**A PROJECT REPORT**

**ON**

**IRV-INDIAN REGISTERED VEHICLES**

*Submitted by*

**DARSHIL PATHAK (140410107008)**

**KARNA DESAI (140410107010)**

**RUCHITA PARMAR (140410107046)**

***In fulfillment***

***Of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER ENGINEERING**

****

**SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY**

**VASAD**

**Gujarat Technological University, Ahmedabad**

**SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY,**

**VASAD**

**COMPUTER ENGINEERING DEPARTMENT**

****

**CERTIFICATE**

**Date:**

**This is to certify that the project entitled “IRV-INDIAN REGISTERED VEHICLES” has been carried out by DARSHIL R PATHAK (140410107008), KARNA H DESAI (140410107010) and RUCHITA H PARMAR (140410107046) under my guidance in fulfillment of the project in Bachelor Of Engineering in Computer Engineering 8th semester of Gujarat Technological University, during the academic year 2017-2018.**

**Internal Guide:**

**Head of Department**

**Mr. Tejas S Bhatt**

**Mrs. B.J. Talati**

**Assistant Professor C.E. Department**

**Company Profile**

# Company Name : PORELAB SCIENTIFIC PRIVATE LIMITED .

# Introduction : Porelab Scientific Private Limited is a Private incorporated on 19 August 2015. It is classified as Non-govt company and is registered at Registrar of Companies, Ahmedabad. Its authorized share capital is Rs. 100,000 and its paid up capital is Rs. 100,000 . It is inolved in Manufacture of domestic appliances, n.e.c. Directors of Porelab Scientific Private Limited are Richa Kartik Gandhi, Arttasharan Dey and Swastisharan Arttasharan Dey. Porelab Scientific Private Limited's Corporate Identification Number is (CIN) U29300GJ2015PTC084265 and its registration number is 84265.Its Email address is SWASTIDEY1@GMAIL.COM and its registered address is A-12 JAIJALARAM NAGAR GOTRI ROAD VADODARA Vadodara GJ 390021 IN .

# Company Address : A-12 JAIJALARAM NAGAR GOTRI ROAD VADODARA Vadodara GJ 390021 IN .

# Service Offering : Research & Development .

# Technology : Research & Development

# Production Based



**93% Unique**

**Total chars , 284 words, 17 unique sentence(s).**

[**Essay Writing Service**](http://plagiarisma.net/editing.php) **- *Paper writing service you can trust. Your assignment is our priority! Papers ready in 3 hours! Proﬁcient writing: top academic writers at your service 24/7! Receive a premium level paper!***

[**STORE YOUR DOCUMENTS IN THE CLOUD**](https://upload.cat/) **- *1GB of private storage for free on our new ﬁle hosting!***

|  |  |  |
| --- | --- | --- |
| Results | Query | Domains (original links) |
| Unique | [App will be used to save the RTO related Documents like Driving License, PUC, RC Book, etc](http://www.google.com/search?lr&amp;hl=en&amp;filter=0&amp;as_qdr=all&amp;as_occt=any&amp;q=%22App%2Bwill%2Bbe%2Bused%2Bto%2Bsave%2Bthe%2BRTO%2Brelated%2BDocuments%2Blike%2BDriving%2BLicence%2C%2BPUC%2C%2BRC%2BBook%2C%2Betc%22&amp;nfpr=1) | - |
| Unique | [App will also have a record of the number of times a person is ﬁned for any rules broken, and the number of times a](http://www.google.com/search?lr&amp;hl=en&amp;filter=0&amp;as_qdr=all&amp;as_occt=any&amp;q=%22App%2Bwill%2Balso%2Bhave%2Ba%2Brecord%2Bof%2Bthe%2Bnumber%2Bof%2Btimes%2Ba%2Bperson%2Bis%2Bfined%2Bfor%2Bany%2Brules%2Bbroken%2C%2Band%2Bthe%2Bnumber%2Bof%2Btimes%2Ba%2Bvehicle%E2%80%99s%2BPUC%2Bis%2Bissued%22&amp;nfpr=1) [vehicle’s PUC is issued](http://www.google.com/search?lr&amp;hl=en&amp;filter=0&amp;as_qdr=all&amp;as_occt=any&amp;q=%22App%2Bwill%2Balso%2Bhave%2Ba%2Brecord%2Bof%2Bthe%2Bnumber%2Bof%2Btimes%2Ba%2Bperson%2Bis%2Bfined%2Bfor%2Bany%2Brules%2Bbroken%2C%2Band%2Bthe%2Bnumber%2Bof%2Btimes%2Ba%2Bvehicle%E2%80%99s%2BPUC%2Bis%2Bissued%22&amp;nfpr=1) | - |
| Unique | [Paying ﬁnes or Renewing of the Driving License can be done with app](http://www.google.com/search?lr&amp;hl=en&amp;filter=0&amp;as_qdr=all&amp;as_occt=any&amp;q=%22Paying%2Bfines%2Bor%2BRenewing%2Bof%2Bthe%2BDriving%2BLicence%2Bcan%2Bbe%2Bdone%2Bwith%2Bapp%22&amp;nfpr=1) | - |

App will be used to save the RTO related Documents like Driving License, PUC, RC Book, etc. App will also have a record of the number of times a person is ﬁned for any rules broken, and the number of times a vehicle’s PUC is issued. Paying ﬁnes or Renewing of the Driving License can be done with app.

**UNDERTAKING ABOUT ORIGINALITY OF WORK**

We hereby certify that we are the sole authors of this IDP/UDP Project report and that neither any part of this IDP/UDP Project report nor the whole of the IDP/UDP Project report has been submitted for a degree to any other University or Institution.

We certify that, to the best of our knowledge, the current IDP/UDP Project does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in our IDP/UDP Project report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, we certify that we have obtained a written permission from the copyright owner(s) to include such material(s) in the current IDP/UDP Project report and have included copies of such copyright clearances to our appendix.

We have checked write up of the present IDP/UDP Project report using anti-plagiarism database and it is in allowable limit. Even though later on in case of any complaint pertaining of plagiarism, we are sole responsible for the same and we understand that as per UGC norms, University can even revoke BE degree conferred to the student submitting this IDP/UDP Project report incase if it is found to be plagiarized.

Date:

Place:

Name:- Enrolment Number:- Signature of Student:-

1) Darshil Pathak 140410107008

2) Karna Desai 140410107010

3) Ruchita Parmar 140410107046

**ACKNOWLEDGEMENT**

Nothing can be done perfectly without a proper guidance. No task can be fulfilled in a land of selfish takers. We are highly indebted to all the people involved in the development of the project. Without their constant motivation, it would have been next to impossible to carve this project to completion. We would like to thank Prof. BIJAL J. TALATI (H.O.D C.E. Department) who gave us the opportunity to grow as software professionals and helped us to acquire such a valuable experience. Most importantly we are thoroughly indebted to Mr. TEJAS P. BHATT, our internal guide who enthusiastically guided us and painstakingly followed up on our progress. Moreover, we wish to acknowledge his unwavering support through the tenure of a degree program. But most of all we would like to thank him for believing in our abilities .Finally, we would also like to thank other faculty members who provided us with Moral support during the course of this project .This has been a priceless learning experience for us and immensely helped us develop outlook regarding the making of a software system and has widened our horizon to a greater extent.

**CONTENTS**

**1. Introduction..................................................................................................1**

1.1 Problem Summary.............................................................................1

1.2 Aim and Objective.............................................................................1

1.3 Sodh Yatra.........................................................................................2

**2. Requirement Gathering...............................................................................2**

2.1 Functional Requirement.....................................................................2

2.2 Non Functional Requirement.............................................................2

2.3 Hardware Requirement......................................................................3

2.4 Software Requirement.......................................................................3

**3. Feasibility Study..........................................................................................3**

3.1 Technical Study.................................................................................3

3.2 Time Feasibility.................................................................................3

3.3 Economical Feasibility.......................................................................4

**4. System Design...............................................................................................5**

4.1 System Model....................................................................................5

4.2 System Architecture..........................................................................6

**5. System Diagram...........................................................................................7**

5.1 Class Diagram...................................................................................7

5.2 UseCase Diagram..............................................................................8

5.3 Activity Diagram..............................................................................11

5.4 State Diagram..................................................................................14

5.5 Sequence Diagram...........................................................................15

5.6 DFD Diagram..................................................................................16

**6. Database Design.........................................................................................17**

6.1 ER Diagram.....................................................................................17

6.2 Data Dictionary................................................................................17

**7. Software Description................................................................................19**

7.1 Android Studio.................................................................................19

7.2 Android SDK...................................................................................20

7.3 FireBase Database............................................................................20

**8. Hardware Component Description...........................................................21**

**9. Implementation……………………………………………………………23**

**10. Testing.........................................................................................................27**

**11. Conclusion..................................................................................................30**

**Canvas.........................................................................................................31**

9.1 AEIOU Canvas...................................................................................31

9.2 Empathy Mapping Canvas.................................................................32

9.3 Ideation Canvas..................................................................................33

9.4 Product Development Canvas............................................................34

**1. Introduction**

**1.1** **Problem Summary**

The main problem with traditional document application is that they all provide same simple traditional features .These apps have limited features. To solve this problem, we are making an application that has unique features that provide user with more facility. To solve this problem, we are making Location based maps, Renewal options, fine payments and other exciting features . As the name suggests this application has details of your registered vehicle, your driving license and other RTO related documents. Also, this application has other modules for reminder, fine payment, etc that are unique then traditional applications.

**1.2 Aim and Objective**

The aim of our system is designing an android application that does following things:

* It is an Android Application to provide documents like driving license, PUC, RC Book, etc whenever required.
* This app will also have a record of the number of times a person is fined for any rules broken, and the number of times a vehicle’s PUC is issued.
* Paying fines or Renewing of the Driving License can be done with this app.
* App will offer to set a reminder for renewal of PUC.
* It will also allow user to post questions on the forum which would be answered only by an authenticated Officials.
* It will allow user to get information regarding Traffic Rules.

**1.3 Sodh Yatra**

* At the outset, we gathered some definitions on which we thought of making the project. We pondered on and discussed the definitions with our project Guide, Ass. Professor Tejas Bhatt, and finally selected IRV-Indian Registered Vehicle which we thought would be interesting as well as0 new project for us.
  + We thought of this app because of the incidents happening with the common people and day to day commuters.
* One day while going to my college on my bike and I was caught by the traffic police. He asked me for my driving license. I had a driving license but I forgot my wallet at home. So was not able to justify myself and had to pay fine.
  + We aim at solving such problems by developing an android application. The major use of this app is that it consists of various features that helps common men to manage his documents with ease.

**2. Requirement Gathering**

This part mentions the requirements set of the project. For the project to be successful and be able to evaluate if it is complete, it was needed to compute a set of functional and non-functional requirements. The functional requirements are how system will work behave, where the non-functional is how the whole system operates.

**2.1 Functional Requirement**

When police official scans the barcode of the user, the details of the user should get displayed on the officers device.

- The fine generated by the official should be displayed on the users application screen and the payment could be done digitally.

-While making an appointment in RTO for any purpose . Application should ask for all Required information.

-Application must retrieve all the documents required by a user from the database.

**2.2 Non Functional Requirement**

-Application should show all the rules and regulation for road transport.

-Backup - System must take regular backup of database.

-Robustness-few errors.

-QR code Scanner

**2.3 Hardware Requirement**

-Android Device

-Camera

-RAM 512 MB

**2.4 Software Requirement**

-Android Version 5.0

-Firebase Database

**3. Feasibility Study**

This study helps in taking the decision that one should design a particular project or not and till what extend or limits the project should be stretched.

**3.1 Technical Study**

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel.

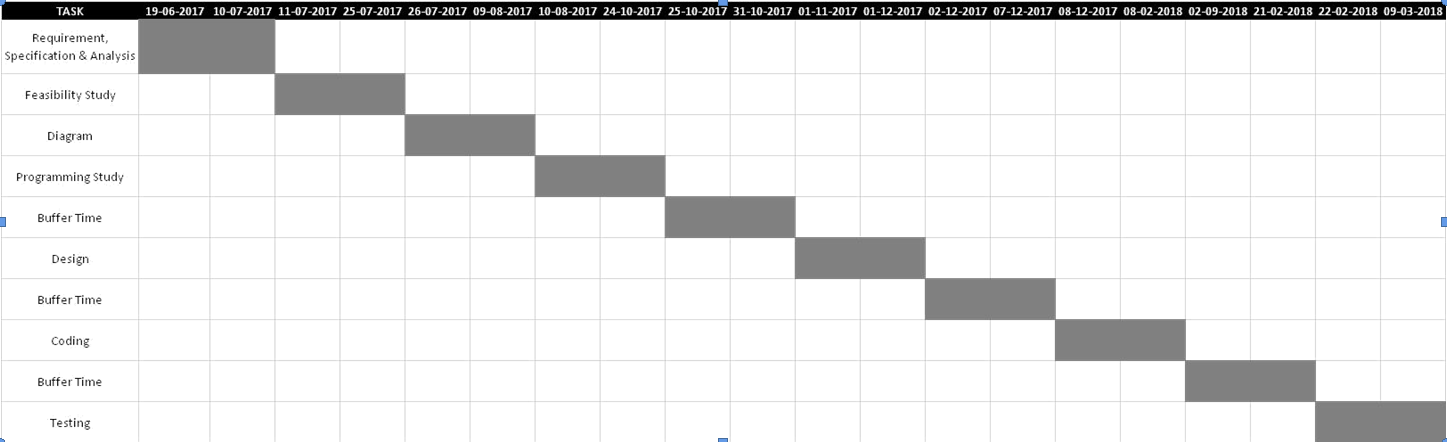
Technical feasibility of proposed project refer to the software and hardware requirements.

The project is developed using Android Studio, Firebase is used for Database.

The proposed project can be implemented on any Android Device.

The application would be using few API like Traffic API and Service API which are provided by google.

**3.2 Time Feasibility(Timeline Chart)**

****

**3.3 Economic Feasibility**

Economic feasibility determines whether there are sufficient benefits in creating to make the cost acceptable, or is the cost of the system too high.

The software’s used to develop the proposed system are cost efficient.

It is assumed that the user already possesses Android Device.

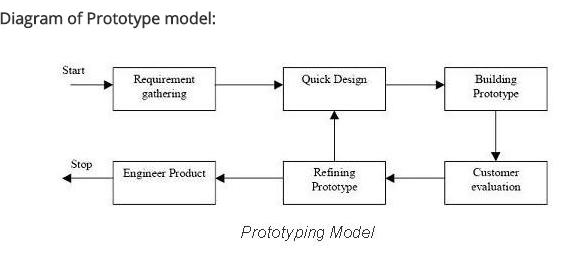
Moreover the payment gateway used is paytm which is free of cost.

But the future scope of application is high, so more payment gateways and paid version of firebase would be used which would cost high around Rs 50,000/-.

**4. System Design**

**4.1 System Model**

We are planning our project according to “Prototyping Model” in Software Engineering.

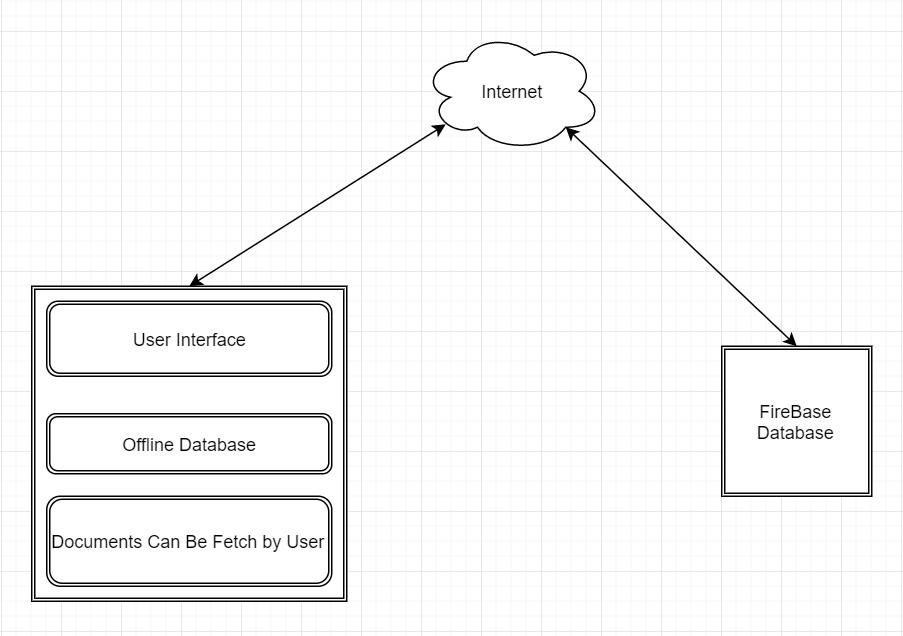


The Prototyping is a Software Development Model in which a prototype is built, tested and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can be developed.

Prototyping Model involves following steps:

1. The new system requirements are defined in as much details as possible.
2. A preliminary design is created for the new system.
3. A first prototype of new system is constructed from preliminary design.
4. The users thoroughly evaluate the first prototype, noting its strengths and weakness, what needs to be added, and what should be removed.
5. The first prototype is modified, based on the comments supplied by the users.
6. The second prototype is evaluated in the same manner as was the first prototype.
7. The preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired.
8. The final system is constructed, based on the final product.
9. The System is thoroughly evaluated and tested.

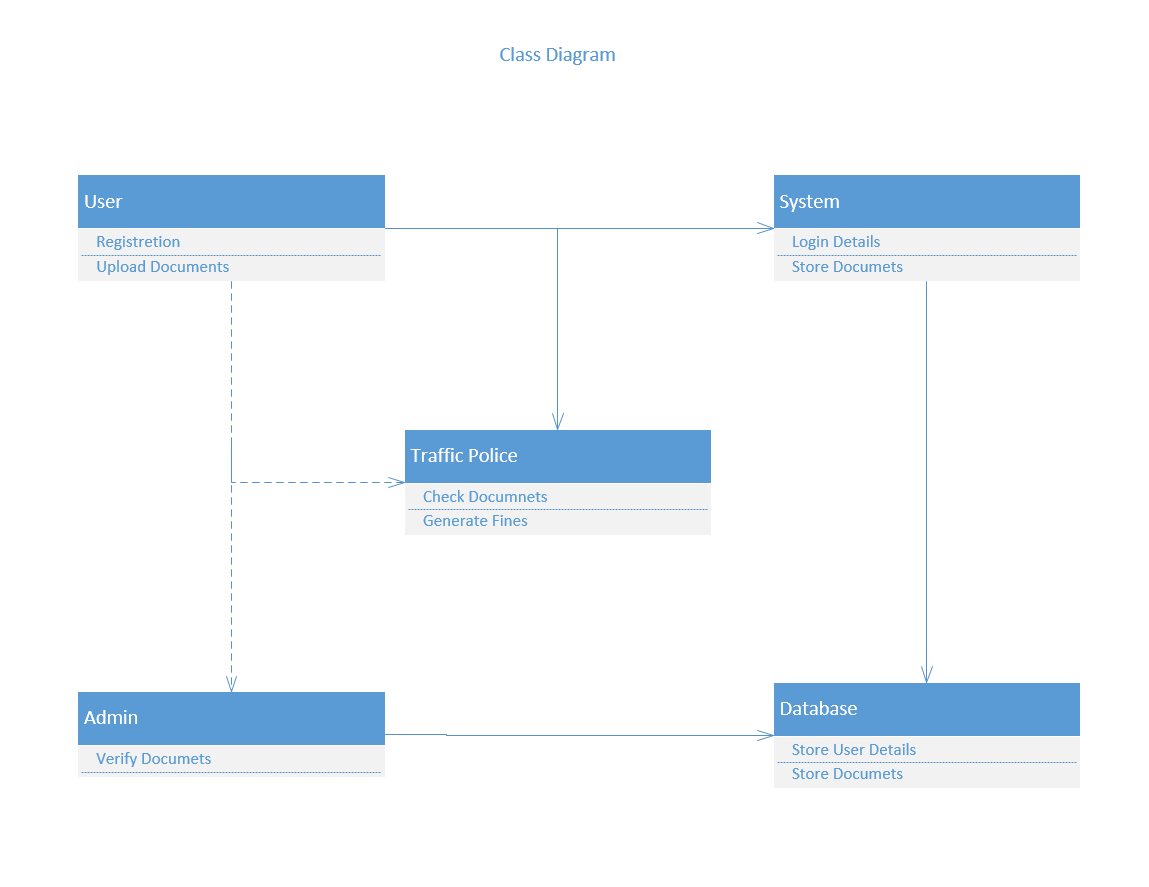
**4.2 System Architecture**

****

**5. System Diagram**

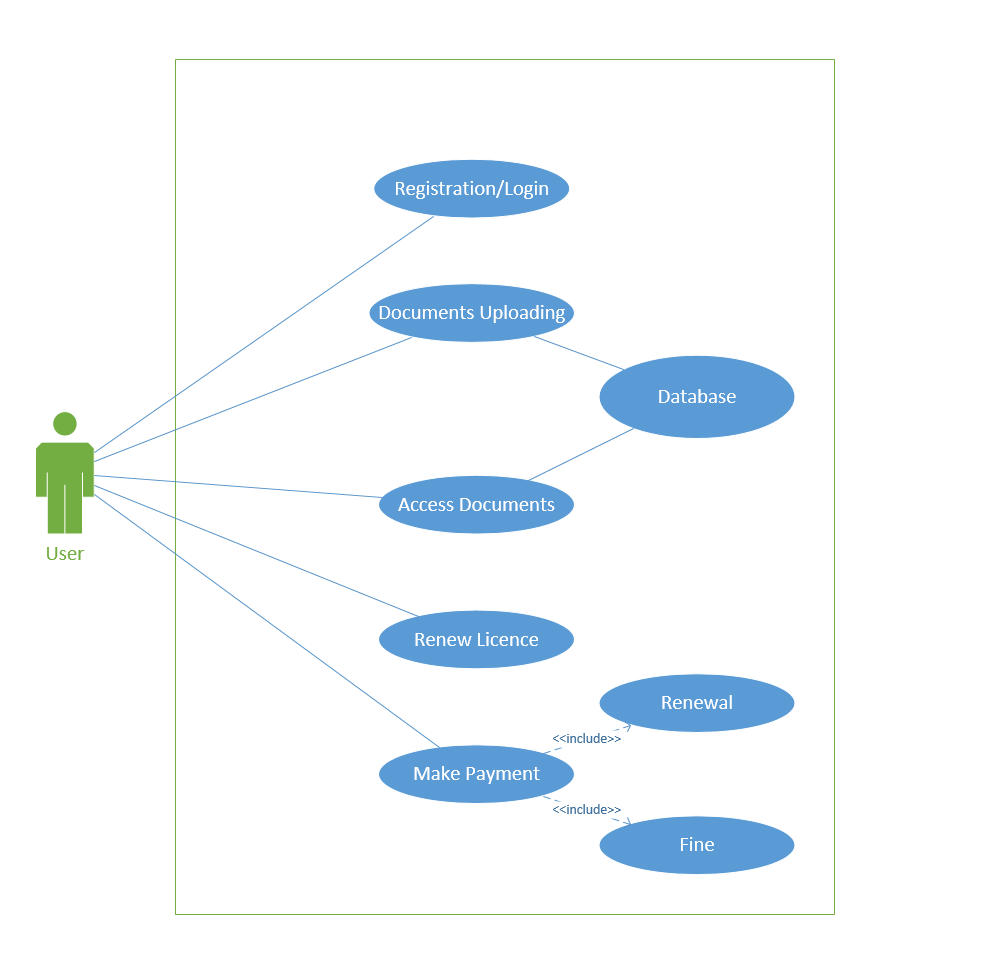
**5.1 Class Diagram**

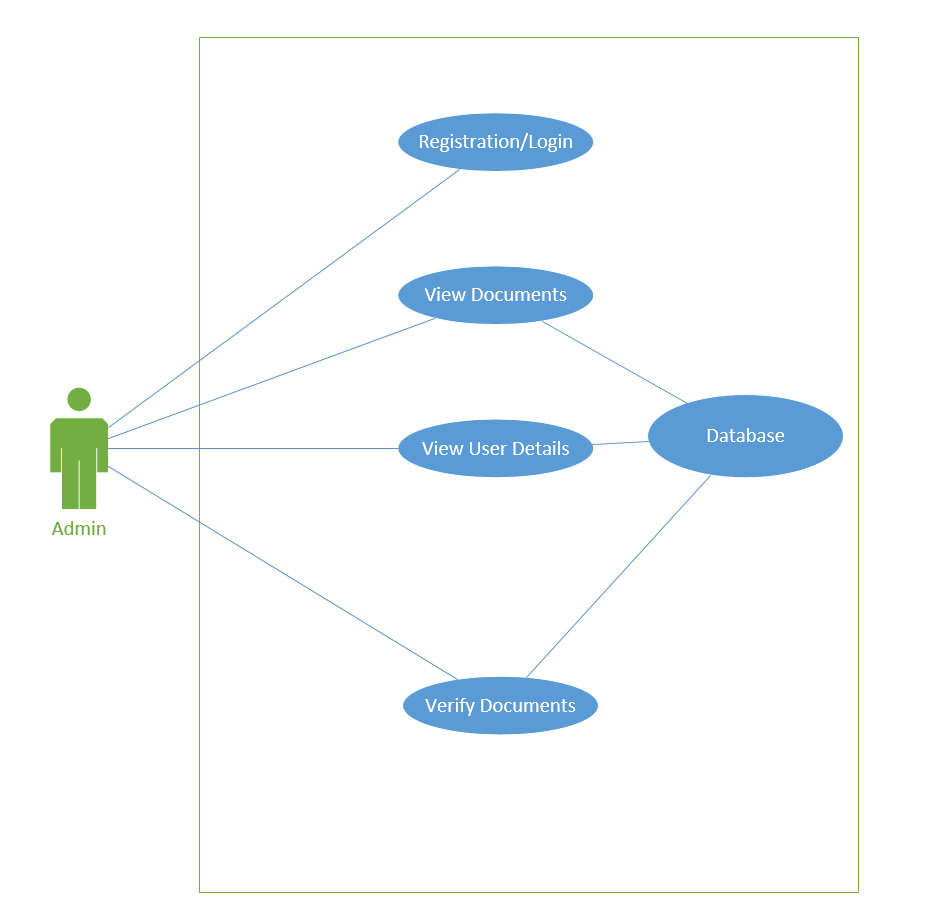
Class diagrams are used to capture the static view of the system. They represent how to put various objects together. It gives an overview of the system, in which various classes and relationship among these classes is represented.

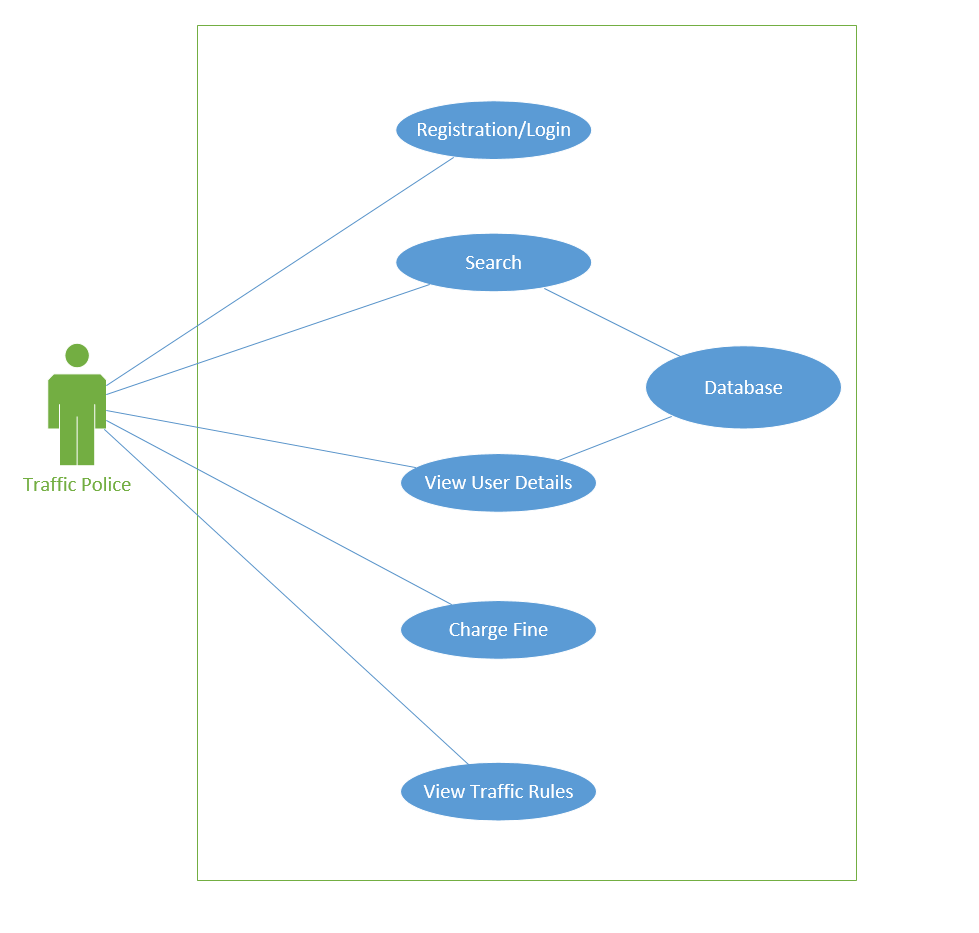
****

**5.2 UseCase Diagram**

The Use case diagram describes about the several activities performed at the office by different users such as Admin, Users, Traffic Police etc.

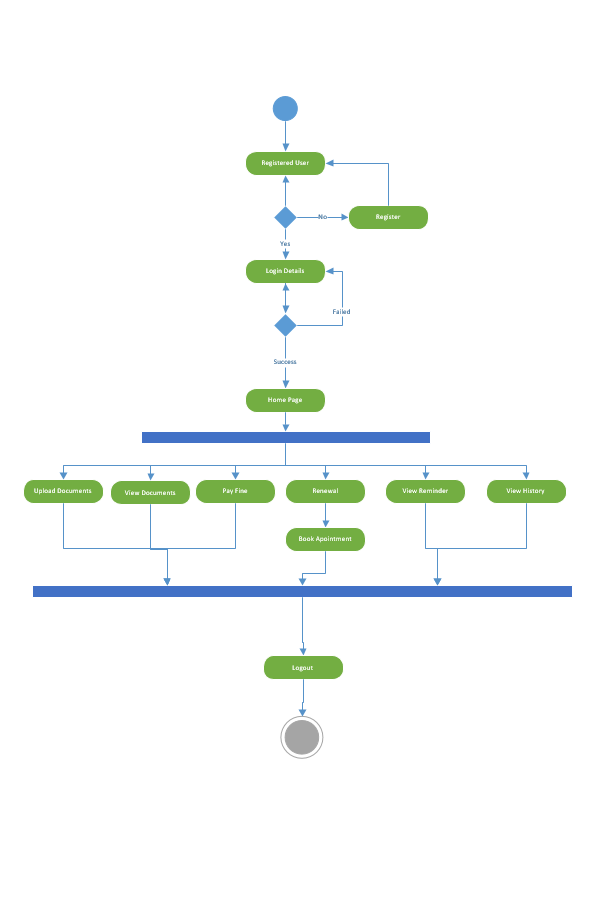
****

****

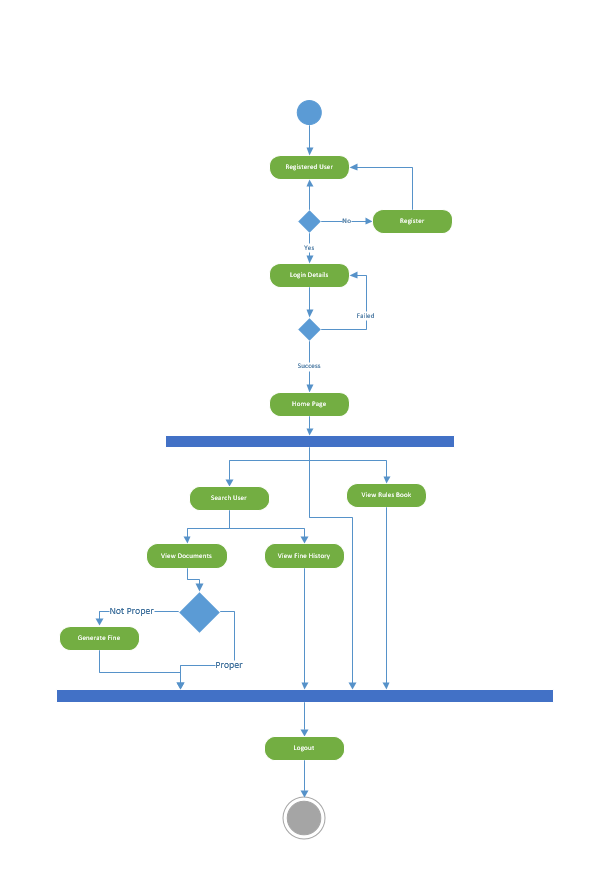
****

**5.3 Activity Diagram (User)**

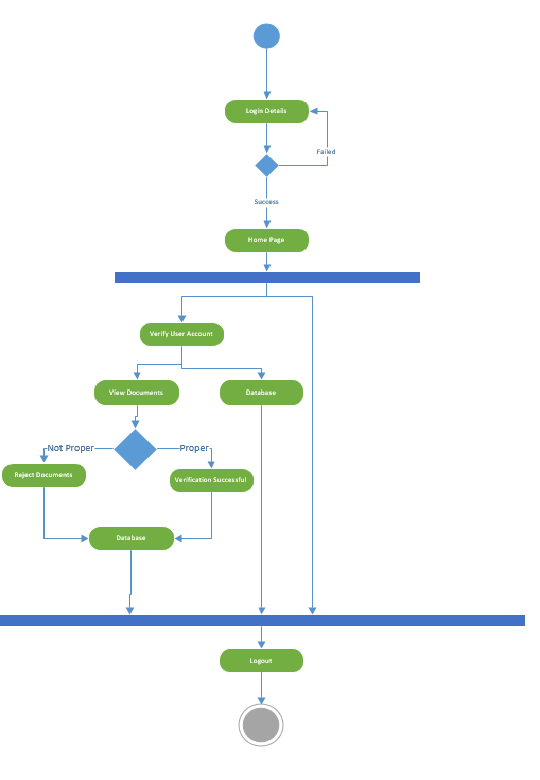
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

****

**(Traffic Police)**

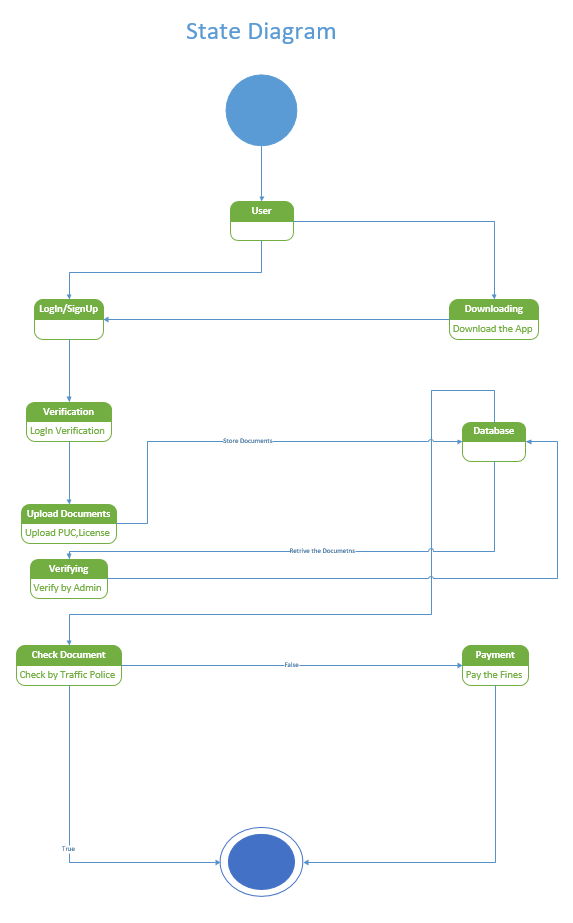
****

**(Admin)**

****

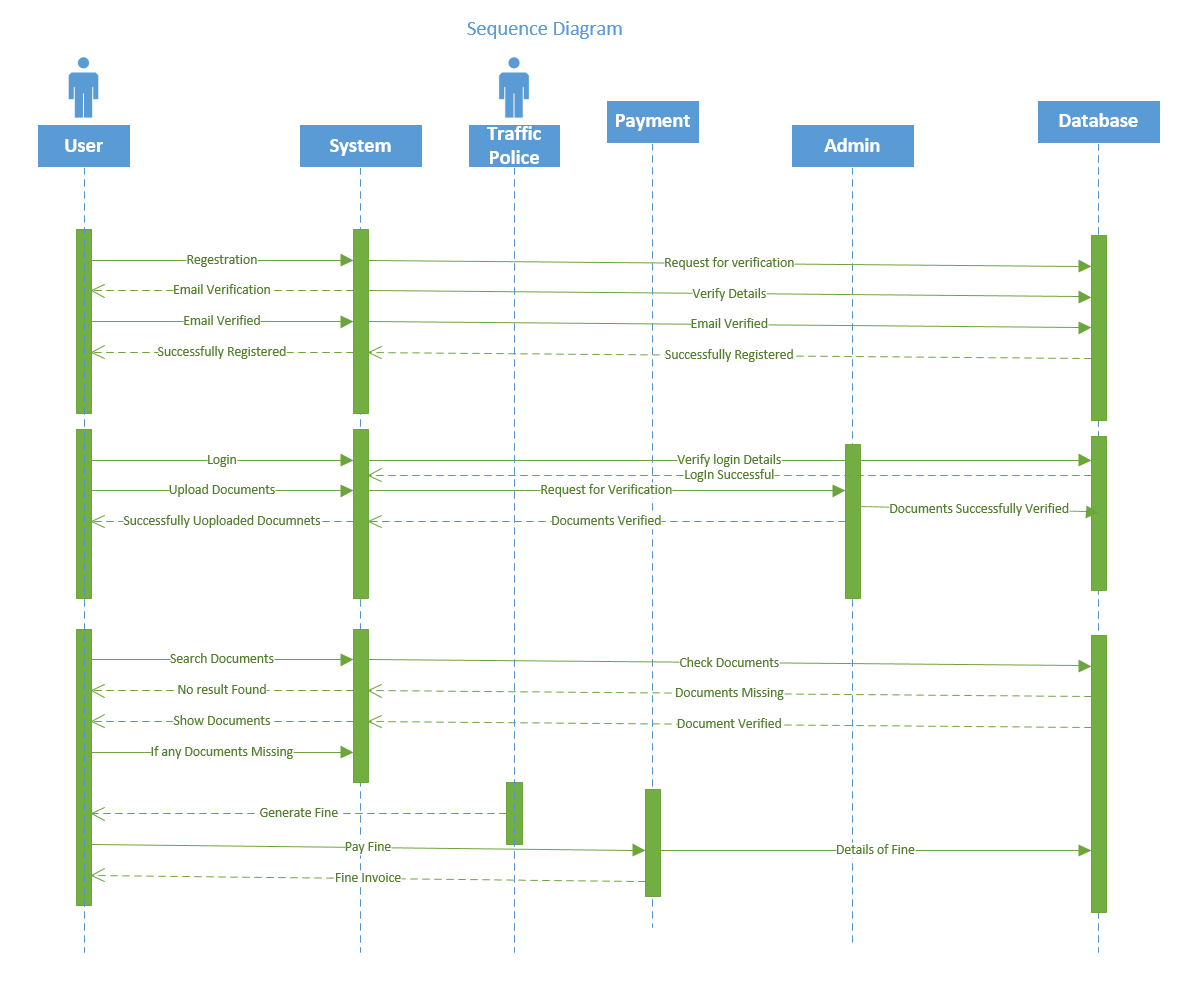
**5.4 State Diagram**

A state diagram is a type of diagram used in computer science and related fields to describe the behaviour of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction .

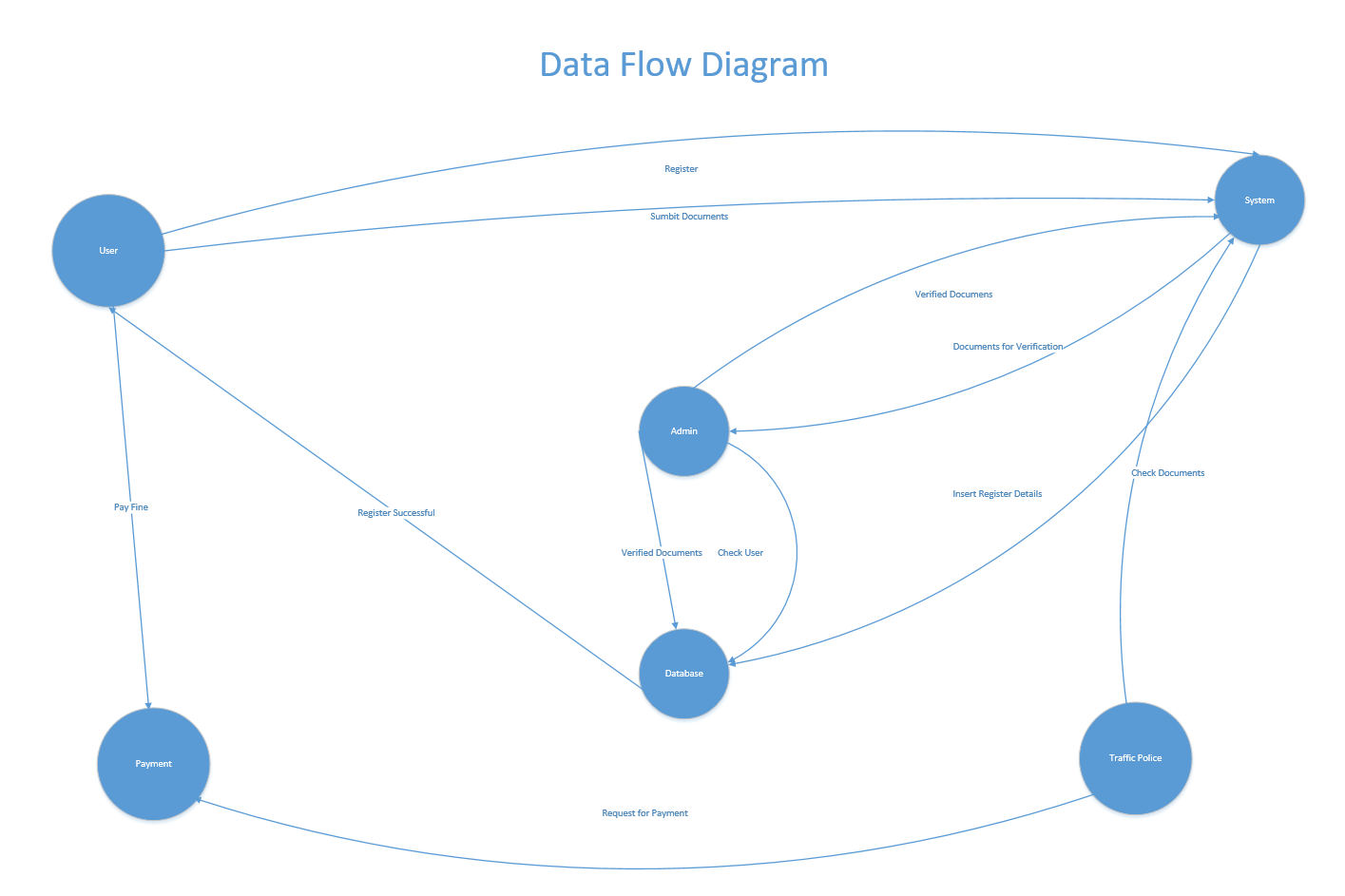
****

**5.5 Sequence Diagram**

The sequence diagram describes the chain of events in series that are done while executing various functions of the software.

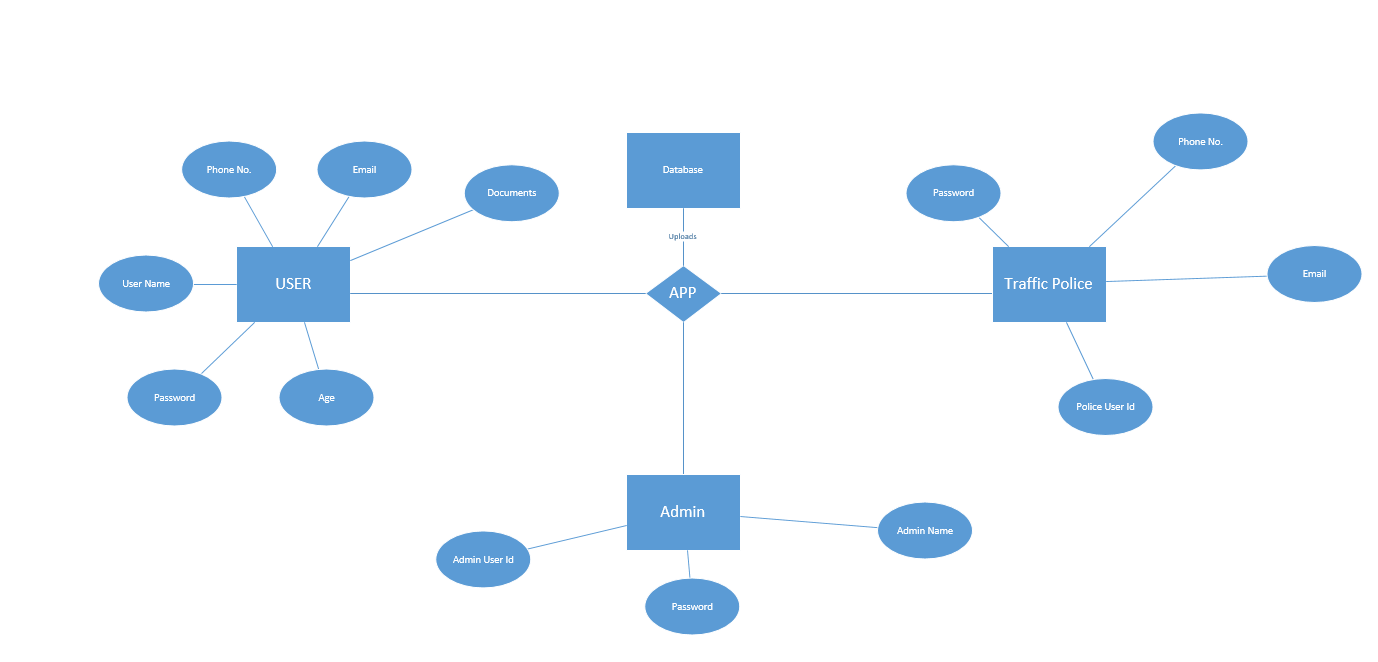
****

**5.6 DFD**



**6. Database Design**

**6.1 E-R Diagram**

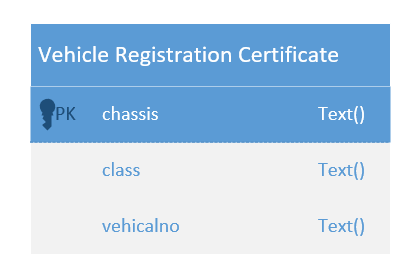
****

**6.2 Data Dictionary**

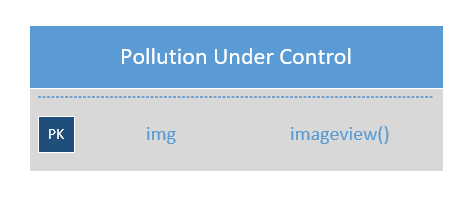
**License Database**

****

**R C Book Database**

****

**PUC Database**

****

**7. Software Description**

**7.1 Android Studio**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 2.3.3, released in June 2017. Next major update, version 3.0, is in preview stage as of September 2017.

New features are expected to be rolled out with each release of Android Studio. The following features are provided in the current stable version:[[13]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-13)[[14]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-14)

* [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support
* Android-specific [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes
* [Lint](https://en.wikipedia.org/wiki/Lint_(software)) tools to catch performance, usability, version compatibility and other problems
* [ProGuard](https://en.wikipedia.org/wiki/ProGuard_(software)) integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations[[15]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-15)
* Support for building [Android Wear](https://en.wikipedia.org/wiki/Android_Wear) apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine[[16]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-16)
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.

**7.2 Android SDK**

The Android software development kit (SDK) includes a comprehensive set of development tools . These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution, Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Android applications.

Until around the end of 2014, the officially supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. As of 2015, Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others, but Google made it clear that ADT was officially deprecated since the end of 2015 to focus on Android Studio as the official Android IDE. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

**7.3 Firebase Database**

The Firebase Realtime Database is a cloud-hosted database.

Data is stored as JSON and synchronized in realtime to every connected client.

Data is synced across all clients in realtime, and remains available when your app goes offline.

When you build cross-platform apps with our IOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

**How does it work?**

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access your data and then [structure it accordingly](https://firebase.google.com/docs/database/web/structure-data).

**8. Hardware then Component Description**

**Android Application**

Android apps are written in the Java programming language. The Android SDK tools compile your code along with any data and resource files into an APK, an *Android package*, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

Each Android app lives in its own security sandbox, protected by the following Android security features:

* The Android operating system is a multi-user Linux system in which each app is a different user.
* By default, the system assigns each app a unique Linux user ID (the ID is used only by the system and is unknown to the app). The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.
* Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps.
* By default, every app runs in its own Linux process. The Android system starts the process when any of the app's components need to be executed, and then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

The Android system implements the *principle of least privilege*. That is, each app, by default, has access only to the components that it requires to do its work and no more. This creates a very secure environment in which an app cannot access parts of the system for which it is not given permission. However, there are ways for an app to share data with other apps and for an app to access system services:

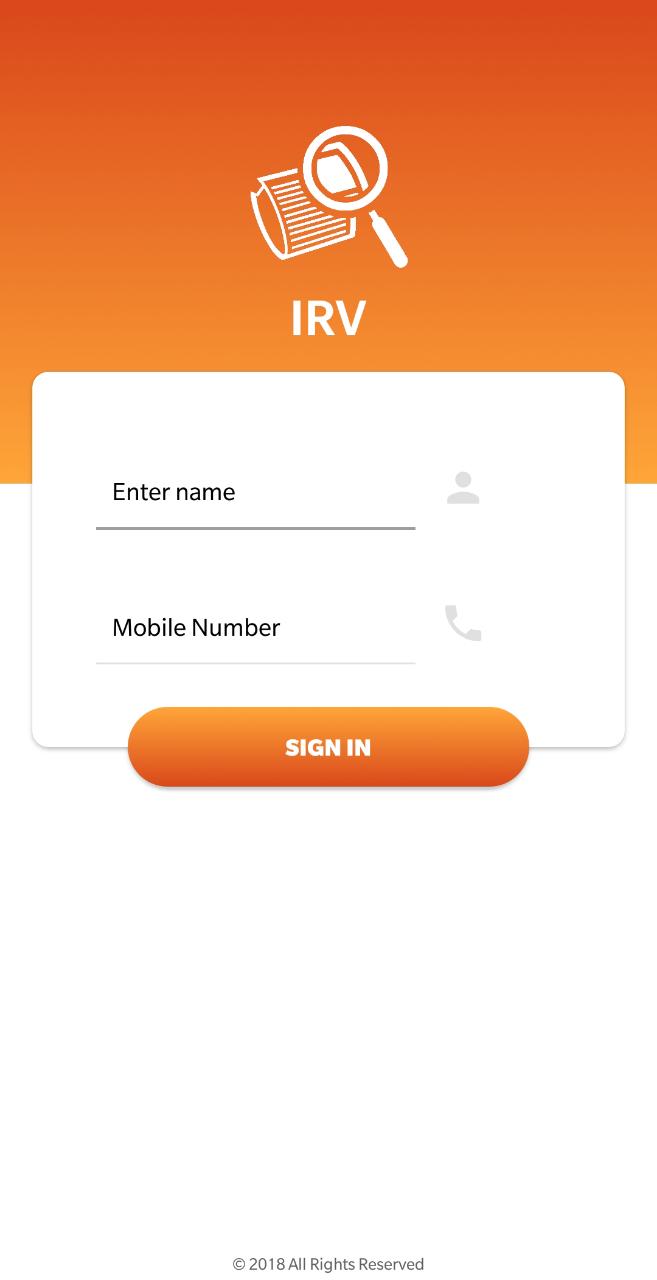
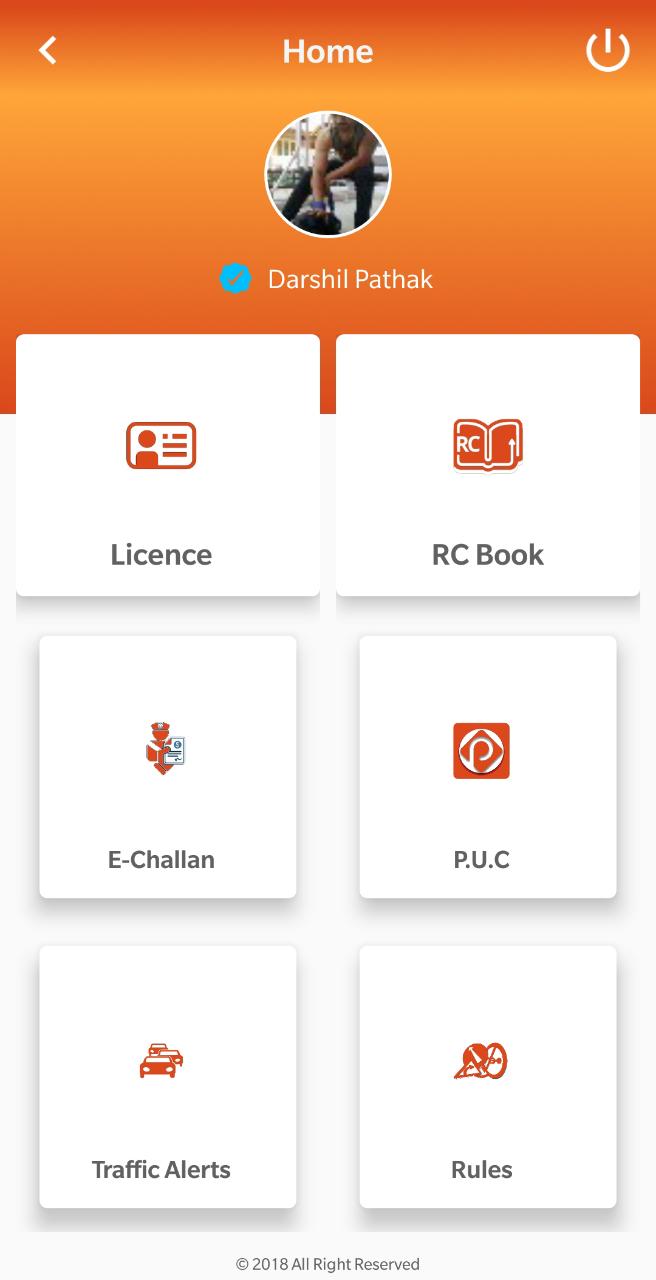
* It's possible to arrange for two apps to share the same Linux user ID, in which case they are able to access each other's files. To conserve system resources, apps with the same user ID can also arrange to run in the same Linux process and share the same VM. The apps must also be signed with the same certificate.
* An app can request permission to access device data such as the user's contacts, SMS messages, the mountable storage (SD card), camera, and Bluetooth. The user has to explicitly grant these permissions. For more information, see [Working with System Permissions](https://developer.android.com/training/permissions/index.html).

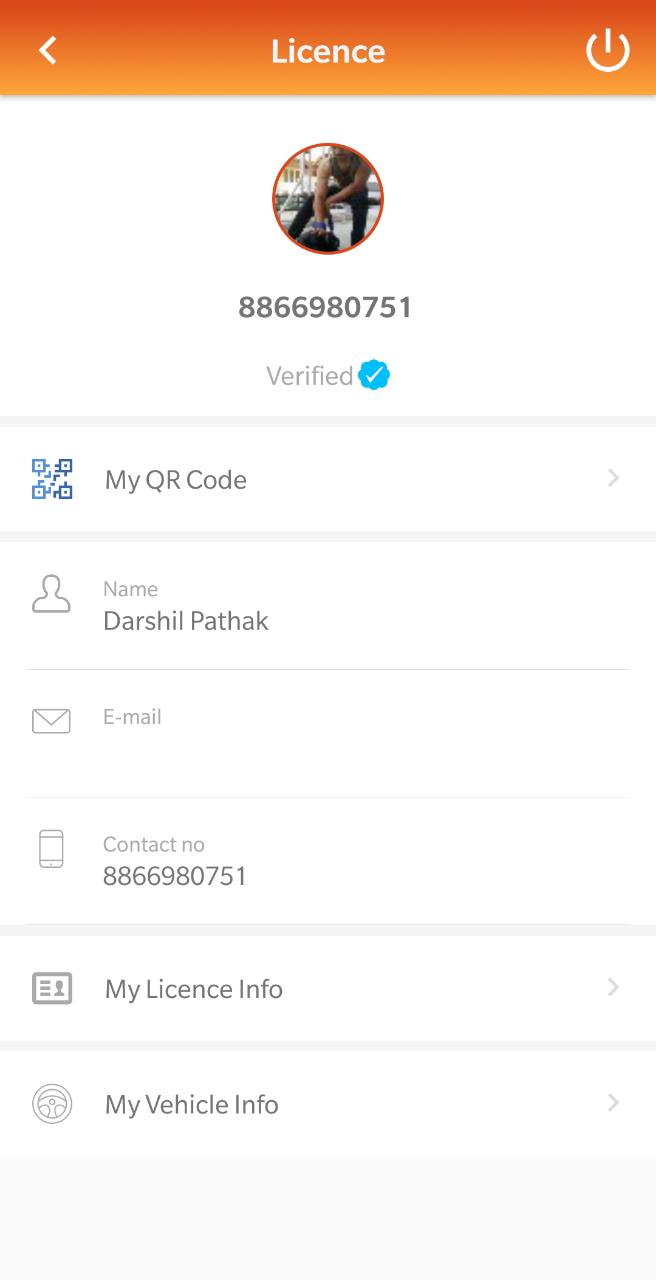
The rest of this document introduces the following concepts:

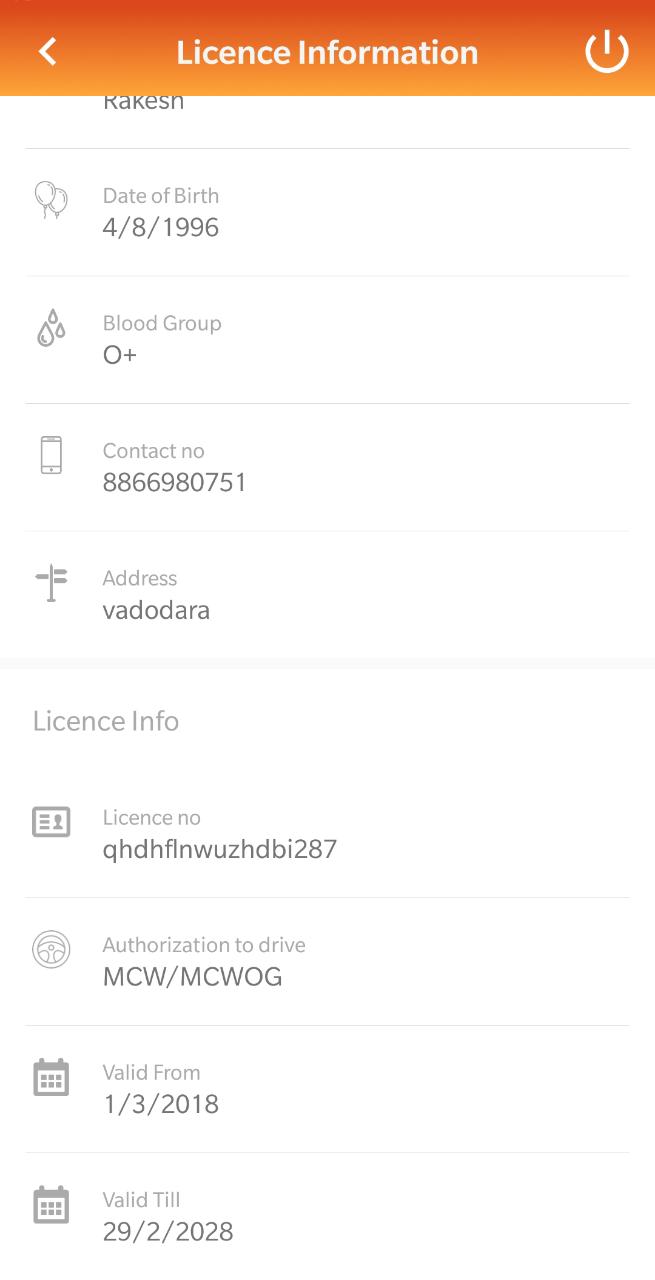
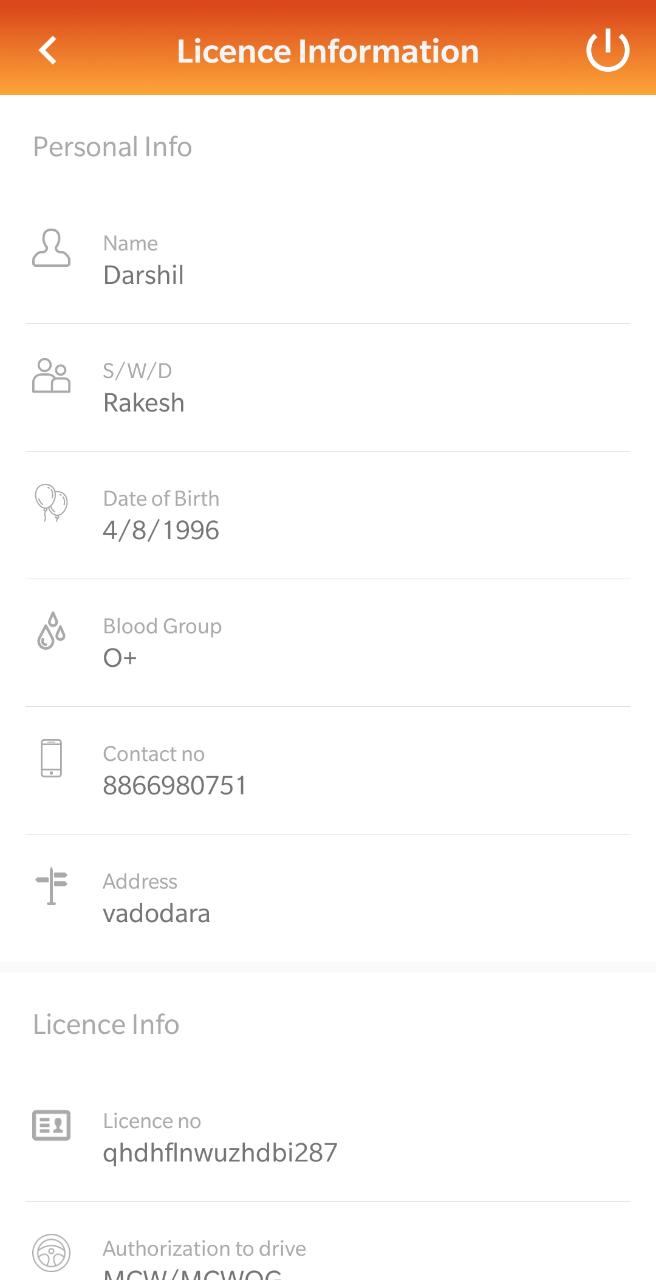
* The core framework components that define your app.
* The manifest file in which you declare the components and the required device features for your app.
* Resources that are separate from the app code and that allow your app to gracefully optimize its behavior for a variety of device configurations.

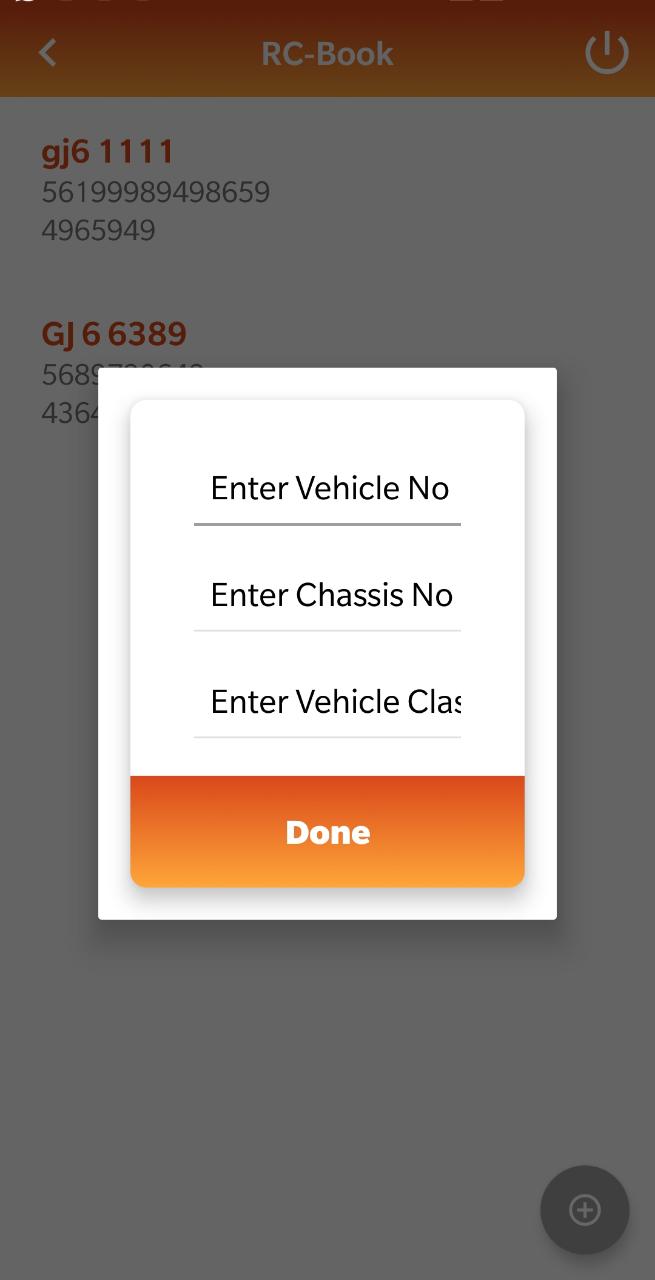
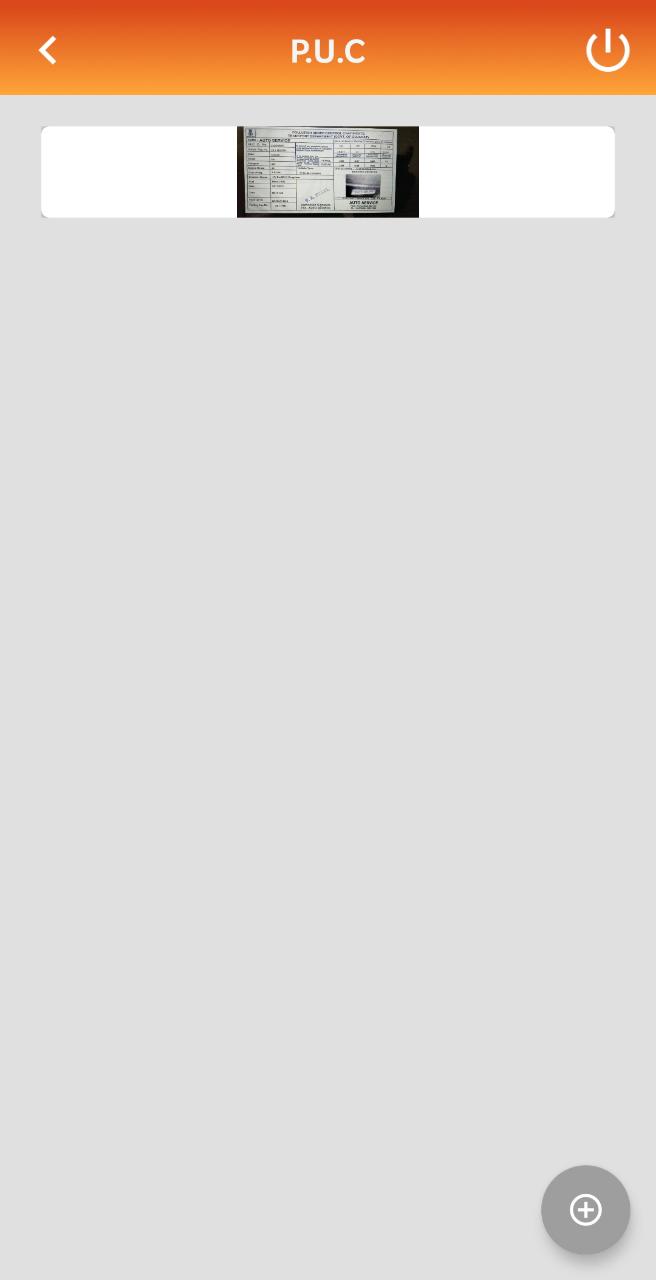
**9. Implementation**

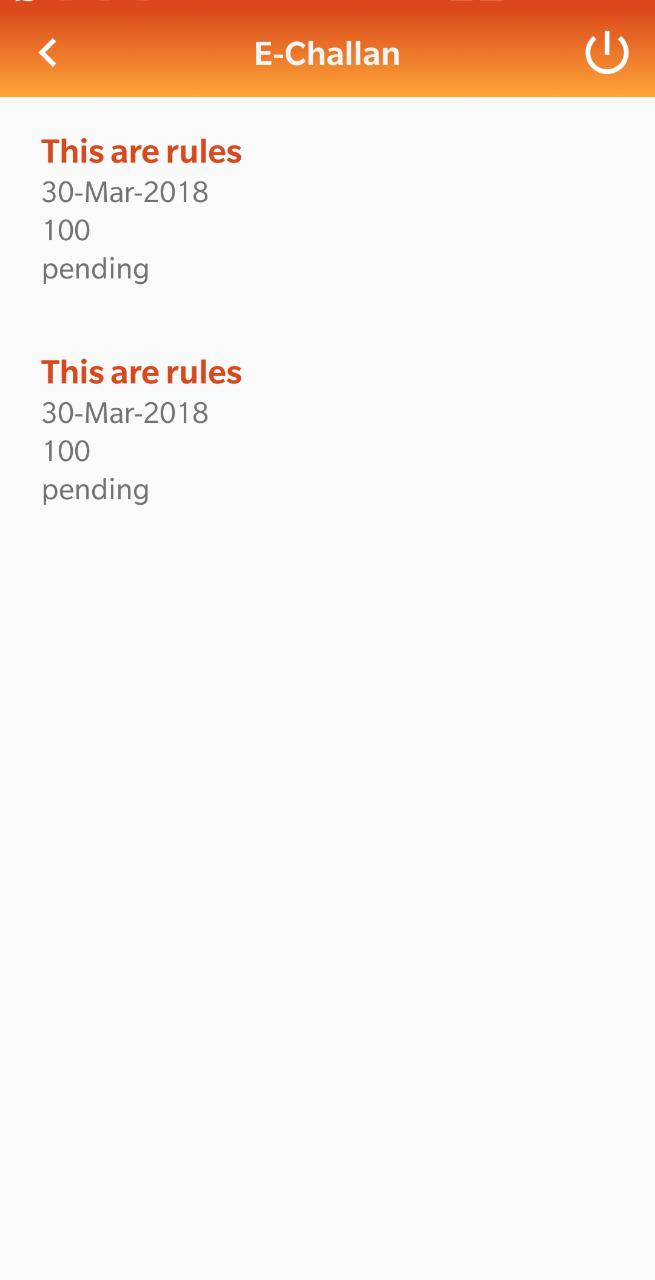
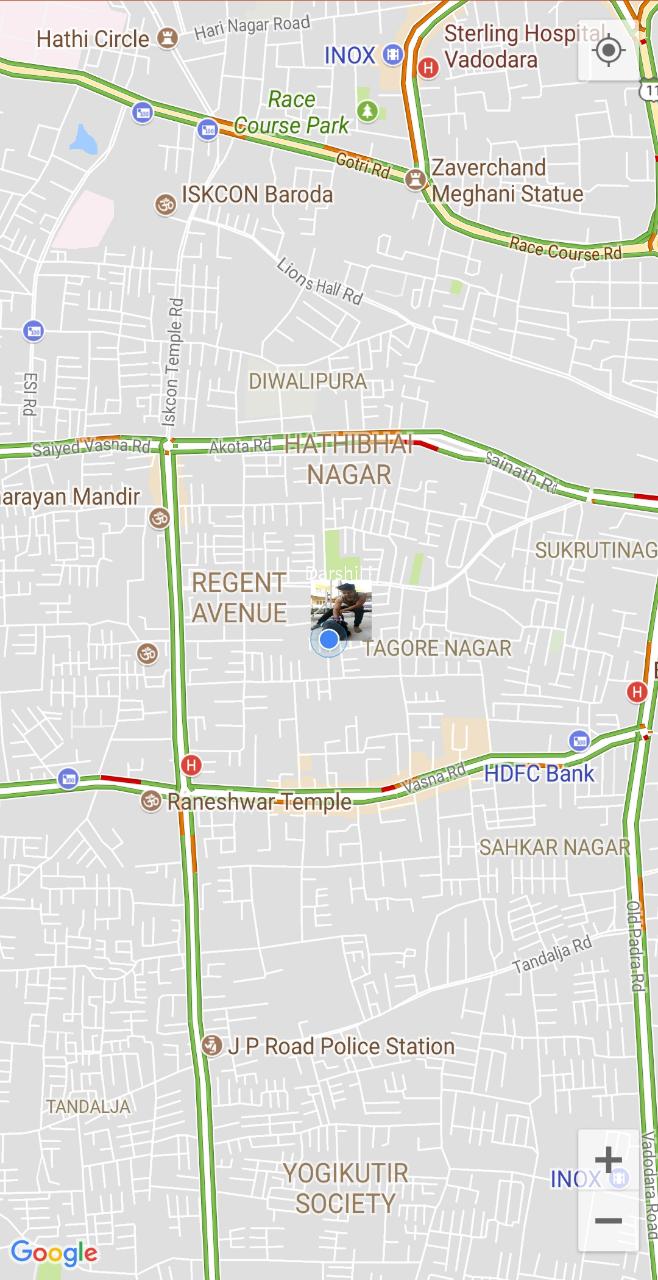
**User side Implementation**

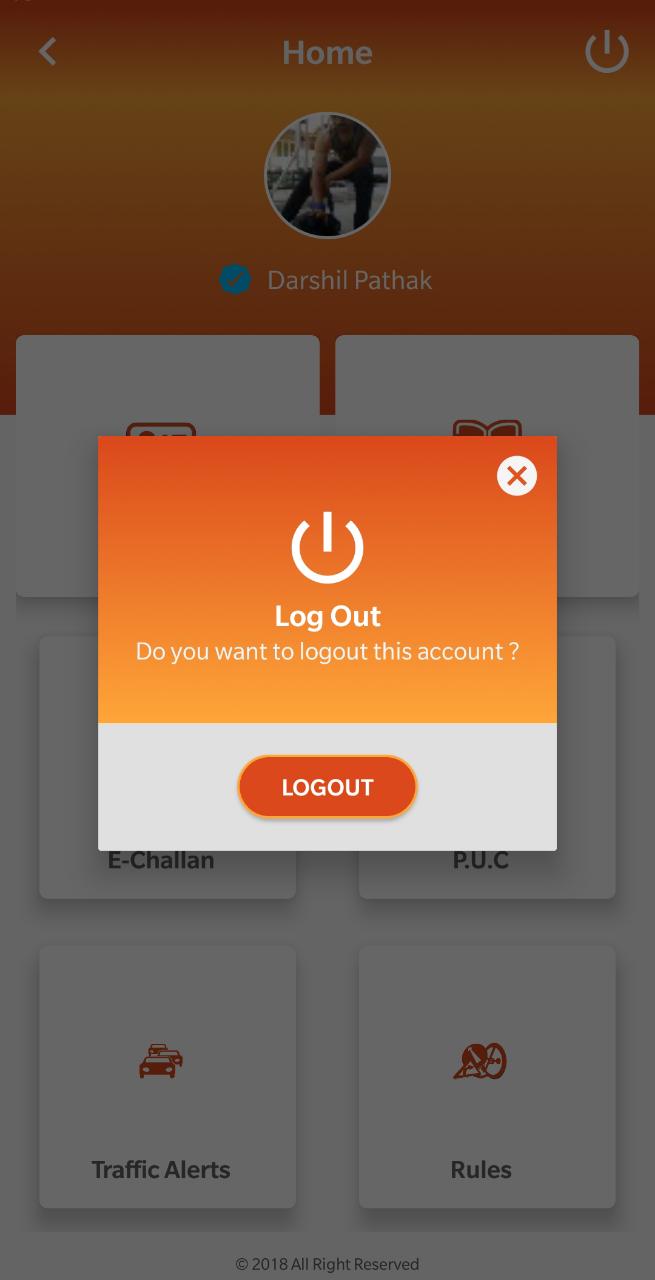
** **

** **

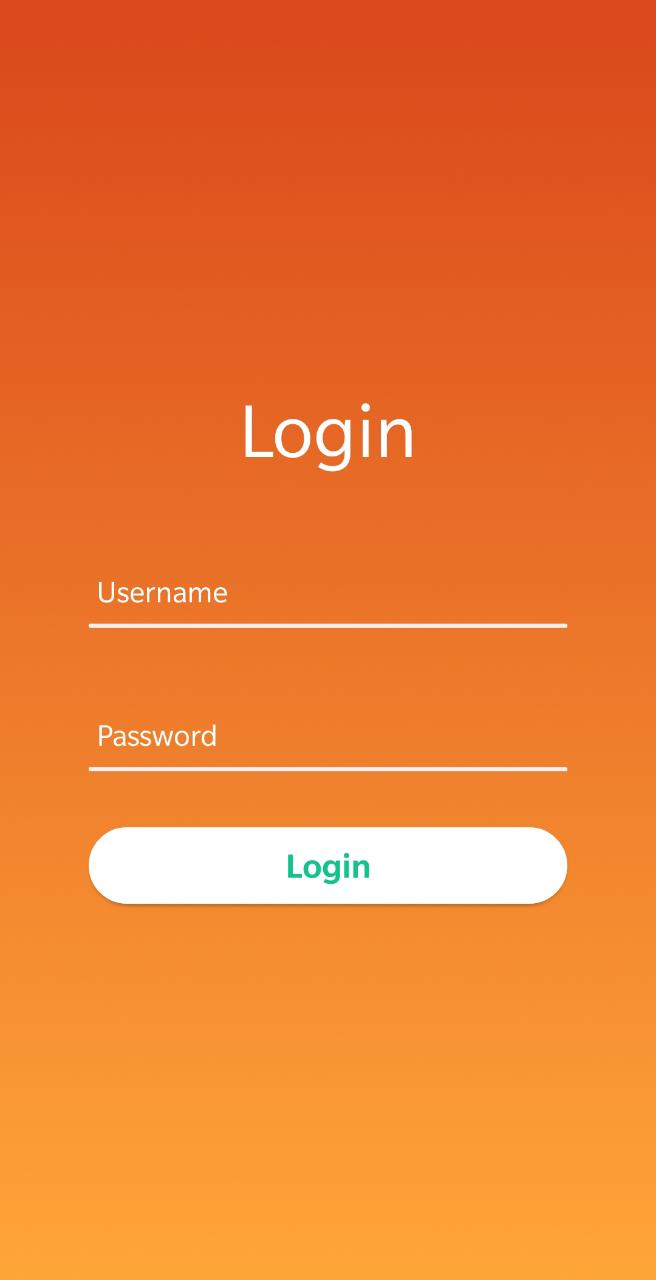
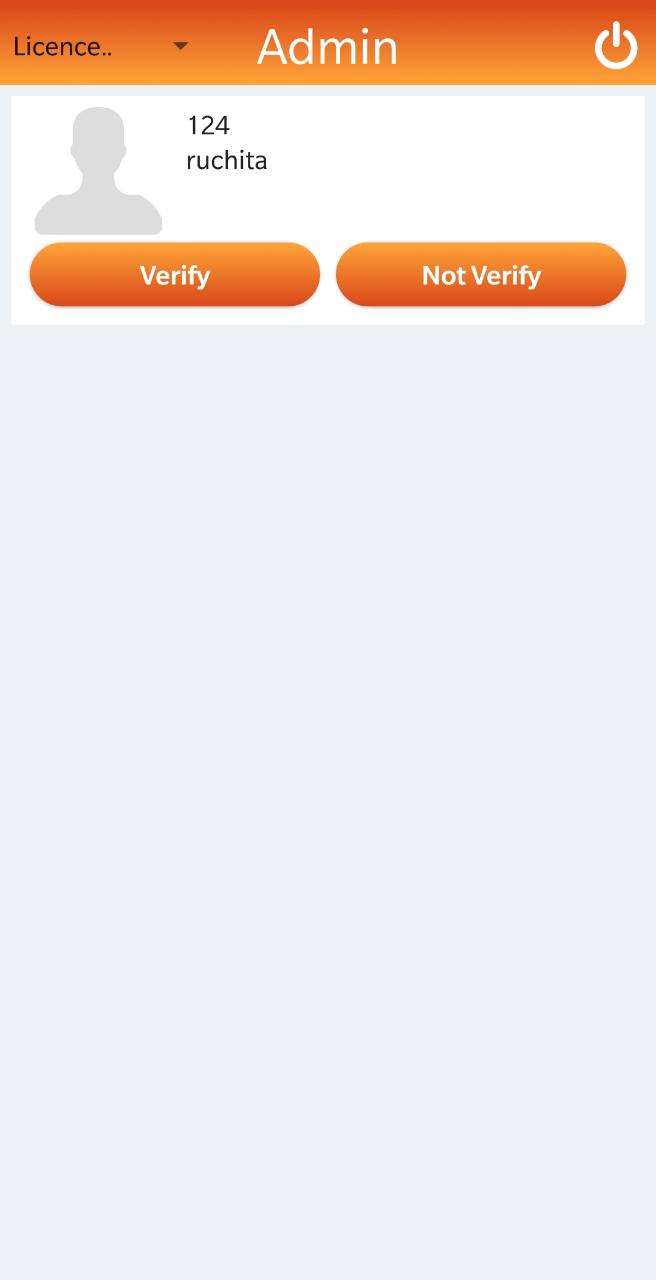


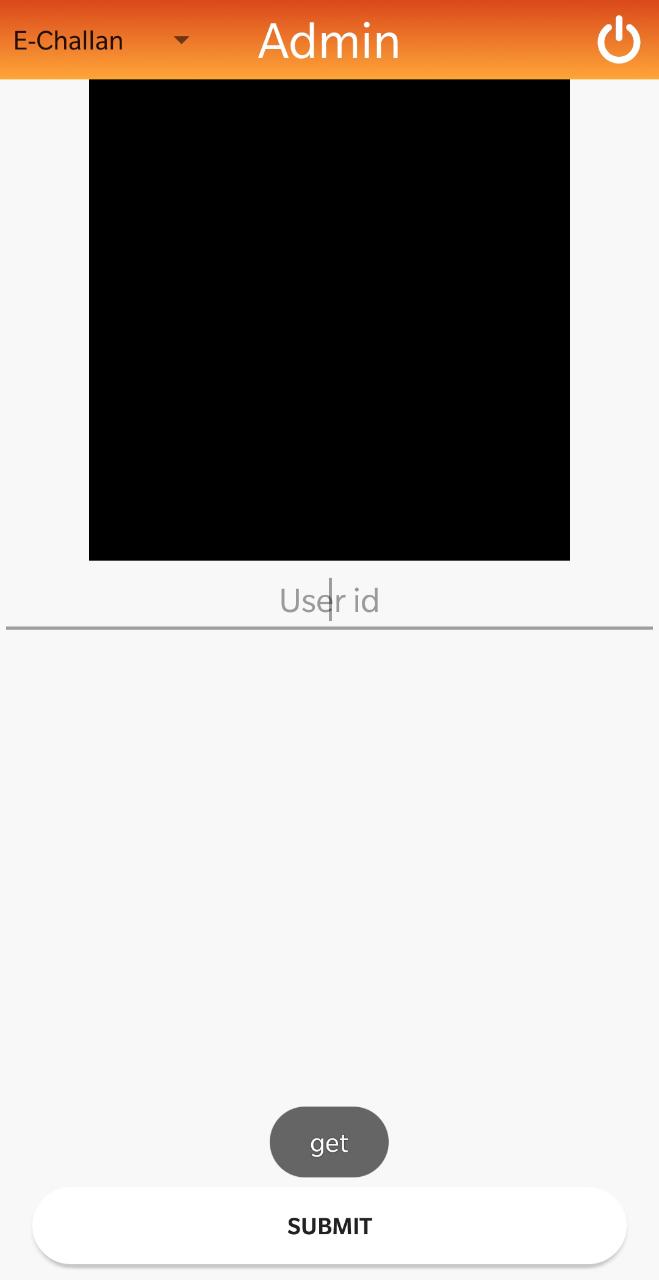


** **

****

**Admin Side Implementation**

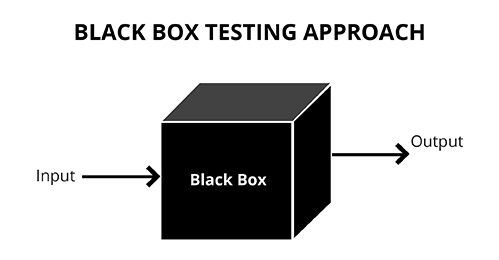
** **

****

**10.TESTING**

The testing of our proposed prototype is done on the bases of “Black Box Testing” in Software Engineering.

**10.1 Black Box Testing**



**Black Box Testing,** also known as Behavioural Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access
* Behaviour or performance error
* Initialization and termination errors

Definition by ISTQB

* **Black box testing:**Testing, either functional or non-functional, without reference to the internal structure of the component or system.
* **Black box test design technique:**Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.

**10.2 Advantages of Black Box Testing**

* Tests are done from a user’s point of view and will help in exposing discrepancies in the specifications.
* Tester need not know programming languages or how the software has been implemented.
* Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
* Test cases can be designed as soon as the specifications are complete.

**10.3 Disadvantages of Black Box Testing**

* Only a small number of possible inputs can be tested and many program paths will be left untested.
* Without clear specifications, which is the situation in many projects, test cases will be difficult to design.
* Tests can be redundant if the software designer/developer has already run a test case.
* Ever wondered why a soothsayer closes the eyes when foretelling events? So is almost the case in Black Box Testing.

**10.4 Test Case**

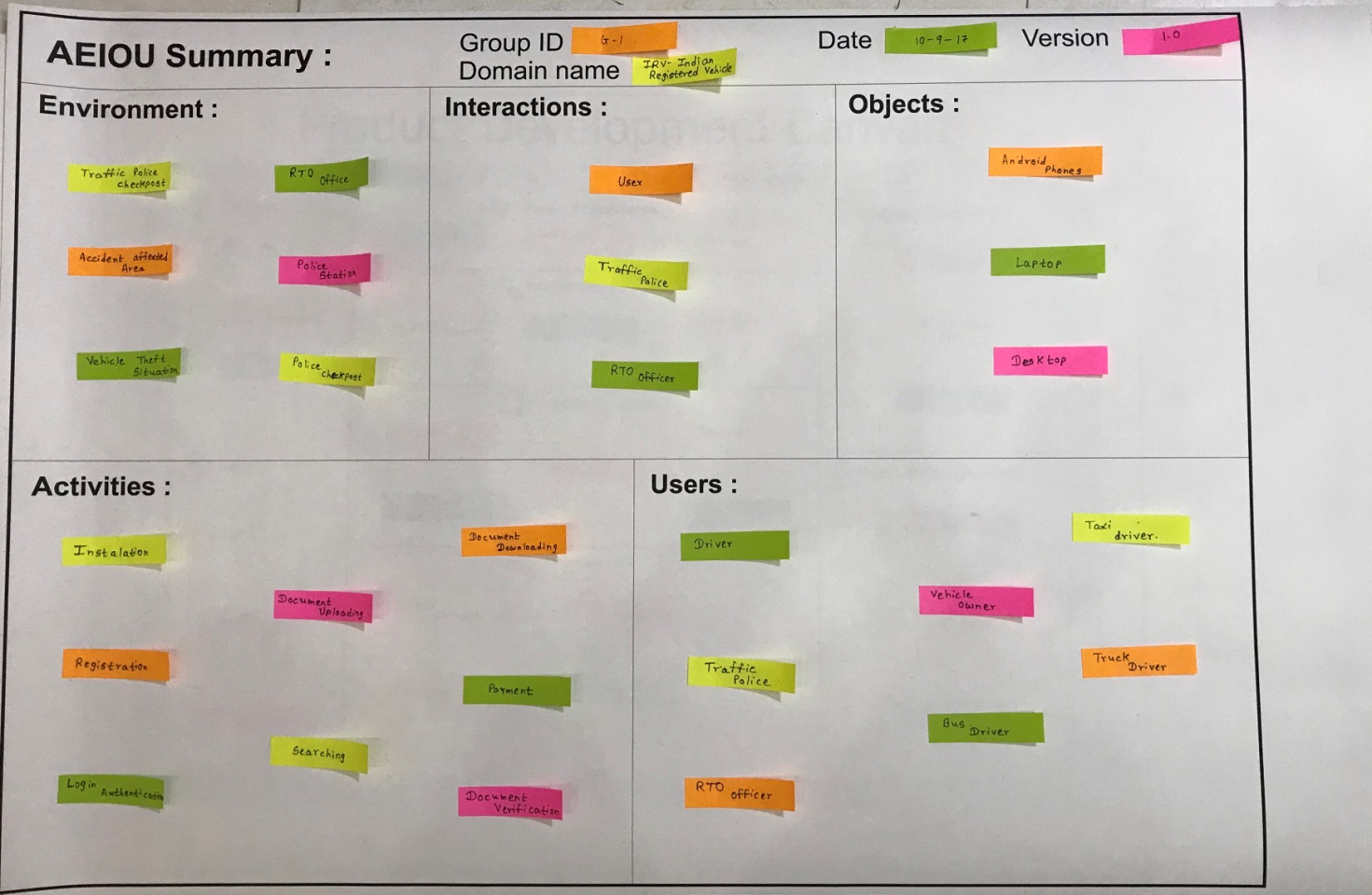
|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Test Case** | **Expected Outcome** | **Actual Outcome** |
| 1 | Login Page | User of the application enters name and phone number | User will login into the application . If the phone number provided is wrong have to re-enter phone number . |
| 2 | Document Uploading | User have to enter their document details and upload document photo in the application . | If the document submitted are false then the user will have to register himself . |

**11. Conclusion**

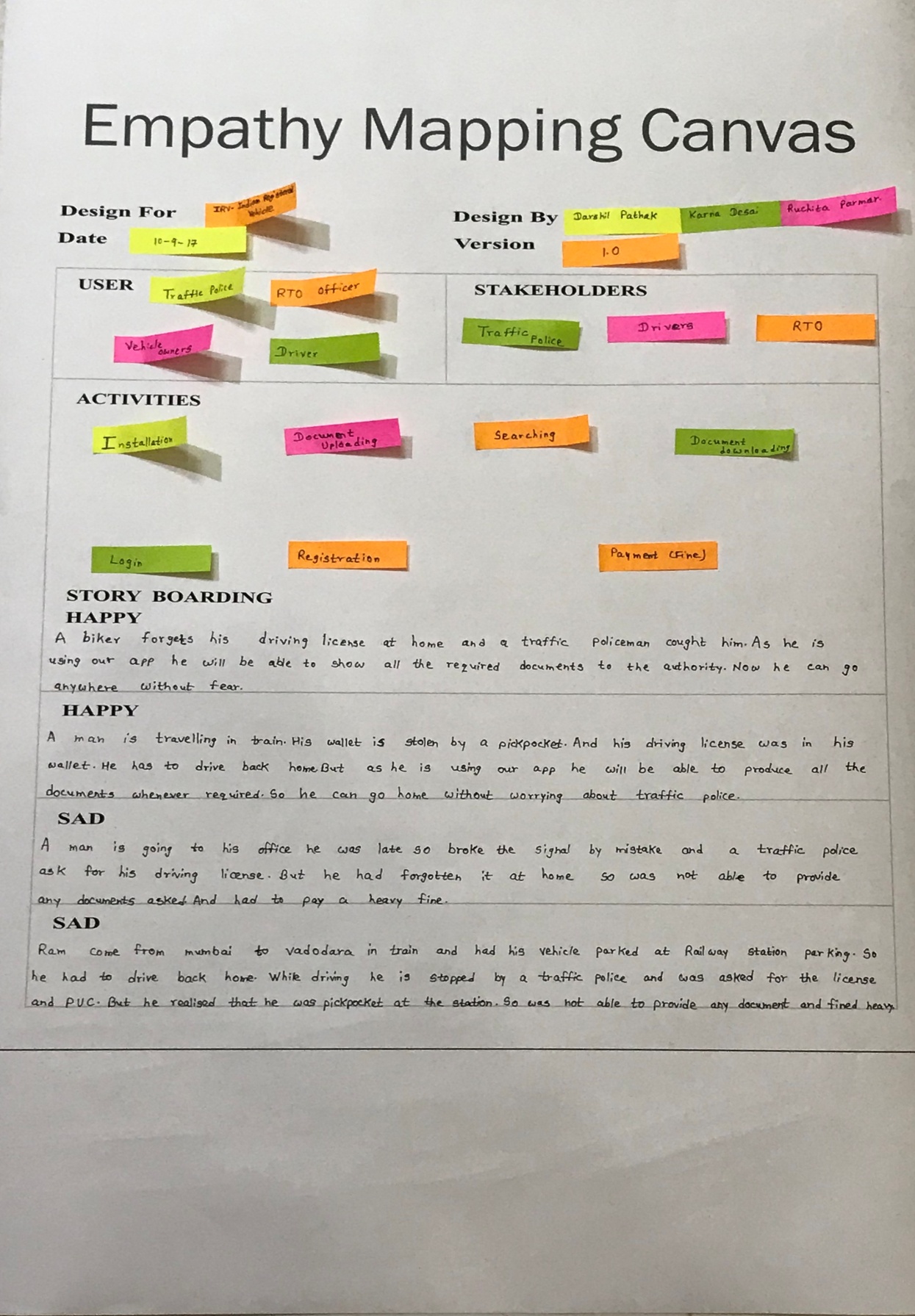
In Conclusion, we have learned different aspects of the project and learned how to manage different modules of the project, we also studied each module in detail and also did the requirement related search for the exact requirement needed for the project. We did analysis of our project and those things which came as hurdles are being modified or solved.

**Canvas**

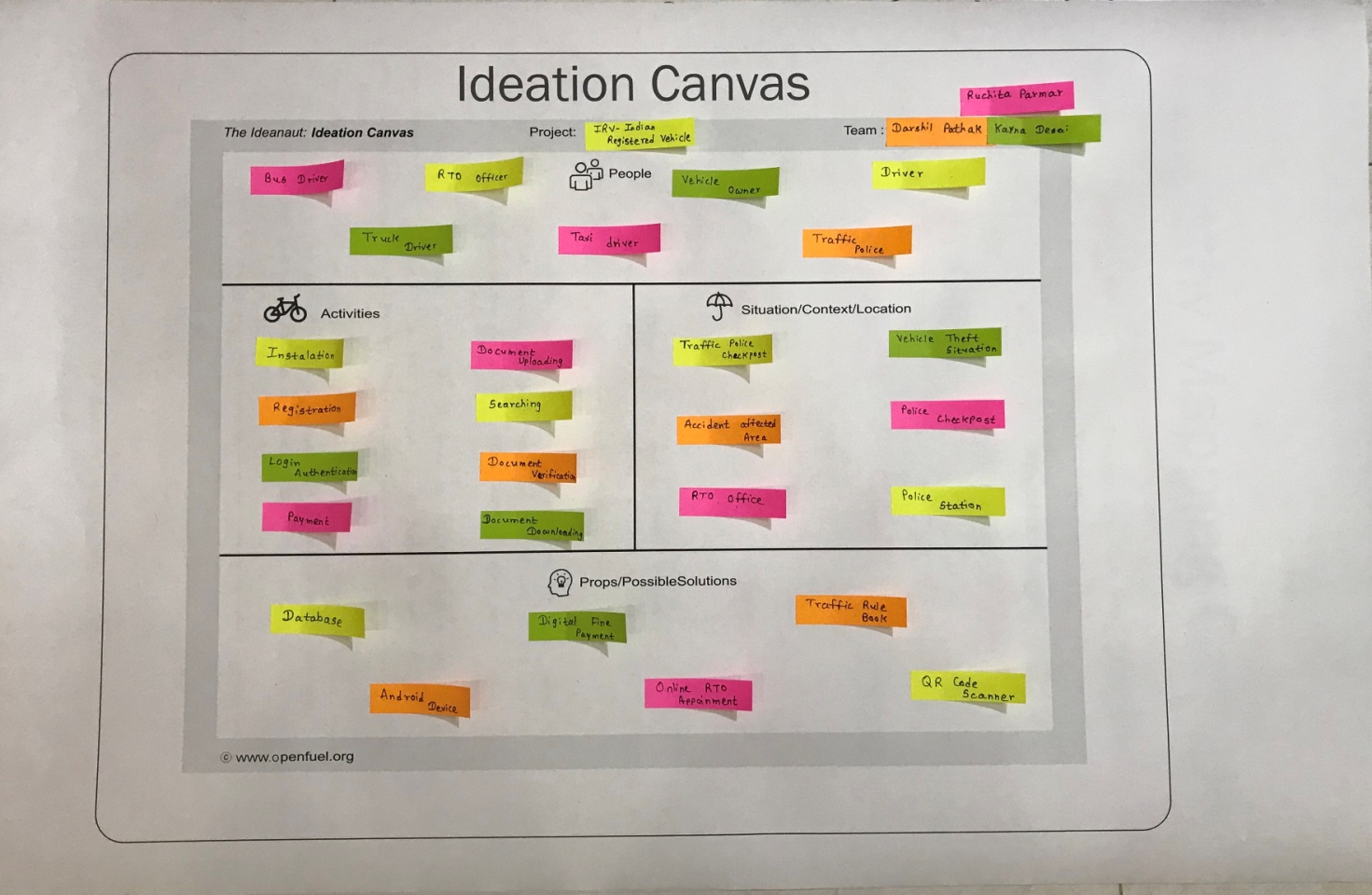
**AEIOU Canvas**

****

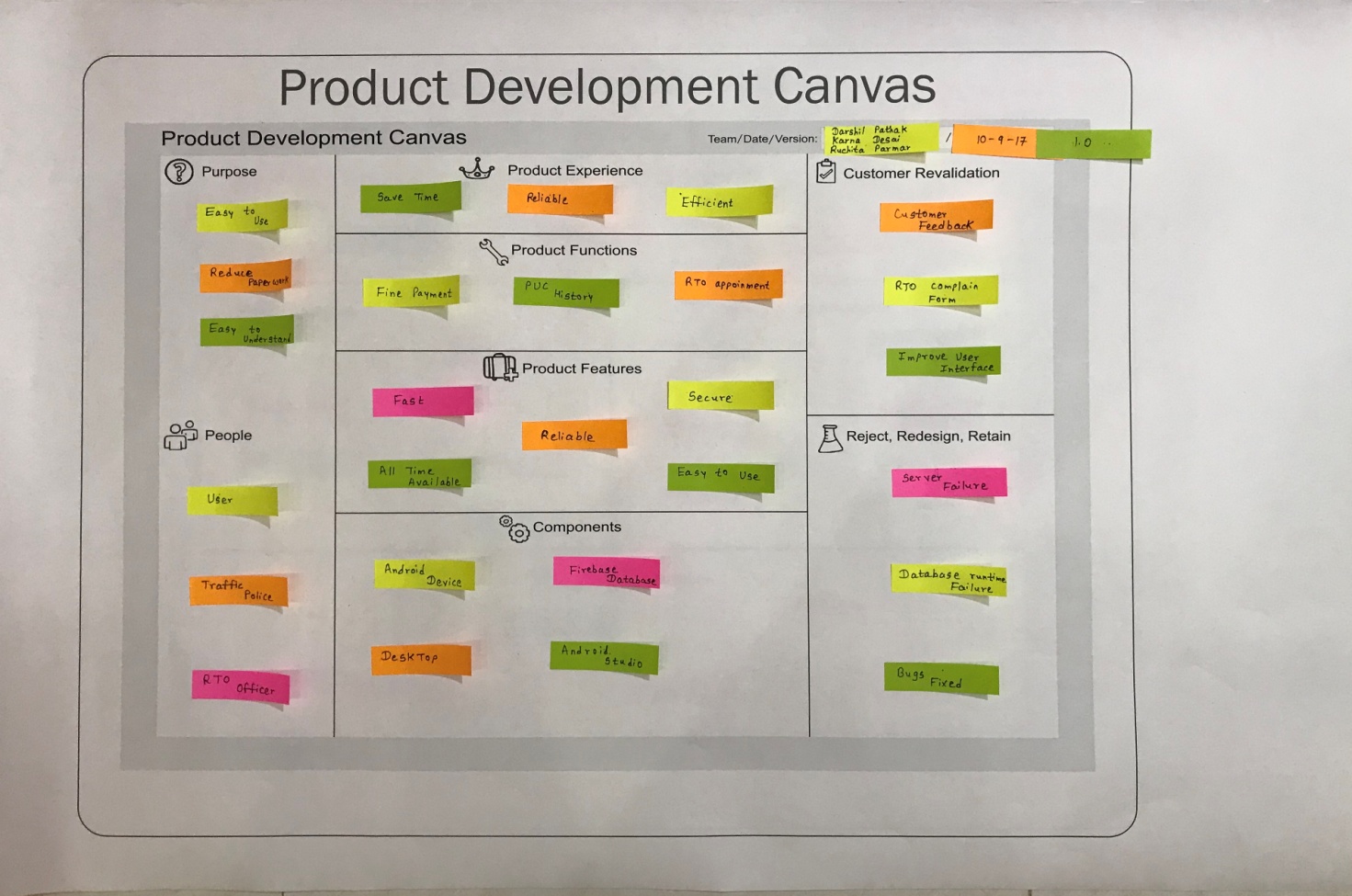
**Empathy Mapping Canvas**

****

**Ideation Canvas**

****

**Product Development Canvas**

****

**Reference**

https://en.wikipedia.org/

<https://en.wikipedia.org/wiki/Android_Studio>

["Android Studio 3.1"](https://android-developers.googleblog.com/2018/03/android-studio-3-1.html). Awndroid Developers Blog. March 26, 2018.

["The Android Source Code"](https://source.android.com/source/index.html). source.android.com. Retrieved February 2, 2017

["Firebase - CrunchBase"](http://www.crunchbase.com/organization/firebase). [CrunchBase](https://en.wikipedia.org/wiki/CrunchBase). Retrieved June 11,2014.

Michael Blaha and James Rambaugh (2004), Oriented Modelling and Design with UML (Second edition).

Roger S Pressman Software Engineering (Fifth Edition) McGraw-Hill Publication.

<https://www.lynda.com/Android-training-tutorials/947-0.html>

Android Application Development Black Book - Pradeep Kothari

Android Programming: The Big Nerd Ranch Guide

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

https://www.github.com/

https://www.tutorialspoint.com/

https://www.youtube.com/

https://developer.android.com/

http://rtogujarat.gov.in/

<https://material.google.com/>