Tribhuvan University

Faculty of Humanities and Social Sciences

A Project Report on

"Digital Vehicle"

In partial fulfilment of the requirement for the degree of Bachelor in Computer Application

(BCA)

<u>Submitted To:</u>

Department of Computer Application

Aadim National College

<u>Submitted By:</u>

Vivek Kumar Jurga Magar (10591190)

2079/03/05

Under the Supervision of

Mr. Navraj Bhattarai

Tribhuvan University

Faculty of Humanities and Social Sciences

Aadim National College

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by "**Vivek Kumar Jurga Magar"** entitled "**Digital Vehicle**" in partial fulfilment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

……………………
SIGNATURE

Navraj Bhattarai Supervisor

Chabahil, Chuchepati, Kathmandu 44600

Tribhuvan University

Faculty of Humanities and Social Sciences

Aadim National College

LETTER OF APPROVAL

This is to certify that this project is prepared by “**Vivek Kumar Jurga Magar”** entitled “**Digital Vehicle**” in partial fulfilment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

………………………………….
Mr. Navraj Bhattarai
Supervisor
Chabahil, Chuchepati, Kathmandu 44600

………………………………….
Mr. Yuba Raj Kalathoki
Coordinator
Chabahil, Chuchepati, Kathmandu 44600

………………………………….
Internal Examiner

………………………………….
External Examiner

Acknowledgements

We sincerely acknowledge and thank to those who have contributed their valuable time in helping us to achieve the success in our project work.

We would like to express our gratitude to all those gave us the possibility to complete this project. We want to thank **AADIM NATIONAL COLLEGE** for giving us the opportunity for doing this project.

We are indebted and thankful to our Project Guide. **Mr. Yuba Raj Kalathoki** to whom we owe his piece of knowledge for his valuable and timely guidance, co-operation, encouragement & time spent for doing this project work. We would also like to thank our HOD (Head of Department) **Mr. Shankar Prasad Sharma** and project supervisor **Mr. Navraj Bhattarai** for their assistance and insightful comments, and who willingly shared their expertise with us.

My sincere thanks to the IT staff for providing us sufficient information which helped us to complete our project successfully.

Lastly, we thanks to our parents and all our friends for inspiring my spirit to achieve this target.

Abstract

Now a day's population has become a major factor to be considered as a result the number of vehicle's are growing by increasing problems of vehicle registration, license registration, emission testing and insurance validity for Transport departments and vehicle related documents verification by traffic police. DOTM employees having lot of work burden of making registration, license issue, transfer etc., which requires lot of paper work. As a result, people can't get the things done in right time, which is the waste of time and energy.

Similarly, the vehicle owner sometimes forgets to carry the license, and forgets the vehicle documents at the time of enquiry. This paper proposed an approach to solve such problems that is by storing all the information related to vehicle and driver at database. And an android application is provided to traffic police to retrieve vehicle and license information. We can also add a provision to track a stolen vehicle. This approach is also useful to track down the vehicle tax and license expiry date.

Abbreviations and Acronyms

API	Application Programming Interface
IDE	Integrated Development Environment
JVM	Java Virtual Machine
TDD	Test Driven Development
XML	Extensible Markup Language
MVC	Model View Controller
MVP	Model View Presenter
MVVM	Model View ViewModel
OMT	Object Modelling
RX	ReactiveX
SSD	System Sequence Diagram
UI	User Interaction
UX	User Experience

List of Figures

List of Tables

Table of Contents

Chapter 1 Introduction

1.1 Introduction

In the current scenario, the key issue faced by people while travelling is to deal with the inefficiency and discrepancy in the traffic system and the people involved in managing the same. On being caught by the police, people must submit his/her driving license, Blue Book and/or other vehicle documents for verification. In case a person forgets/misplaces the documents, he is unnecessarily fined. In case a vehicle is booked for any traffic violations, then rider license or bluebook is taken by traffic officer and people have to do various paper works to get back their document which is time consuming.

In case a vehicle is stolen, the owner must contact the nearest police station. The process to lodge a complaint and subsequent response is slow and inefficient. In this age of science and technology, where the internet is in the palm of everyone's hands, a better system for managing the above process can be set up which would greatly reduce the burden on the daily people life as we as the traffic police.

1.2 Problem Statement

In today's scenario, people find difficulty carrying those paper form documents which occupy more space in pockets. People might forget it or lose it somewhere due to which they might have to pay fine to traffic officer. Externals factors like fire, water can damage documents physically and to remake the same documents people to go through different paper work and have to visit different government office which is time consuming and energy consuming. Documents might get stole or people might lose them somewhere.

1.3 Objectives

Some of the objectives of this project are:

- To make important vehicle documents digitalized,
- To make document secured with external factors or thieves,
- To check vehicle tax,

1.4 Scope and Limitations

1.4.1 Scope

- It will provide users to see their vehicle details and tax,
- It will provide easy checking of document,
- It will provide user to share their document details using QR code.

1.4.2 Limitations

Some of the limitations that exist in our sites are:

- Individual can only login from one device at a time,
- Tax renewable through app is not present,
- Only those who has license can use this system,
- Proper internet connection is required to run this app.

1.5 Development methodology

Software development methodology is a process or series of processes used in software development. Again, quite broad but that it is things like a design phase, a development phase. It is ways of thinking about things like waterfall being a non-iterative kind of process. Generally, it takes the form of defined phases. It is designed to describe the how the life cycle of a piece of software.

Being a simple and compact application, I have used Waterfall Model to develop my application Digital Vehicle because it seemed the best model for my project. Waterfall model is a classical model used in SDLC which follows linear and sequential approach. Meaning one only moves forward onto the next phase once the completion of previous phase during SDLC.

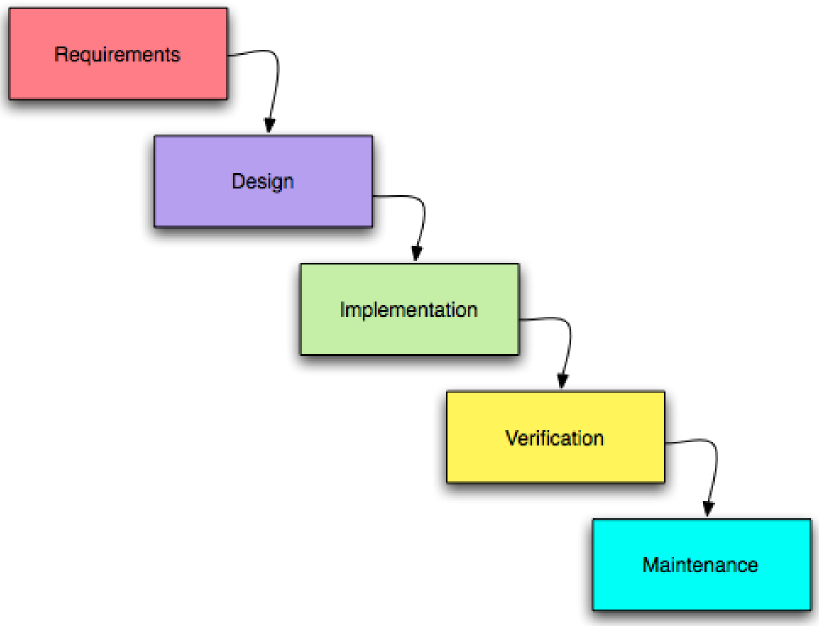


Fig 1.1: Waterfall Model [2]

1.6 Report Organization

This report document contains five chapters in total. Chapter 1 being Introduction i.e., current chapter. Background Study and Literature Review, explained in chapter 2, System Analysis and Design in chapter 3. Similarly, chapter four explains the Implementation and Testing; chapter 5 explains Conclusion and Future Recommendations.

Chapter 2 Background Study and Literature Review

2.1 Background Study

As in today's world, internet is in the palm of every hand. Along with population the vehicle's uses is increasing day by day. The internet has been favored by more and more people for its high efficiency. The digital vehicle app is a form of digitalizing vehicle document in database and accessing it through mobile app. The Digital Vehicle app is mobile-based application where user can add vehicle documents, see the vehicle documents. To address this issue, we adopt Digital Vehicle which offers the best features among all the other features that are on the web, mobile. This project will help people to store their vehicle documents and see tax of the vehicle documents among all the features given by others app.

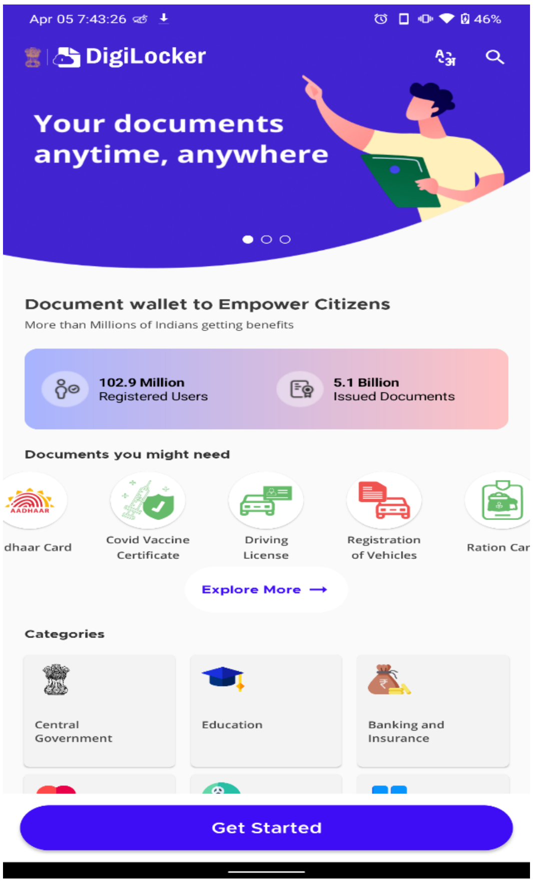
2.2 Literature Review

2.2.1 Study of existing system

Many applications which store vehicle documents already exist in today's world. But some of the most popular are DigiLocker, Nagarik App etc. Digital Vehicle will somehow be similar to them. The main goal of Digital Vehicle is to provide the best and user-friendly environment to the users than other application. To get a rough idea about how I want to build my application, I reviewed some of the well renowned applications.

I. **https://www.digilocker.gov.in [3]**

DigiLocker is a flagship initiative of Ministry of Electronics & IT (MeitY) under Digital India program. DigiLocker aims at 'Digital Empowerment' of citizen by providing access to authentic digital documents to citizen's digital document wallet. The issued documents in DigiLocker system are deemed to be at par with original physical documents as per Rule 9A of the Information Technology (Preservation and Retention of Information by Intermediaries providing Digital Locker facilities) Rules, 2016 notified on February 8, 2017 vide G.S.R. 711(E).[3]



Apr 05 7:43:26 46%

DigiLocker

Your documents anytime, anywhere

Document wallet to Empower Citizens

More than Millions of Indians getting benefits

102.9 Million Registered Users

5.1 Billion Issued Documents

Documents you might need

dhaar Card

Covid Vaccine Certificate

Driving License

Registration of Vehicles

Ration Car

Explore More ⟶

Categories

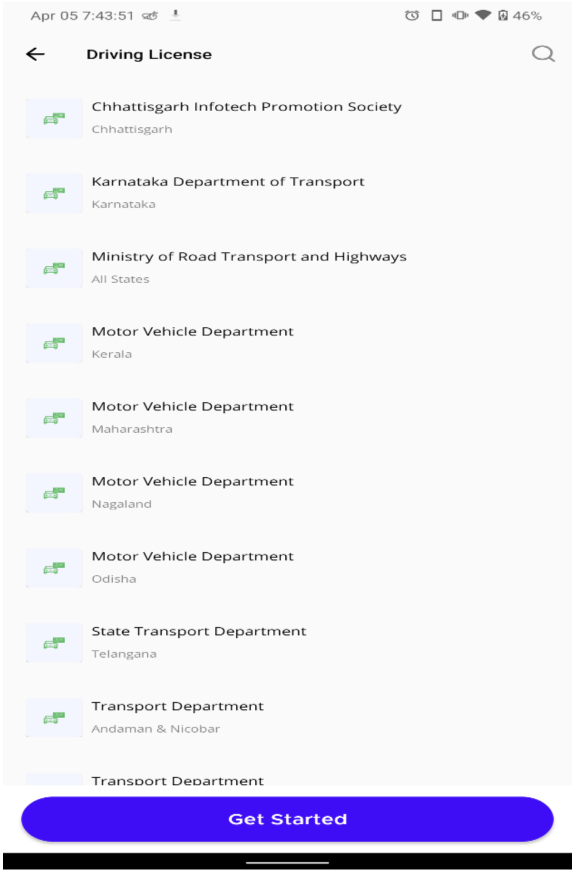
Central Government

Education

Banking and Insurance

Get Started

Fig 2.1: Home Page of DigiLocker



Apr 05 7:43:51 46%

← Driving License

Chhattisgarh Infotech Promotion Society
Chhattisgarh

Karnataka Department of Transport
Karnataka

Ministry of Road Transport and Highways
All States

Motor Vehicle Department
Kerala

Motor Vehicle Department
Maharashtra

Motor Vehicle Department
Nagaland

Motor Vehicle Department
Odisha

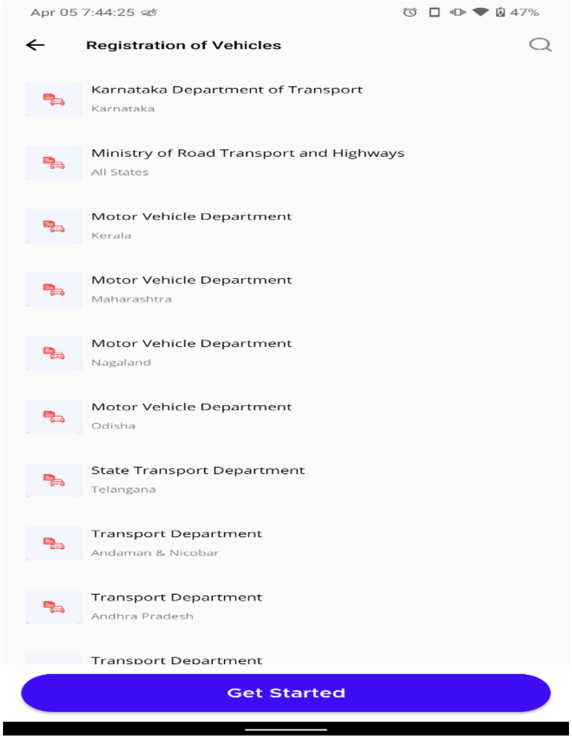
State Transport Department
Telangana

Transport Department
Andaman & Nicobar

Transport Department

Get Started

Fig 2.2: Driving License Page



Apr 05 7:44:25 47%

← Registration of Vehicles

Karnataka Department of Transport
Karnataka

Ministry of Road Transport and Highways
All States

Motor Vehicle Department
Kerala

Motor Vehicle Department
Maharashtra

Motor Vehicle Department
Nagaland

Motor Vehicle Department
Odisha

State Transport Department
Telangana

Transport Department
Andaman & Nicobar

Transport Department
Andhra Pradesh

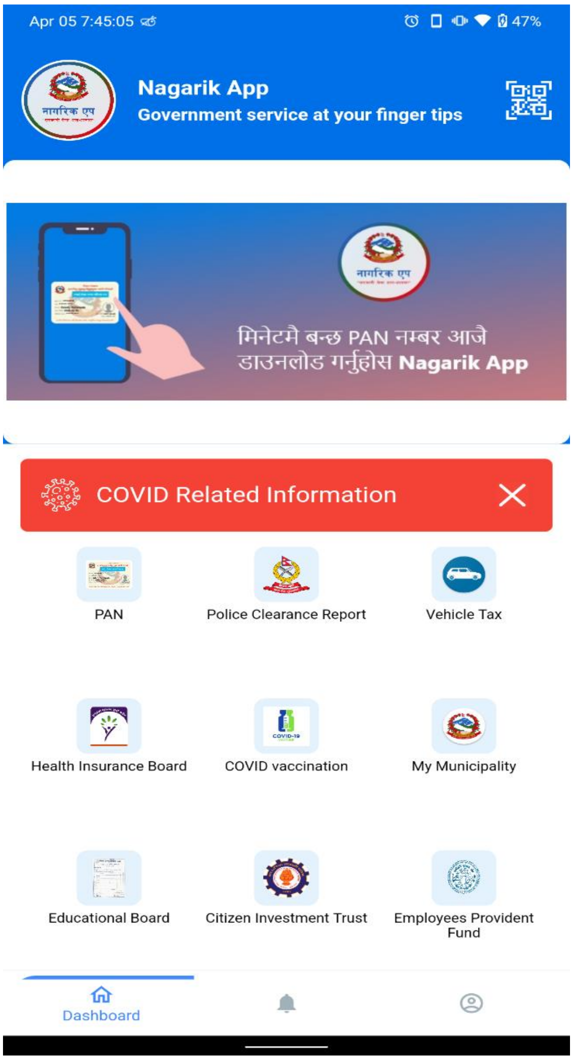
Transport Department

Get Started

Fig 2.3: Vehicle Registration Page

II. https://nagarikapp.gov.np [4]

Nagarik Mobile App is a software system that runs services on mobile and tablets. The Nagarik Mobile App is the beginning of a new era of digital Nepal. If the government continues to make this app more effective in the coming days, the era of paper documentation in Nepal will come to an end. The government has announced plans to build Digital Nepal for good governance, development, and prosperity. The citizen app envisioned by the government policy and program of 2075/76 is believed to be an important cornerstone for Digital Nepal. The online services of various government and public bodies can be easily accessed from the same app through this app, and this app will also act as a service delivery gateway so that the electronic systems of government and public bodies do not have to be interconnected separately to establish contact.[4]



Apr 05 7:45:05 47%

Nagarik App
Government service at your finger tips

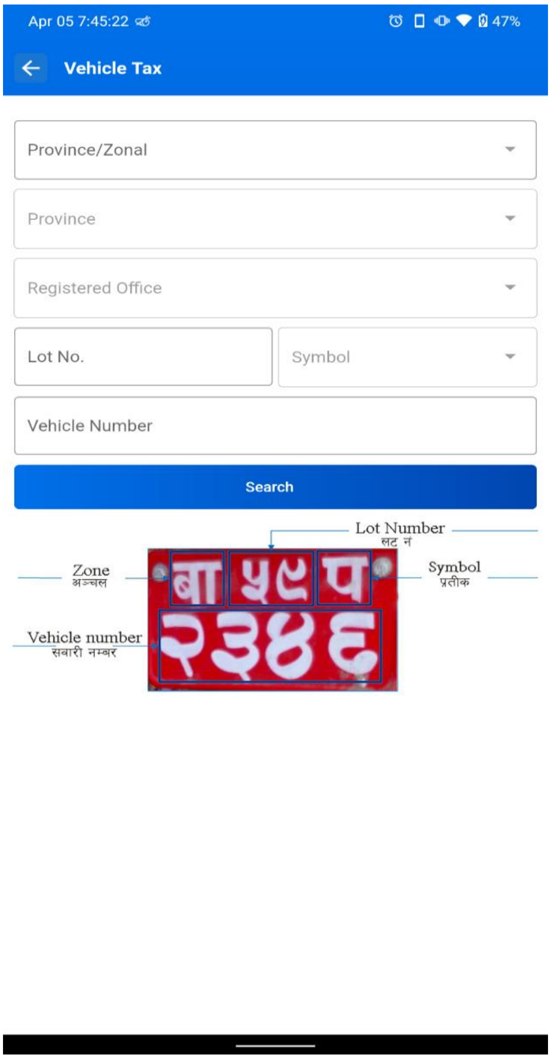
मिनेटमै बन्छ PAN नम्बर आजै
डाउनलोड गर्नुहोस Nagarik App

COVID Related Information

PAN
Police Clearance Report
Vehicle Tax
Health Insurance Board
COVID vaccination
My Municipality
Educational Board
Citizen Investment Trust
Employees Provident Fund

Dashboard

Fig 2.4: Home Page of Nagarik APP

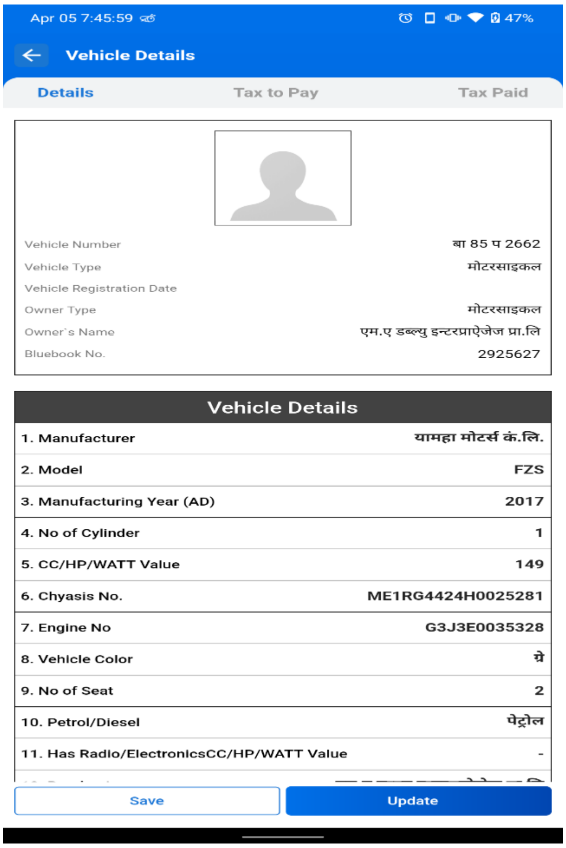


Apr 05 7:45:22 47%

Vehicle Tax

Province/Zonal
Province
Registered Office
Lot No.
Symbol
Vehicle Number
Search

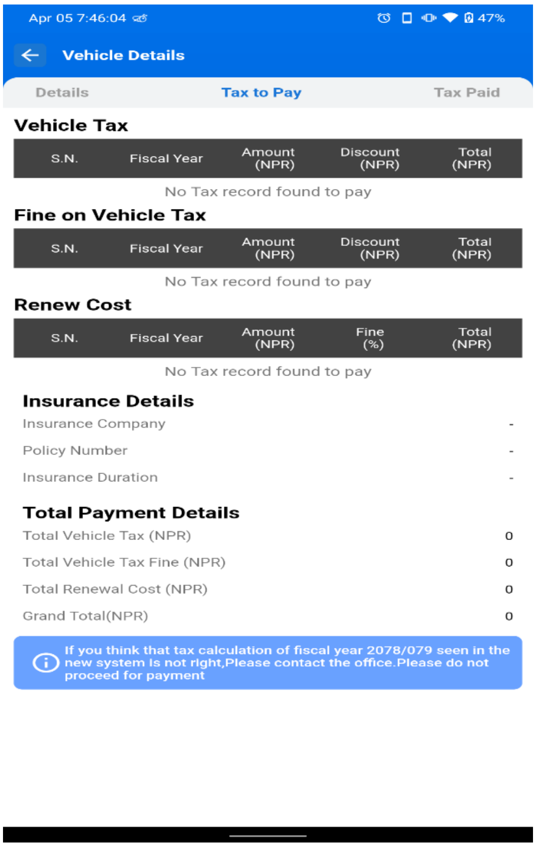
Fig 2.5: Bluebook Search Page of Nagarik APP



Vehicle Number	बा 85 प 2662
Vehicle Type	मोटरसाइकल
Vehicle Registration Date	
Owner Type	मोटरसाइकल
Owner`s Name	एम.ए डब्ल्यु इन्टरप्राऐजेज प्रा.लि
Bluebook No.	2925627

Vehicle Details	
1. Manufacturer	यामहा मोटर्स कं.लि.
2. Model	FZS
3. Manufacturing Year (AD)	2017
4. No of Cylinder	1
5. CC/HP/WATT Value	149
6. Chyasis No.	ME1RG4424H0025281
7. Engine No	G3J3E0035328
8. Vehicle Color	ग्रे
9. No of Seat	2
10. Petrol/Diesel	पेट्रोल
11. Has Radio/ElectronicsCC/HP/WATT Value	-

Fig 2.6: Vehicle Details Page



Vehicle Tax

S.N.	Fiscal Year	Amount (NPR)	Discount (NPR)	Total (NPR)

No Tax record found to pay

Fine on Vehicle Tax

S.N.	Fiscal Year	Amount (NPR)	Discount (NPR)	Total (NPR)

No Tax record found to pay

Renew Cost

S.N.	Fiscal Year	Amount (NPR)	Fine (%)	Total (NPR)

No Tax record found to pay

Insurance Details

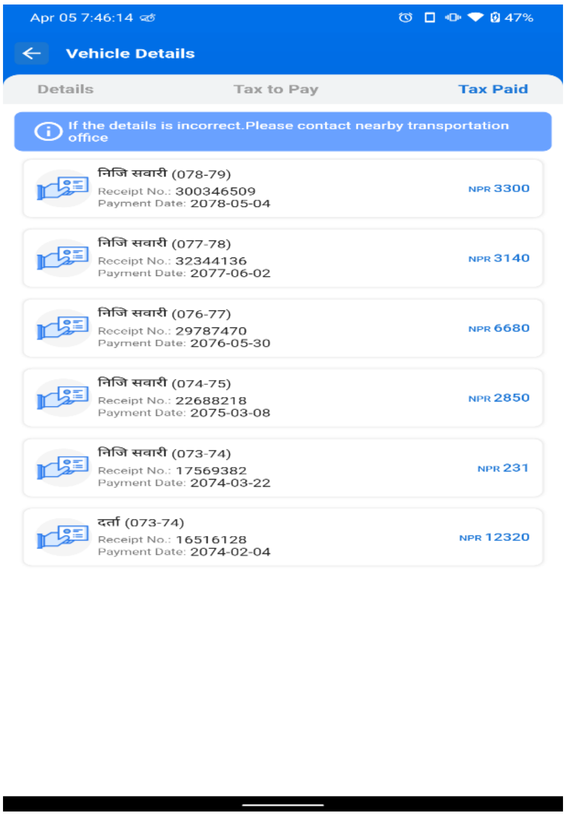
Insurance Company	-
Policy Number	-
Insurance Duration	-

Total Payment Details

Total Vehicle Tax (NPR)	0
Total Vehicle Tax Fine (NPR)	0
Total Renewal Cost (NPR)	0
Grand Total(NPR)	0

If you think that tax calculation of fiscal year 2078/079 seen in the new system is not right,Please contact the office.Please do not proceed for payment

Fig 2.7: Tax to pay Page



If the details is incorrect.Please contact nearby transportation office

निजि सवारी (078-79)
Receipt No.: 300346509
Payment Date: 2078-05-04
NPR 3300

निजि सवारी (077-78)
Receipt No.: 32344136
Payment Date: 2077-06-02
NPR 3140

निजि सवारी (076-77)
Receipt No.: 29787470
Payment Date: 2076-05-30
NPR 6680

निजि सवारी (074-75)
Receipt No.: 22688218
Payment Date: 2075-03-08
NPR 2850

निजि सवारी (073-74)
Receipt No.: 17569382
Payment Date: 2074-03-22
NPR 231

दर्ता (073-74)
Receipt No.: 16516128
Payment Date: 2074-02-04
NPR 12320

Fig 2.8: Tax Paid Page

Chapter 3 System Analysis and Design

3.1 System Analysis

Before starting any new system/ software development, it is important to plan how it will be developed, tested and maintained. It is a key to success for any project. For the success of the project, I along with my supervisor were involved in the discussions regarding the development and work flow of the project

3.1.1 Requirement Analysis

Requirement analysis is done while developing a system before implementing it, it is necessary to analyses the whole system requirement. It is categories into mainly two parts, namely: functional and non-functional requirements. it is necessary to visualize the layout, design and all features intended to be incorporated. In addition, how users will interact with each page and icon and how the website/App should perform (behavior, load time etc.).

3.1.1.1 Functional Requirement

A functional requirement is done while developing a system should do when it is given input. Some of functional requirement are given below:

1. User
   - Can Login/ Register (Request)
   - Can view License
   - Can view Bluebook
   - Can view Vehicle Tax
   - Can Lend Bluebook
   - Can report for theft
2. Traffic
   - Can view user License
   - Can Login
   - Can view user Vehicle Details

The below figure shows the use case diagram of our application (Digital Vehicle). The three main module of this application are users, traffic and admin. User can login/register into application, view license details and vehicle details, share license details and vehicle details. Traffic can view license and vehicle details along with log in into application whereas admin can manage and view user and traffic details.

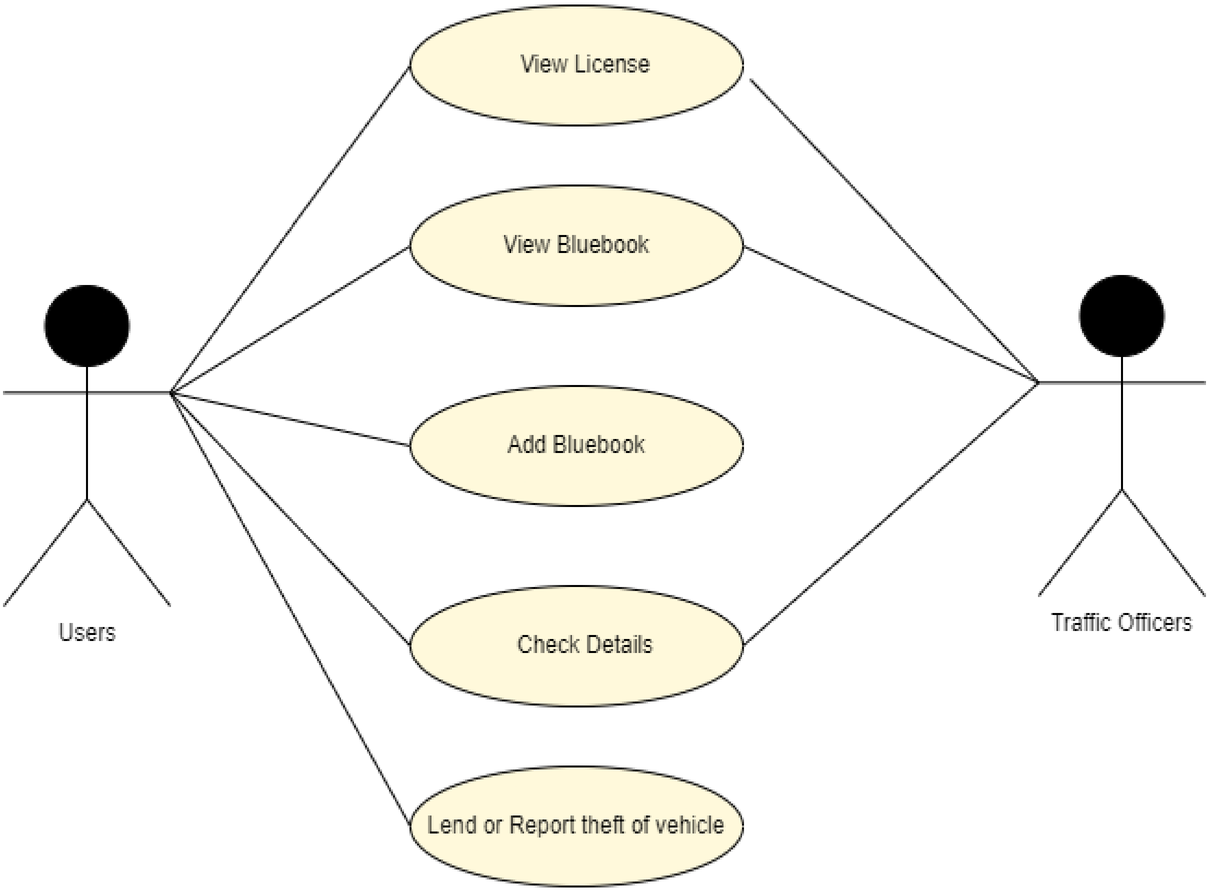


Fig 3.1: Use Case of Digital Vehicle

3.1.1.2 Non-Functional Study

The non-functional requirements of the tourism information service system include the security, reliability, portability and scalability, and maintainability of the system. Each specific function is introduced as follows:

- Security: The application can only be accessed by registered and validate user using their mobile number. Also, the system uses Firebase Authentication for better security.
- Reliability: The application is reliable as it has considered the damages that can be caused by incomplete data.

- Usability: Digital Vehicle provides an appropriate output, when necessary, features in the correct format are given as input.
- Maintainability: The application is trained and tested according to the correct set of features provided and the data is maintained in appropriate format.
- Availability: The application is made accessible and available anywhere and anytime. All you need is an android mobile device and a good internet connection.

3.1.2 Feasibility Study

Feasibility analysis is a part of system analysis carried to confirm that the system being developing is actually feasible or not. This is the phase where any system designer is able to know whether to start the project or not.

We perform some study and analyzed the system and got to know that it is feasible to make the system. Mainly three types of feasibility studies were done with this system analysis, namely: Economic feasibility, Operational feasibility and technical feasibility.

Feasibility studies undergo three major analyses to predict the system to be success and they are as follows:

- Operational Feasibility
- Technical Feasibility
- Economic Feasibility
- Schedule Feasibility

- **Economic Feasibility**

  Developing and deploying this system has a very little economic cost. All the resources required to develop the system are computer and designer information. For development, PCs that support any operating system with some application is sufficient. Further, it does not cost much to develop and access this system and hence, we can say that it is economically feasible to develop the system.

- **Operational feasibility**

This system provides a simple user interface, which can be easily used by user having PCs. This system will provide correct results. The user must not feel threatened by the system, instead must accept it as a necessity. This system uses simple technologies to design. So, it is user friendly. Hence, this system is operationally feasible too.

- **Technical feasibility**

  This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources**.** User will require internet browser. It will run on all the existing web browsers with latest version. Keeping all these facts in mind we had selected the favorable hardware and software utilities to make it more feasible.

- **Schedule Feasibility**

  In it, I estimated how much time the project will take to complete. To calculate and continually re-examining whether it is possible to complete all the amount and scope of work lying ahead, utilizing the given number of resources, within the required period of time.

3.1.3 Object Modelling (E-R diagram)

Object modelling (OMT) is an approach for software modelling and designing, it is used to test physical entities before building them. OMT is used for visualization and reduction of complexity. OMT can be categorized into Object Model, Dynamic Model, and Functional Model.

I. **Class Diagram:** Class diagram is also called a static diagram because it represents the static view of an application. In other words, class diagram is used for visualizing, describing, and documenting different aspects of a system. It contains of attributes and operations of a class and also the constraints imposed on the system.

   The below diagram shows some of the classes and their interrelationship with each other. Whenever we create a model of a class, its divided into three sections; Class Name, Attributes, and their Methods. In order to represent a class's attributes and method, we follow following approach + for Public Access Modifier, - for Private

Access Modifier and # for Protected Access Modifier. The relationship between classes is represented by crowfoot's notation.

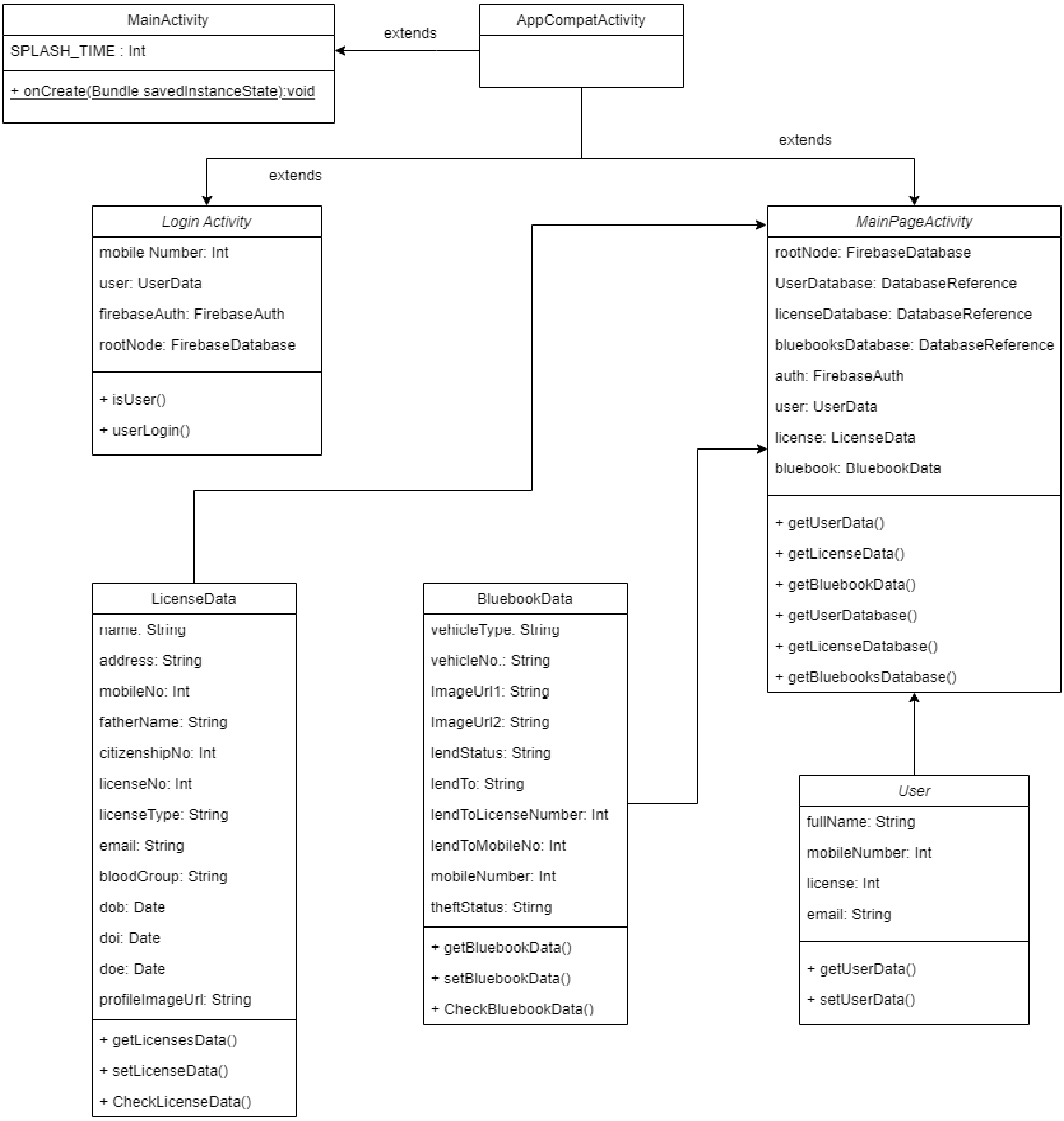


Fig 3.2: Class Diagram

II. **Object Diagram:** Object diagram are dependent on class diagram because they are derived from it. It represents an instance of class diagram. Object diagram represents a snapshot of the system at a particular moment.

The below diagram shows how the state of the system flows when a user enters a detail of name and mobile number then it creates a new object for license data and a bluebook data at a point in time. Mobile number works as the primary key which is available in bluebook and license too.

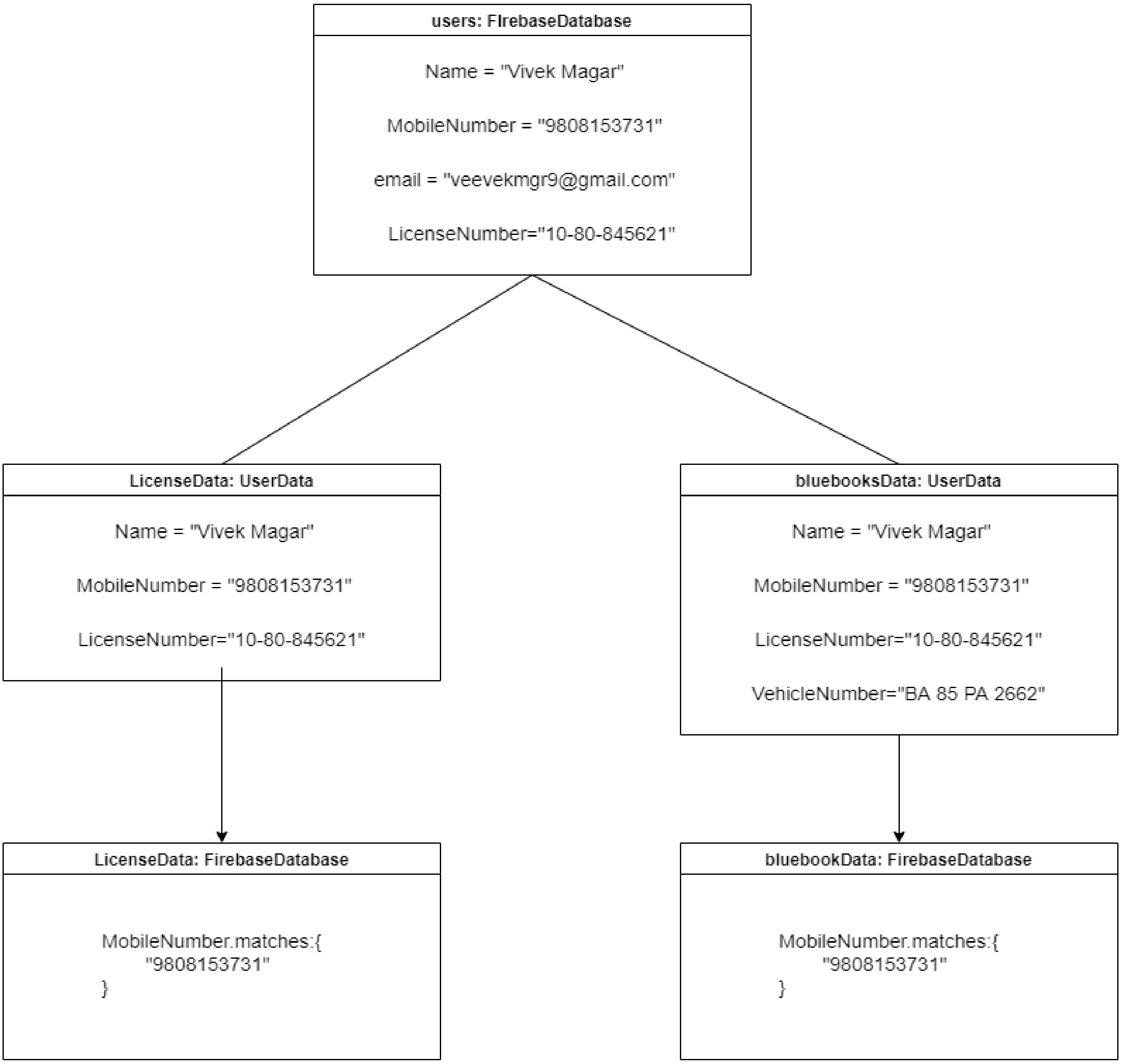

users: FIrebaseDatabase
Name = "Vivek Magar"
MobileNumber = "9808153731"
email = "veevekmgr9@gmail.com"
LicenseNumber="10-80-845621"
LicenseData: UserData
Name = "Vivek Magar"
MobileNumber = "9808153731"
LicenseNumber="10-80-845621"
bluebooksData: UserData
Name = "Vivek Magar"
MobileNumber = "9808153731"
LicenseNumber="10-80-845621"
VehicleNumber="BA 85 PA 2662"
LicenseData: FirebaseDatabase
MobileNumber.matches:{ "9808153731" }
bluebookData: FirebaseDatabase
MobileNumber.matches:{ "9808153731" }

Fig 3.3: Object Diagram

3.1.4 Dynamic Modelling

Dynamic Modelling describes those aspects of the system that are concerned with time and sequencing of the operations. It is used to specify and implement the control aspect of the system. Dynamic model is represented graphically with the help of state and sequence diagram.

I. **State Diagram:** A state diagram shows the behavior of classes in response to external stimuli. Specifically, a state diagram describes the behavior of a single object in response to a series of events in a system. Sometimes it's also known as a Harel state chart or a state machine diagram. This UML diagram models the dynamic flow of control from state to state of a particular object within a system.

The below state diagram, explains the transition from when a user logs in and until user logs out of the system. It also shows how a state is being transitioned to another state with the help of events such as Do Login, Do Logout, etc.

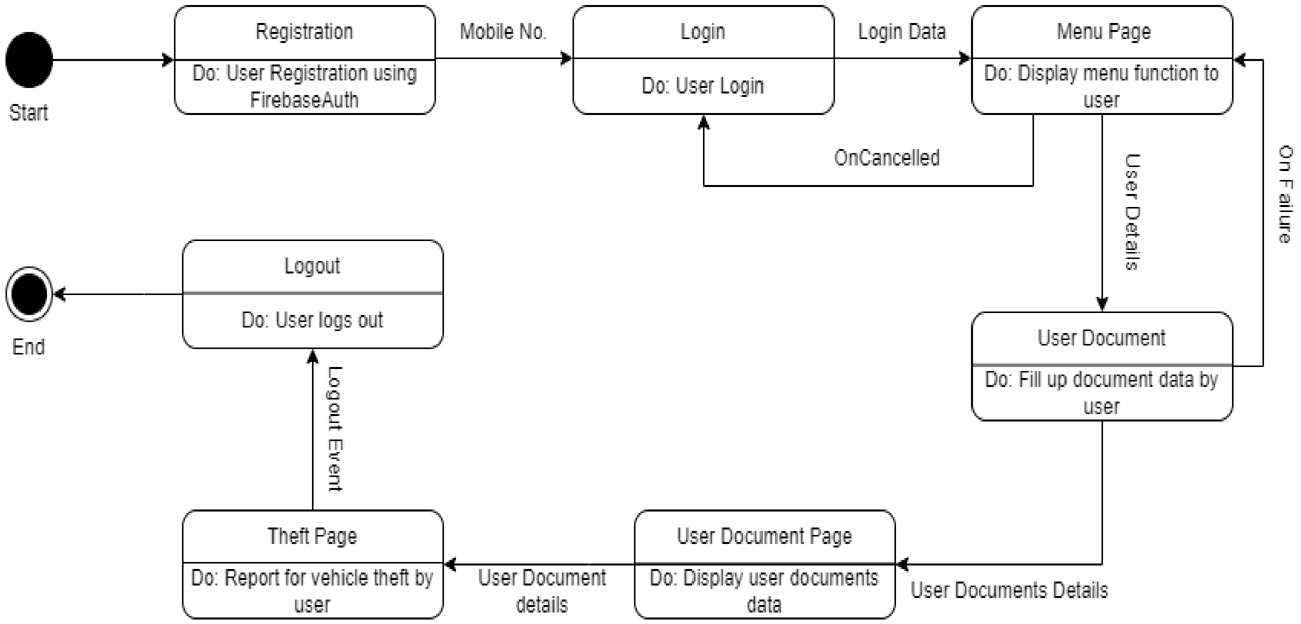


Fig 3.4: State Diagram

II. **Sequence Diagram:** Sequence diagram is an interaction diagram that shows how operations are carried out and what messages are received and sent. It is also called as System Sequence Diagram (SSD). SSD focus on lifelines, or the processes and the object that lives simultaneously.

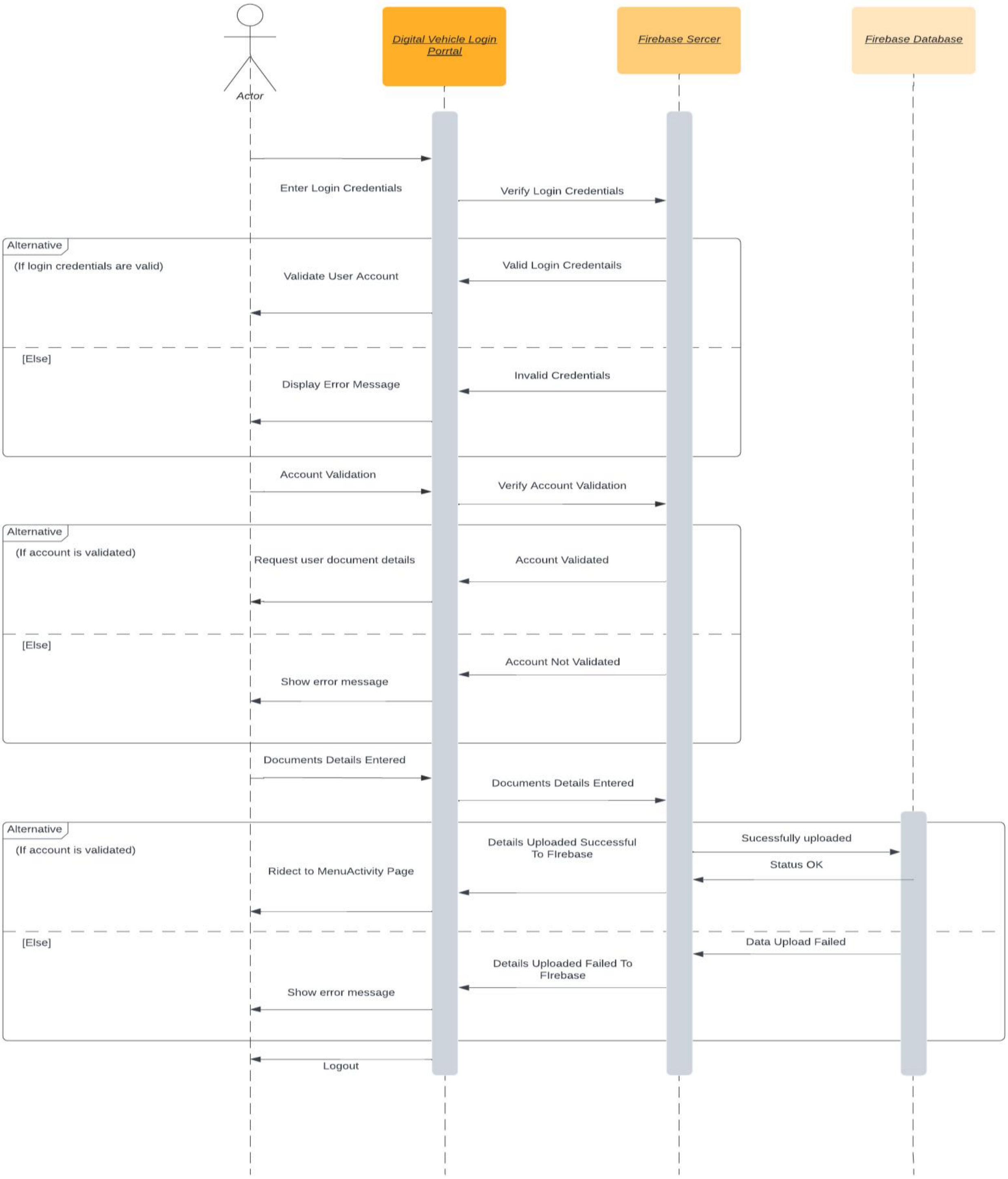


Fig 3.5: Sequence Diagram

The above sequence diagram explains about the login and logout process of user. It explains the order of those process. Once the user logs in data is sent to the firebase database to check if user has uploaded personal details (license No, email, name, mobile Number.). After that is check actions are done accordingly.

3.1.6 Process Modelling (Activity Diagram):

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

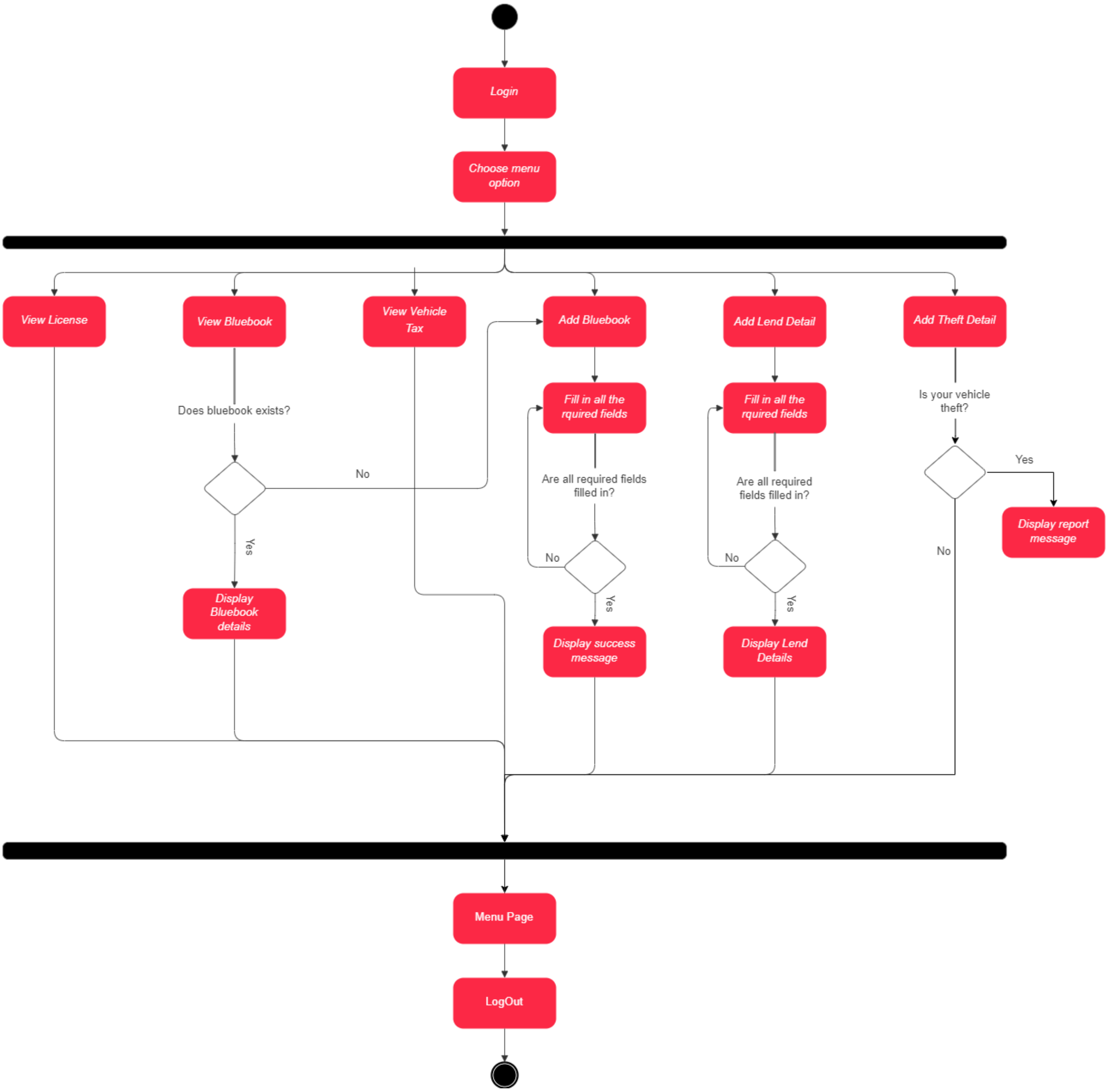


Fig 3.6: Activity Diagram

The above figure shows the series and action of flow one can perform in a system once the user logs in. It visualizes what process can be done in the system and also concurrent processes that can be done in the system, e.g., View Bluebook, View Vehicle Tax, View License etc.

3.2.1 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

The below figure shows what component are required for user Activity. It visualizes how those components are used to make the functionality and how components are helping to make the functionality of the system using the help of interfaces.

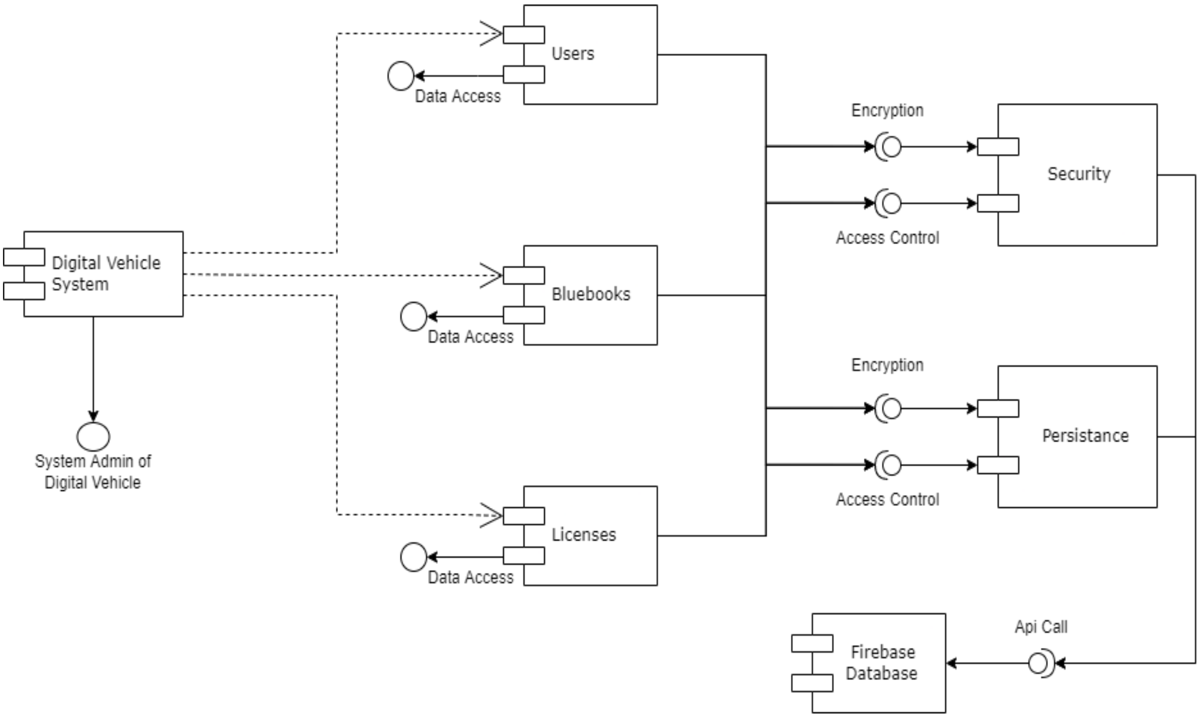


Fig 3.7: Component Diagram

3.2.2 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

The below figure assumes that the application is an android application, which is deployed in an environment using Firebase Server. The application connects to the Firebase Server using Application server using HTTP protocol

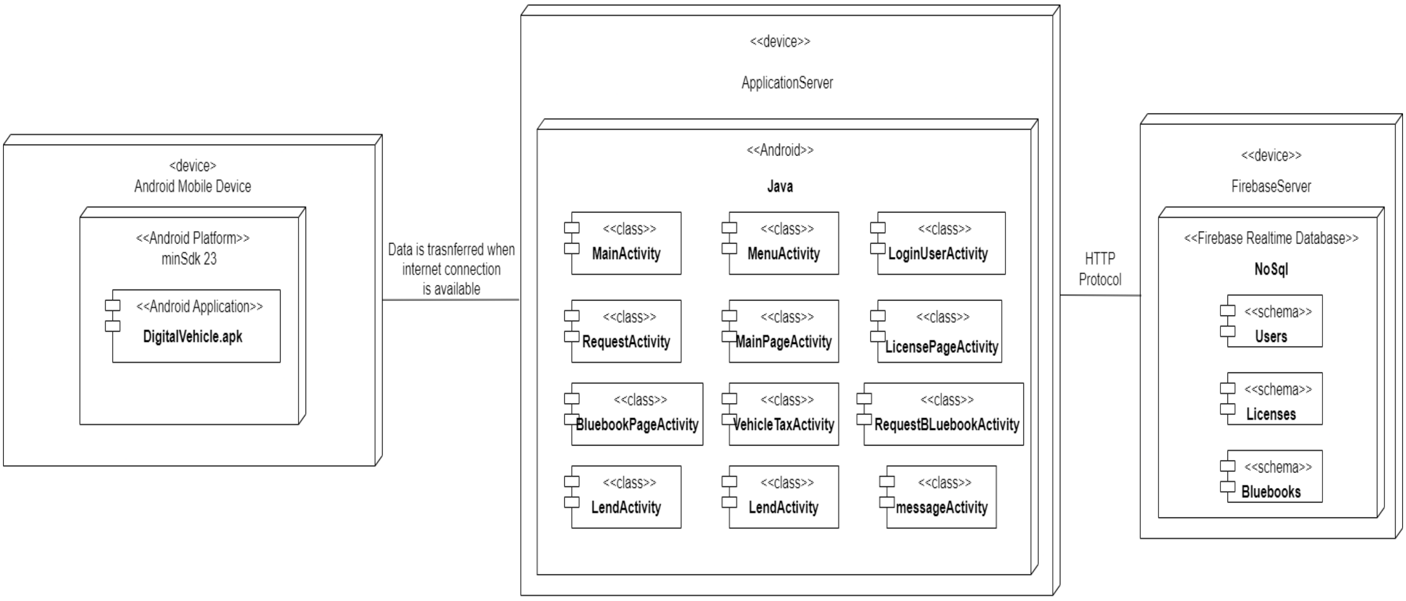


Fig 3.8: Deployment Diagram

Chapter 4 Implementation and Testing

4.1 Implementation

After designing the app, the only thing that needs to be done is implement it so that we can release it as per the user satisfaction. Implementing the system requires a lot of resources and explanation which will not be completely explained in this report; however, some major aspects of the system are described below:

4.1.1 Tools Used

The various different tools that have been used for development of this application “Digital Vehicle” are as below:

a. **UI:** Being made on native android and XML being the only platform to design the app, the entire UI of the app was made with the help of XML.

b. **Design:** Before building any system, it is better to design the UI of the application. A design tools called Figma was used to design the UI of this project “Digital Vehicle”. Figma is a vector graphics editor and prototyping tool which is primarily web-based, with additional offline features enabled by desktop application for windows and macOS.

c. **IDE:** Android Studio is the IDE which has been used during the development of this application. Android Studio is an IDE based on IntelliJ IDEA which provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV etc.

d. **Programming Language:** As JAVA provides the concept of OOPS (Object Oriented Programming) and it is more proficient because they are extensible, scalable and adaptable, I have used JAVA as programming languages to develop or build this project.

e. **Database:** For database, I used Firebase Realtime Database which is provided by Firebase. It is a cloud-hosted database in which data is stored as JSON and the data is synchronized in real-time to every connected client. Similarly for storing images Firebase Storage was used which is a service that developers can use to store and download files generated directly by client without the need of server-side code.

4.1.2 Implementation details of modules

a. **UI and UX of the app:** After the design that was made using Figma, it was time to start implementing the design into real application. The whole UI of the app is made using XML as the app is not made using any frameworks such as Flutter nor is based on Jetpack Compose.

b. **Phone Verification:** The SMS verification provided by Firebase is used to provide OTP code to user which provide more security and provide easy login to user.

c. **Displaying Network Images:** Android doesn't provide any widgets or features to display network image into the app directly as it cannot be done in main thread. So, a third-party dependency called Glide was used in this project.

d. **QR Code Generator and Scanner:** For the generation of QR code of license details and bluebook details and for scanning of those generated QR code I have used a third-party dependency called barcode zxing.

4.2 Testing

After the completion of the implementation phase and even during the development phase, the app was tested. Some parts of the app followed TDD approach while many parts of the app were tested afterwards. Many types of tests were carried out into the system. From UI testing to Instrumented test and also Local Unit Test, all these tests were performed to check the app's performance, functionality, time handling, user's action and much more.

4.2.1 UI Testing

UI Testing is to verify whether the results from the actions performed by user are as expected or not. From opening an activity to check whether a button is clickable or not, all these test fall under UI test. For UI Test in the application Espresso was used. Some of the UI test done in the app are displayed below.

Test Case ID	Test Description	Text Function	Expected Result	Actual Result	Remarks
TC_1	Check if splash activity launches	public void isSplashedActivityLaunched() { ActivityScenario.launch(MainActivity.class); }	Splash Activity should launch	Splash Activity Launched	Pass
TC_2	Check if login and OTP confirmation activity launches	public void isLoginActivityLaunched() { ActivityScenario.launch(LoginUserActivity.class); ActivityScenario.launch(LogInOTPConfirmation.class); }	Login and OTP Confirmation activity should launch	Activities launched	Pass
TC_3	Check if register activity launches	public void isRegisterActivityLaunched(ActivityScenario.launch(RequestActivity.class); }	Register activity should launch	Register activities launched	Pass
TC_4	Check if menu activity launches	public void isMenuActivityLaunched() { ActivityScenario.launch(MainPageActivity.class); }	Menu activity should launch	Menu Activity launched	Pass

Table 4.1: Test Cases for Activity Isolation

Test Case ID	Test Description	Text Function	Expected Result	Actual Result	Remarks
TC_	Check if	public void	Should be	Is	Pass

5	login button and signup button redirect to its respective activity from main activity	chkLoginSignUpRedirection() {
		ActivityScenario.launch(MenuActivity.class);
		onView(withId(R.id.login_btn)).perform(click());
		ActivityScenario.launch(MenuActivity.class);
		onView(withId(R.id.request_btn)).perform(click());
		}
			redirected	redirected to respective activity	

TC_6	Check if pressing back from an activity redirect to previous activity	public void chkBackBtnRedirection() {
		ActivityScenario.launch(LoginUserActivity.class);
		onView(withId(R.id.back_Btn)).perform(click());
		}
			Should redirect	Is redirected to previous activity	Pass

TC_7	Check if back button from main activity or menu activity closes the app	public void chkExit() {
		ActivityScenario.launch(MainActivity.class);
		ViewActions.pressBack();
		}
			Should close the app	App Closes	Pass

Fig 4.2: UI test Case for Activity Navigation

4.2.2 Local Unit Test

A local test runs directly on your own workstation, rather than an Android device or emulator. As such, it uses your local Java Virtual Machine (JVM), rather than an Android device to run tests. Local tests enable you to evaluate your app's logic more quickly. However, not being able to interact with the Android framework creates a limitation in the types of tests you can run.

Chapter 5 Conclusion and Future Recommendations

5.1 Conclusion

My main goal/objectives were to create an application which store peoples license details and vehicle details which can be used to show to traffic police or make it secure. The current application has fulfilled these goals. I followed the specifications strictly but enhanced some of the features when there was need for it to be done. With the goals achieved the basis of the application and this project has been achieved. Building this project has been both fun and challenging for me since I got to learn many new things along the way while I enjoyed building it. As I came to the end of the project, I realized that there are many enhancements that can be made on the application. Many suggestions and new ideas came down the way when I gave my friends to test the application. But for now, I decided to follow the specification because there were realistic to achieve in this given amount of time. Any other enhancements to the application can be done in future development of the application.

5.2 Lesson Learnt/Outcome

After the completion of the project, a person doesn’t need to carry license or bluebooks as these can be accessed through this application. It will also allow owner to lend their bike or report for theft.

5.3Future Recommendations

Some of the things that can be added in this project in the near future are as follow:

a) This application is only available for Android devices right now, so in the near future this system can be made for IOS and even web platform,

b) We can only see the tax information right now but we cannot pay tax through the app, so in the near future this system can be enhanced which can pay the tax direct through this application.

c) Right now, admin has to manage the user directly in Realtime database; as web application can be used manage users by admin.

d) Almost all operations on this app runs on Main thread which can be improvised by making intensive tasks in the app run on background thread by applying the concepts of multithreading using RXJava and Coroutines.

e) Firebase doesn’t allow or provides many crucial features; hence the app’s own API can be made to tackle this problem.

f) Firebase doesn’t allow or provides many crucial features; hence the app’s own API can be made to tackle this problem.

References

[1] "Wikimedia," Wikimedia, [Online]. Available: https://commons.wikimedia.org/wiki/File:Waterfall_model_(1).svg.

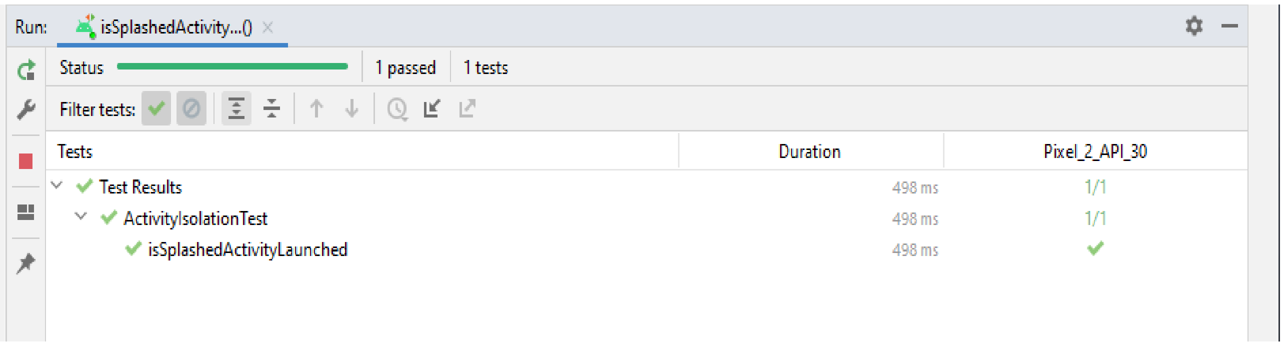
[2] "TutorialsPoint," [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.

[3] "DigiLocker," National eGoverance Division, [Online]. Available: https://www.digilocker.gov.in/.

[4] "Nagarik App," Government, [Online]. Available: https://nagarikapp.gov.np.

[5] "Draw.io," [Online]. Available: https://app.diagrams.net/.

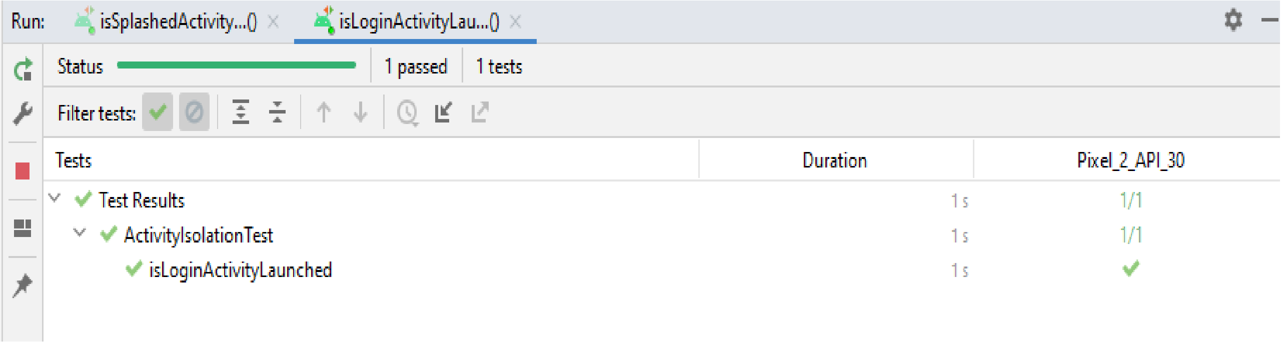
Appendix



Status 1 passed 1 tests

Tests	Duration	Pixel_2_API_30
✓ Test Results	498 ms	1/1
✓ ActivityIsolationTest	498 ms	1/1
✓ isSplashedActivityLaunched	498 ms	✓

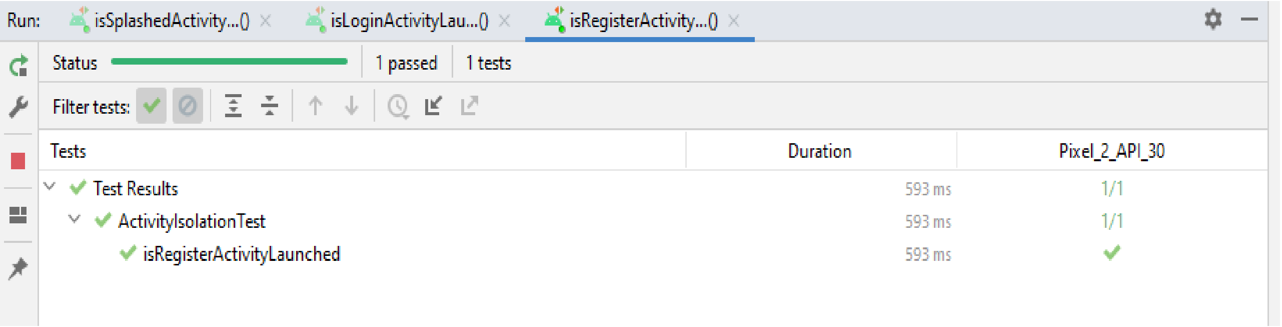
Figure TC_1



Status 1 passed 1 tests

Tests	Duration	Pixel_2_API_30
✓ Test Results	1 s	1/1
✓ ActivityIsolationTest	1 s	1/1
✓ isLoginActivityLaunched	1 s	✓

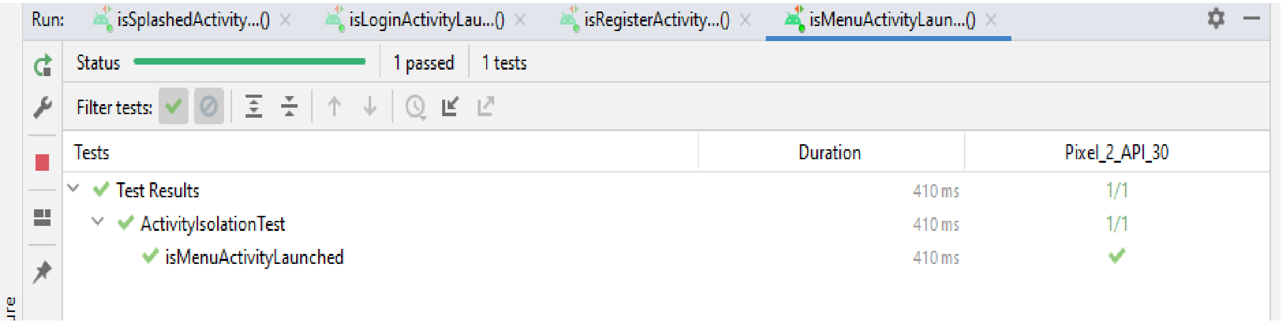
Figure TC_2



Status 1 passed 1 tests

Tests	Duration	Pixel_2_API_30
✓ Test Results	593 ms	1/1
✓ ActivityIsolationTest	593 ms	1/1
✓ isRegisterActivityLaunched	593 ms	✓

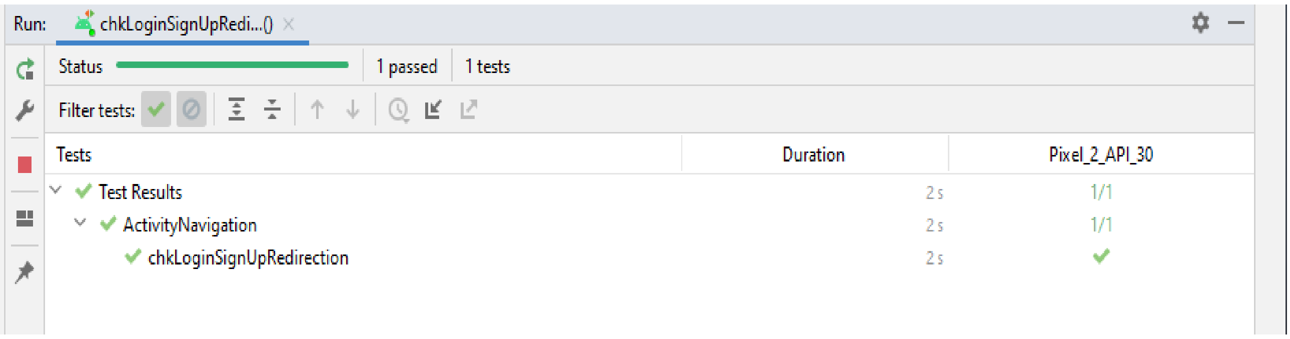
Figure TC_3



Status 1 passed 1 tests

Tests	Duration	Pixel_2_API_30
✓ Test Results	410 ms	1/1
✓ ActivityIsolationTest	410 ms	1/1
✓ isMenuActivityLaunched	410 ms	✓

Figure TC_4

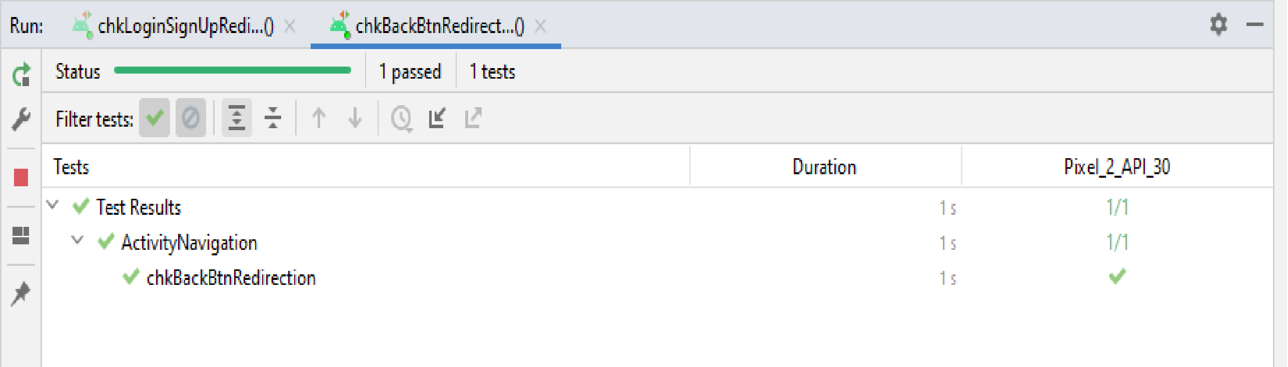


Run: chkLoginSignUpRedi...()

Status 1 passed 1 tests

Tests	Duration	Pixel_2_API_30
✓ Test Results	2 s	1/1
✓ ActivityNavigation	2 s	1/1
✓ chkLoginSignUpRedirection	2 s	✓

Figure TC_5

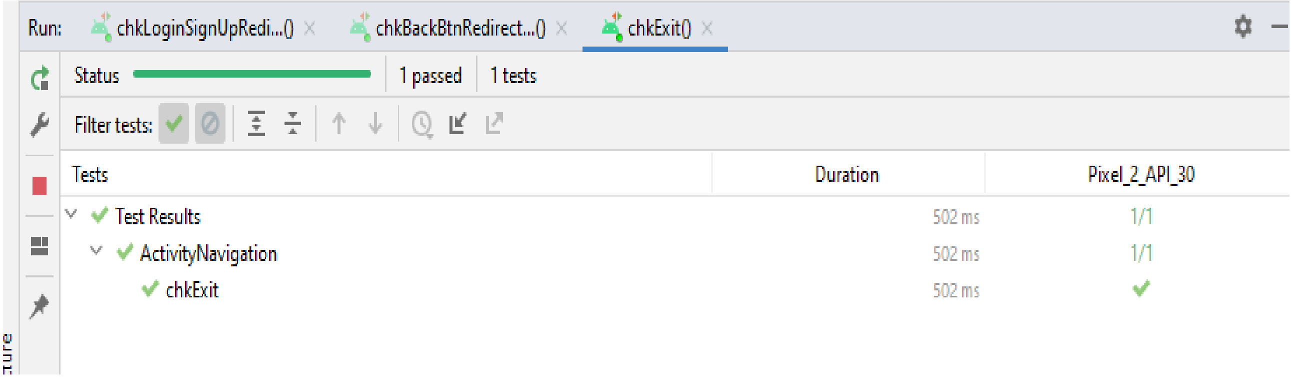


Run: chkLoginSignUpRedi...() chkBackBtnRedirect...()

Status 1 passed 1 tests

Tests	Duration	Pixel_2_API_30
✓ Test Results	1 s	1/1
✓ ActivityNavigation	1 s	1/1
✓ chkBackBtnRedirection	1 s	✓

Figure TC_6



Run: chkLoginSignUpRedi...() chkBackBtnRedirect...() chkExit()

Status 1 passed 1 tests

Tests	Duration	Pixel_2_API_30
✓ Test Results	502 ms	1/1
✓ ActivityNavigation	502 ms	1/1
✓ chkExit	502 ms	✓

Figure TC_7