by Customer Name

Find Customer

This is Home page of mechanic side in our application using this page mechanic find customers by selecting areas.

11. View customer list



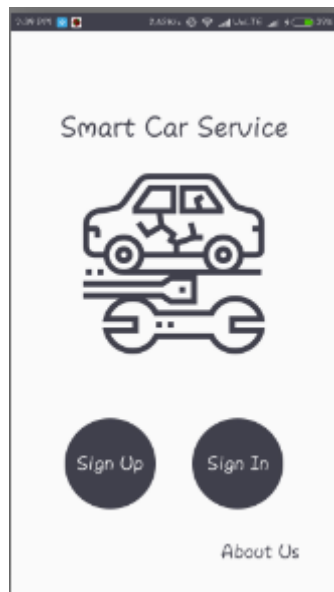
This is view customer list page in this list of customer display to mechanic.

CHAPTER-5-USER MANUAL

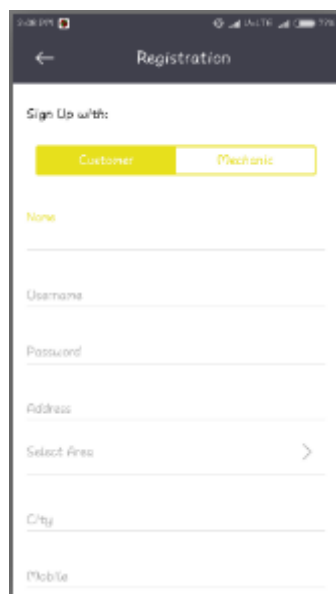
5.1 USER MANUAL

This is the user manual that guides the user to use the application. The manual involves steps the user must follow to effectively use the android application.

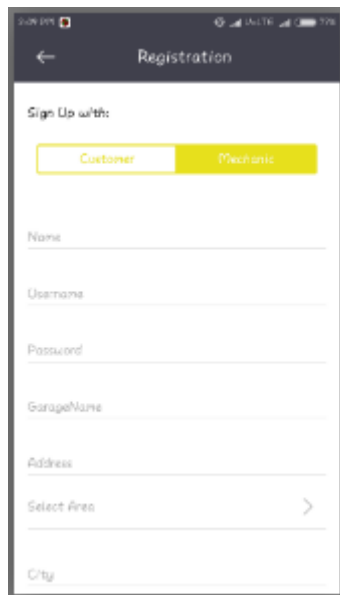
- 1) User can download the app from the Google Play Store.
- 2) Upon downloading the app, the app requests for various permissions that need to be allowed by the user.
- 3) Click on install button and the app will be installed on the device and can be used.
- 4) Installing the app, it will first show the splash screen which shows the application's logo after home page as given below.



- 5) Next is the REGISTRATION page where the customers need to enter their details to go ahead.

The image shows the 'Registration' page of the application. It has a title bar with a back arrow and the word 'Registration'. Below the title bar, there is a section 'Sign Up with:' with two buttons: 'Customer' and 'Mechanic'. Below this, there are several input fields: 'Name', 'Username', 'Password', 'Address', 'Select Area' (with a right arrow), 'City', and 'Mobile'.

6) Next is the REGISTRATION page where the mechanics need to enter their details to go ahead.



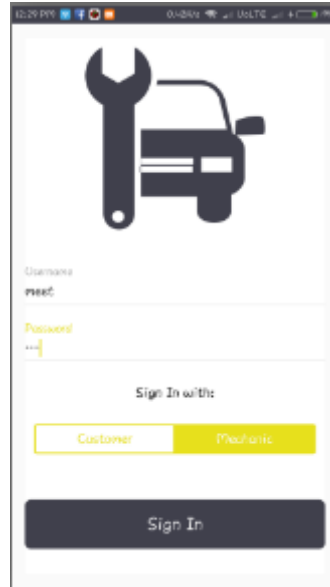
The image shows a mobile app interface for the 'Registration' page. At the top, there is a back arrow and the title 'Registration'. Below this, a section titled 'Sign Up with:' contains two yellow buttons: 'Customer' and 'Mechanic'. The 'Mechanic' button is selected. Below the buttons are several input fields: 'Name', 'Username', 'Password', 'GarageName', 'Address', 'Select Area' (with a right arrow), and 'City'.

7) Next is the LOGIN page where the customers need to enter their details to go ahead.

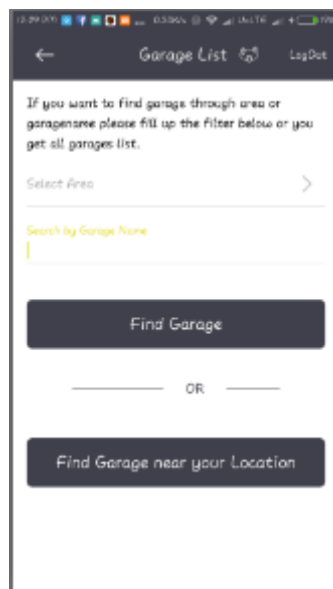


The image shows a mobile app interface for the 'Login' page. At the top, there is a status bar with the time '7:28 PM' and battery level '70%'. Below this is a large icon of a wrench and a car. The form contains two input fields: 'Username' with the text 'test' and 'Password' with masked characters. Below these fields is a section titled 'Sign In with:' with two yellow buttons: 'Customer' and 'Mechanic'. The 'Customer' button is selected. At the bottom is a large dark blue button labeled 'Sign In'.

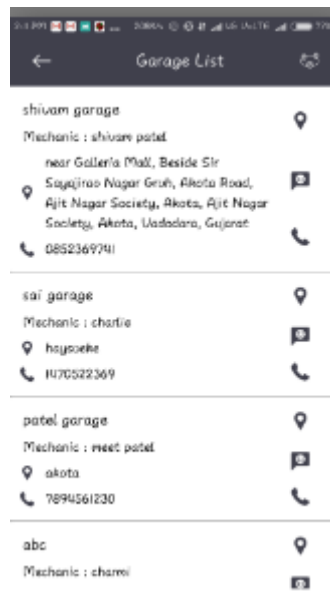
8) Next is the LOGIN page where the mechanics need to enter their details to go ahead.



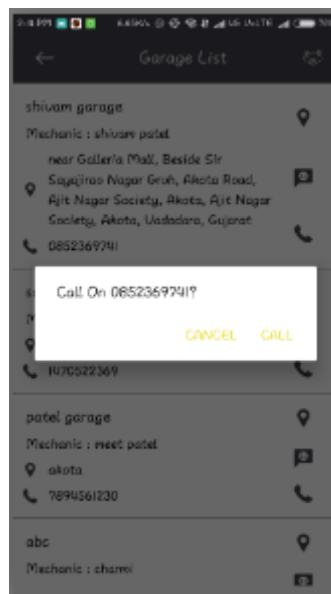
9) Once you login in app you have to give permissions to allow manage phone calls and access location of device to app. after that you can find garage using find garage page.

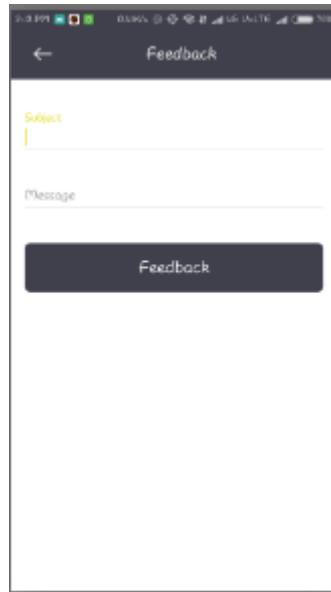


10) Once you click on that find garage near your location and, a list of garage will appear which show your number of mechanics are register in app.



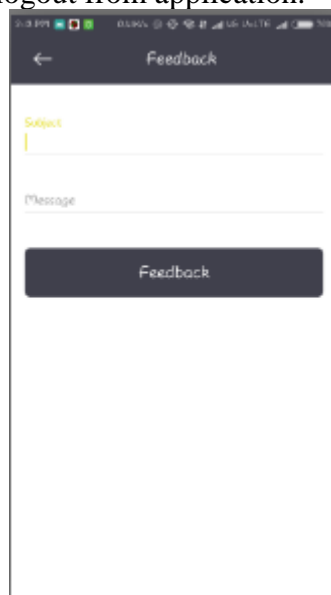
2025 RELEASE UNDER E.O. 14176



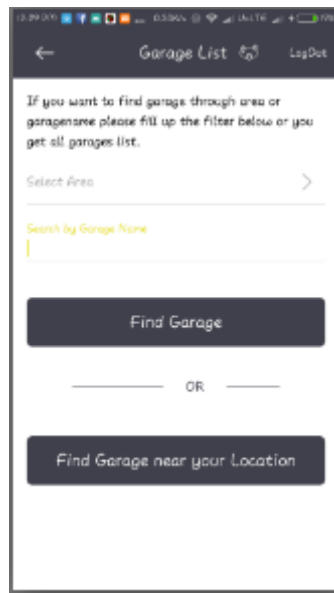


A screenshot of a mobile application's feedback form. The form has a dark header with a back arrow and the title "Feedback". Below the header, there are two input fields: "Subject" and "Message". The "Subject" field has a yellow cursor. Below the input fields is a dark button labeled "Feedback".

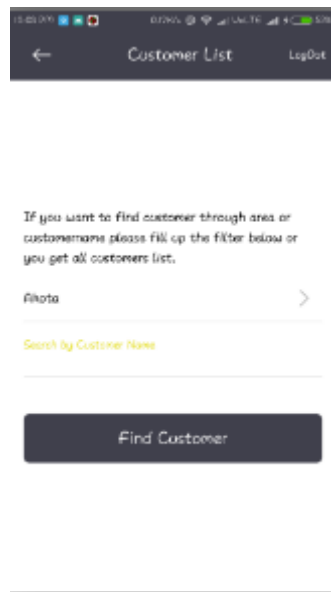
12) Once you reach at mechanic and mechanic complete it work you can give feedback in app and you can logout from application.

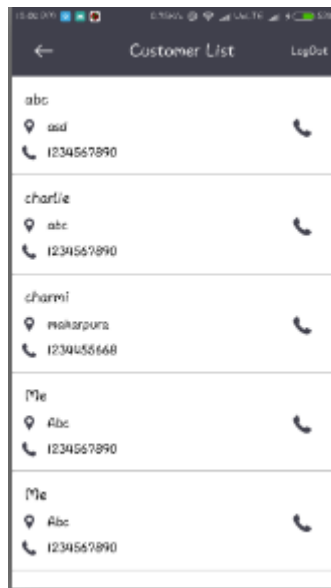


A screenshot of a mobile application's feedback form, identical to the one above. It features a dark header with a back arrow and the title "Feedback". Below the header, there are two input fields: "Subject" and "Message". The "Subject" field has a yellow cursor. Below the input fields is a dark button labeled "Feedback".

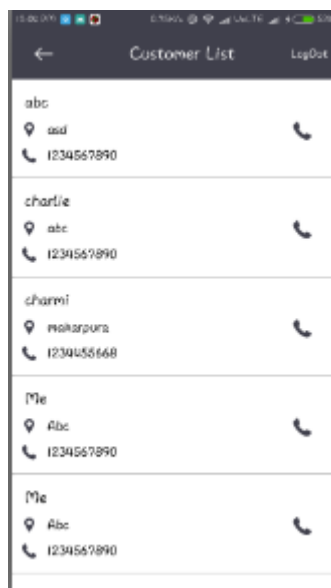


13) Once mechanic login in application he/she allows to gives permission to app. After login mechanic, he/she select the area and click on find customer button to view customer list of that area.





14) After showing customer list mechanic can logout from application by clicking logout button at top of the customer list page.



CHAPTER-6-TOOLS AND **TECHNOLOGY**

6.1 Tools and Technology

Front End	Android
Back End	C#.net, My SQL server 2012
Diagram designing tools	Microsoft office Visio 2007, 2013
OS - Platform	Windows
Tools	Android Studio, Visual studio, sql studio manager

CHAPTER-7-TESTING

7.1 TESTING

The aim of testing process is to identify all defects existing in software product. If the program fails by any means or gives unexpected results than the failure occurred are noted for later debugging and corrections.

7.2 TESTING PLANS

The objective of the system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified and ensure that the computer system and the associated clerical and other procedures work together. Systems are not designed as entire Systems not they are tested as single system. The analyst must perform both unit and system testing.

Different types of testing methods are available. We have tested our system for different aspects like:

Does the application meet the goals for which it has been designed? This was a very important question that stood before us as the application was designed to be implemented on such a large network.

To fulfill its goal of being able to run on different system we went through a series of test at different places where this is supported to be used the most. As we need to make our system efficient enough, we need to test it thoroughly.

Finally we tested the system with the real-time data, for which it is actually designed. We are almost successful in satisfying our customers as it was designed according to their requirements. But it is very necessary to maintain this application and so our work is not still over.

7.3 TESTING STRATEGY

The test strategy is a formal description of how a software product will be tested. A test strategy is developed for all levels of testing as and when required. The testing team analysis the requirements, writes the test strategy and reviews the plan with the project team. The test plan may include test cases, conditions and the test environment, a list of related tasks, pass/fail criteria and risk assessment.

As we were working in a team it was a bit difficult to check the work done as no individual was assigned for that particular task or work. Moreover checking the work done was very important to reduce risk factor. Right from the starting we planned to adopt one technique called Assessment by rotation. Thus the work done by one member was assessed by the other for some time and again revolved for other level check reduced the errors.

7.4 TESTING METHODS

Testing methods are mainly divided into two parts:

- 1) Black Box Testing
- 2) While Box Testing

➤ **Black box testing**

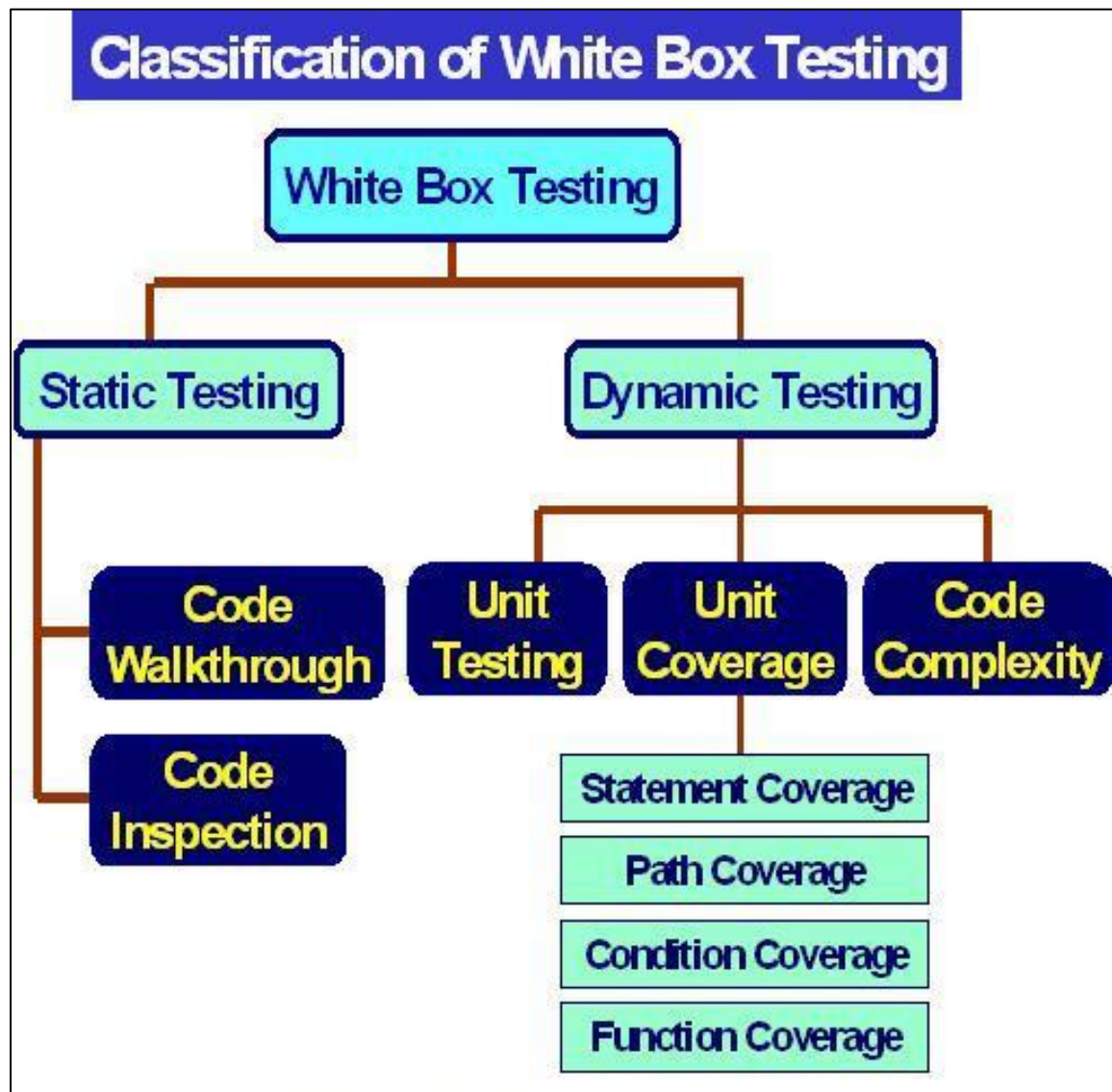
It is the testing without the knowledge of the internal working of the item being tested. For this testing test groups are often used. Due to the nature of this testing the test planning can begin as soon as the specifications are written. This testing has some advantages like it is more effective on larger units of code than glass box testing, tester needs no knowledge of implementation, including specific programming languages, tester and programmer are independent of each other, tests are done from a user's point of view, will help to expose any ambiguities or inconsistencies in the specifications, test cases can be designed as soon as the specifications are complete.

It focuses on the functional requirements of the software. It enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of the project. Black box testing attempts to find errors in the following categories: incorrect or missing links, interface errors, errors in data structure, behavior or performance errors and the initialization and termination errors.

➤ **White box testing**

It is the testing strategy that deals with the internal logic and structure of the code. White box testing is also called as the glass testing, structural testing, open testing or the clear box testing. The advantages of the white box testing are that as the knowledge of internal coding structure is needed, it becomes easy to find out which type of input data can help in testing the application effectively. It helps in optimizing the code helping in removing the extra code which may cause hidden

defects in the system. It is test case design method that uses the control structure of the procedural design to derive the test cases. Using white box testing techniques the software engineer can derive test cases that: guarantee that all independent paths within a module have been exercised at least once, exercise all global decisions on their true and false sides, execute all loops at their boundaries and within their operation bounds and also exercise the internal data structure to ensure their validity.



The other sub testing techniques are as follows:

➤ **Unit Testing**

Unit testing focuses on verification effort on the smallest unit of the project that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit testing.

➤ **Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

➤ **Task Testing**

Task testing helps the system to test whether all the tasks are carried out properly or not. It also solves logical errors. By carrying this test we were assured that all processes in our project are appropriate.

➤ **Behavioral Testing**

This testing is used to check the behavior of the system. This test helped us to develop the test case for the particular system. This testing thus helps to test the different events associated with the system.

➤ **Inter Task Testing**

There are several processes or tasks which are carried simultaneously. Moreover this testing also helps to test the errors associated with the data storage activities. This testing helped our project to solve the error while data designing which is very much important for a successful project.

➤ **Final Testing**

After the integration of all the modules we deploy whole application. One of the member of the development team will do the testing of the whole project and create the excel sheet of the bugs. After the completion of the solving of these errors the project would be ready to use.

➤ Alpha Testing

Alpha testing is done by the person not involved in the project. In our case it would be done by out concerned faculty members itself.

7.5 TEST CASES

In software engineering the most common definition of a test case is set of conditions or variables under which a tester will determine if a requirement or use case upon an application is partially or fully satisfied. In the situation each sub requirement must have at least one test case.

- A good test has a high probability of finding an error. To achieve this goal the tester must understand the software and attempt to develop a mental picture of how the software might fail.
- A good test is not redundant. Testing time and resources are limited. There is no point in conducting that has the same purpose as another test.
- A good test should be best of the all. In a group tests that have a similar intent, time and resources limitation may mitigate towards the execution of only a subset of these tests.
- A good test should neither be too simple nor be too complex.

APPENDIX

<https://www.tutorialspoint.com/android/index.htm>

<https://developer.android.com/guide/>

<https://www.javatpoint.com/android-tutorial>

<https://www.tutorialspoint.com/webservices/index.htm>

<https://www.javatpoint.com/web-services-tutorial>

https://www.tutorialspoint.com/asp.net/asp.net_web_services.htm

<https://www.tutorialspoint.com/asp.net/index.htm>

<https://www.w3schools.com/asp/default.asp>