



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B.Tech. CE Semester – VI

Subject: (CE-621) System Design Practice

Project Title:

Next Generation Toll System

By:

Raj Panchal (Roll No:CE074 ID:17CEUBG104)

Jaydeep Mahajan (Roll No:CE066 ID:17CEUOG098)

Guided By:

Prof. Jigar M. Pandya



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

CERTIFICATE

This is to certify that System Design Practice project entitled “Next Generation Toll System” is the bonafied report of work carried out by

1) Raj Panchal (17CEUBG104)

2) Jaydeep Mahajan (17CEUOG098)

Of Department of Computer Engineering, Semester VI, academic year 2019-20,
under our supervision and guidance.

Guide

Prof. Jigar M. Pandya

Assistant Professor of
Department of Computer Engineering
Dharmsinh Desai University, Nadiad.

Table of Contents

| | |
|---|----|
| 1. Abstract..... | 1 |
| 2. Introduction..... | 2 |
| 3. Software Requirements Specification..... | 3 |
| 4. Design..... | 5 |
| 5. Implementation..... | 13 |
| 6. Testing..... | 16 |
| 7. Screen-shots..... | 20 |
| 8. Conclusion..... | 22 |
| 9. Limitation and Future Extension..... | 23 |
| 10. Bibliography..... | 24 |

1.ABSTRACT

Today we all aware about FASTag is an electronic toll collection system in India, operated by the National Highway Authority of India (NHAI). It employs Radio Frequency Identification (RFID) technology for making toll payments directly from the prepaid or savings account linked to it or directly toll owner. It is affixed on the windscreen of the vehicle and enables to drive through toll plazas without stopping for transactions. The tag can be purchased from official Tag issuers or participating Banks and if it is linked to a prepaid account, then recharging or top-up can be as per requirement. As per NHAI, FASTag has unlimited validity. 7.5% cashback offers were also provided to promote the use of FASTag. Dedicated Lanes at some Toll plazas have been built for FASTag.

We just tried to upgrading version of this technology. It assumes that in user's car having one device installed. Before journey he could set the source and destination as his/her requirement then device shows the all the tollbooth which comes in journey and total amount that he has to pay for this trip. So, user can pay in advance via login into the system using credit card, debit card or takes another way to pay like whenever he crosses the toll booth automatically amount deducted from his account.

2. INTRODUCTION

2.1 Brief Introduction:

This project is going to build on subject “Next Generation Toll System”. In which we emphases on consuming less amount of time. An individual can get rid from long queue of tollbooth. Here we are going to develop virtual tollbooth between source & destination. We will try to build system that compatible with cars. Now a days, though FASTAG comes in picture, but some of us are wait for paying toll by standing in long queue and lose precious time. Because of this many disadvantages like wasting fuel, time for journey get exceed. A system using android auto android automotive OS technologies that helps to pay toll-tax automatically without any kind of problem based on car location.

Currently we have developed app for this kind of system using flutter framework, which is able to display source and destination and virtual tollbooth between them. It can help you to pay the tax before journey started or during the journey whenever toll comes.

- Advantages of System:
 - Convenience to Drivers/Consumers.
 - Saves Manpower cost.
 - Safer and more secure payment.
 - Time saver.

2.2 Tools/Technologies Used:

Technologies:

- ✓ Flutter – Framework
- ✓ Dart – Single codebase
- ✓ Firebase
- ✓ Google Map API for android SDK
- ✓ Google direction API (if possible)
- ✓ Payment Gateway API (Razorpay)

Tools:

- ✓ Microsoft Visual Code
- ✓ Android Studio
- ✓ Android Emulator
- ✓ Android Device & USB cable

3. SOFTWARE REQUIREMENTS SPECIFICATION

Types of Users :

- End User/Traveller

System function requirements:

R.1 End-User / Traveller

R.2.1 Register / Sign Up :

Input : Email and Password.

Output : Confirmation message.

R.2.2 Login / Sign In :

Input : Email and Password.

Output : Home Screen.

R.2.3 Select source and destination for trip :

Input : Source and destination.

Output : Put marker on source and destination.

R.2.4 View toll booths:

Input : User Selection .

Output : Display virtual tolbooths.

R.2.5 Edit trip details :

Input : Car type, Car no., Journey Date etc.

Output : Payment Page.

R.2.6 Calculate toll amount:

Input : toll between source and destination.

Output : Total Amount

R.2.7 Make Payment :

Input : Payment details.

Output : Payment success or Payment error message.

R.2.8 Display history :

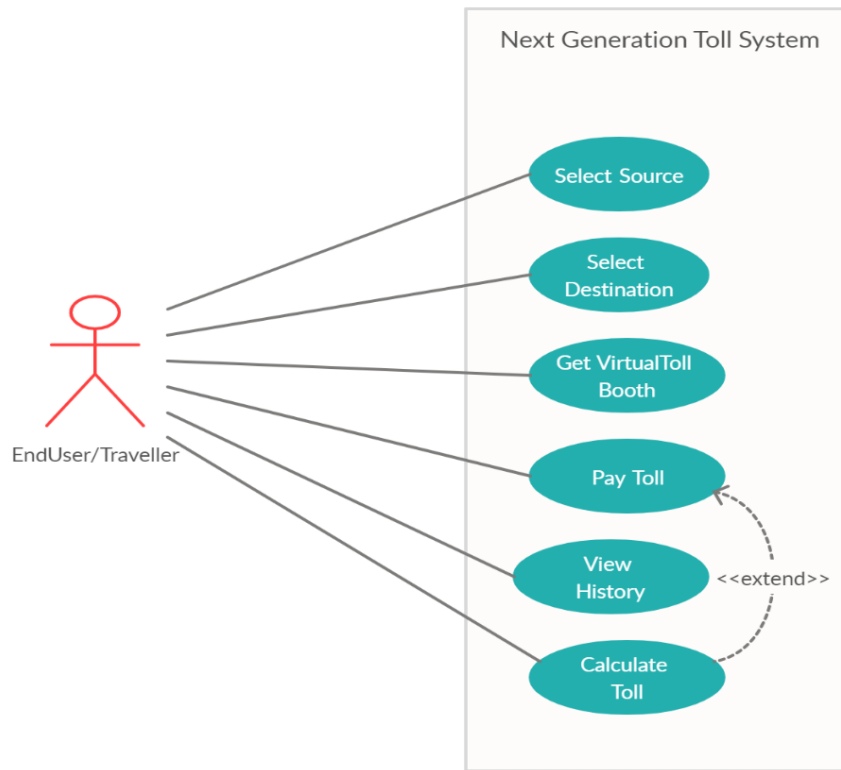
Description : One can show all previous transaction.

Input : User selection.

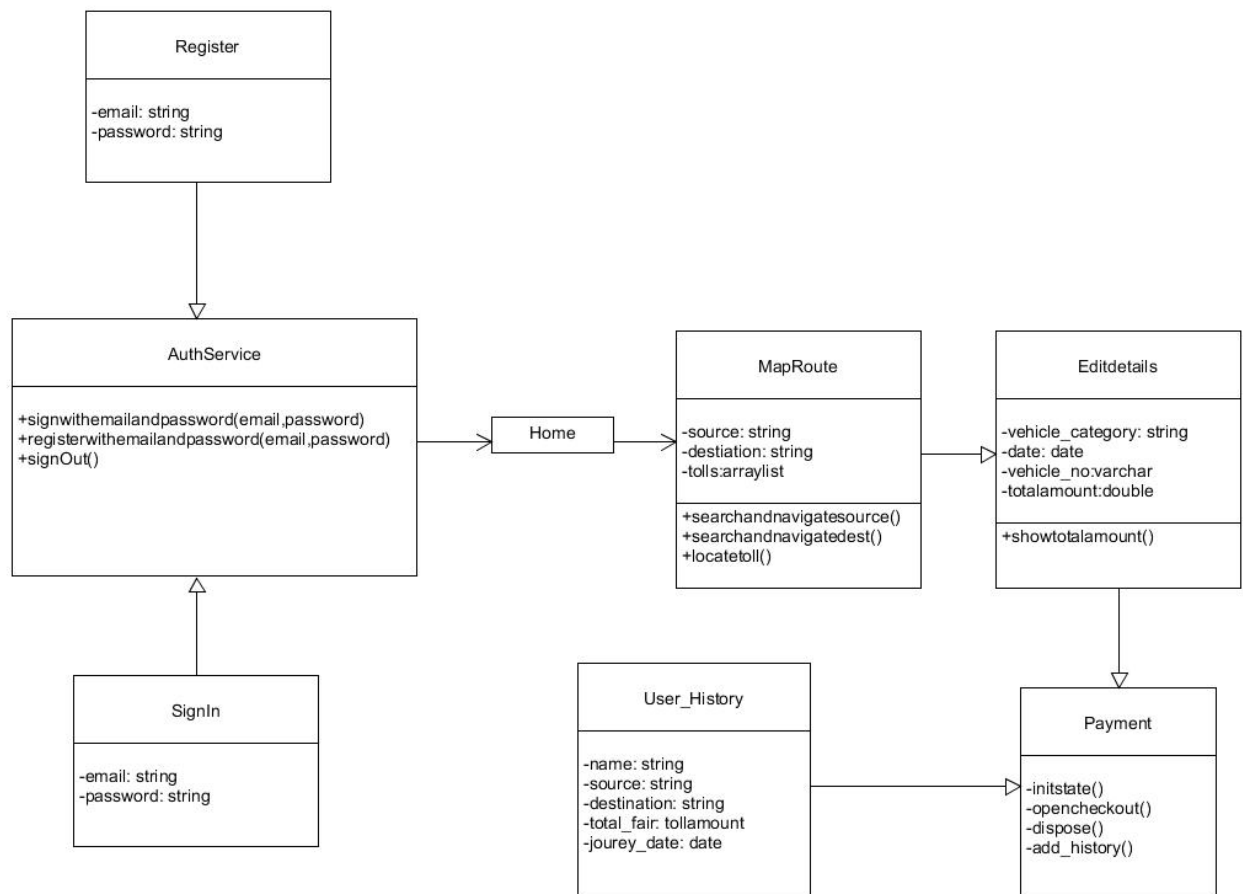
Output : List of all previous transaction.

4. DESIGN

Use case Diagram :

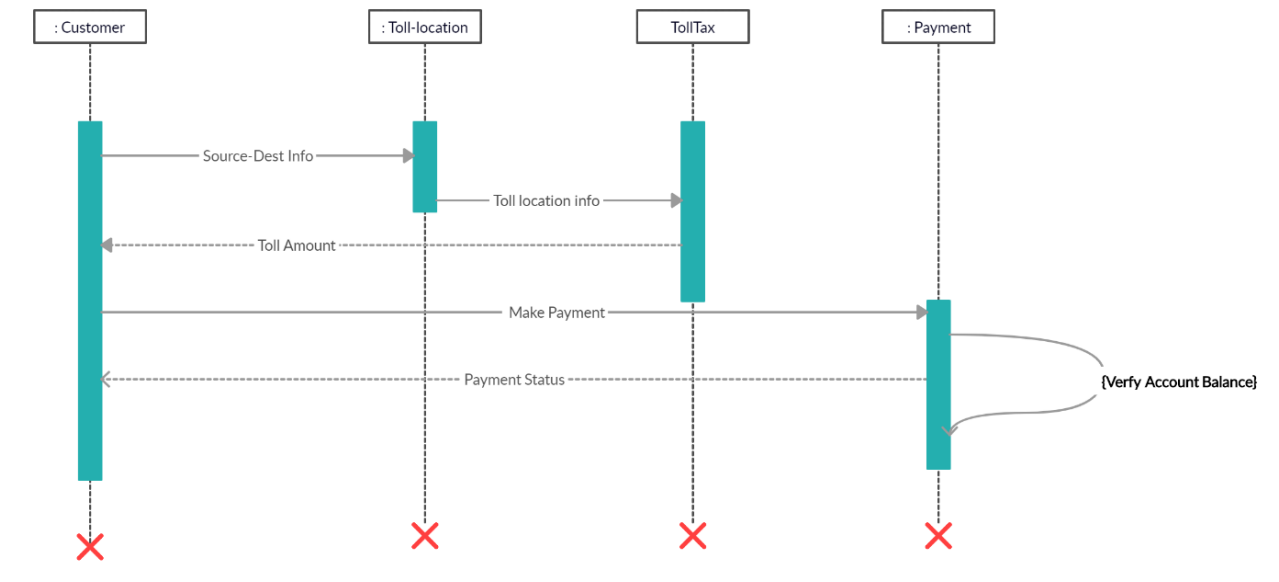


Class diagram :



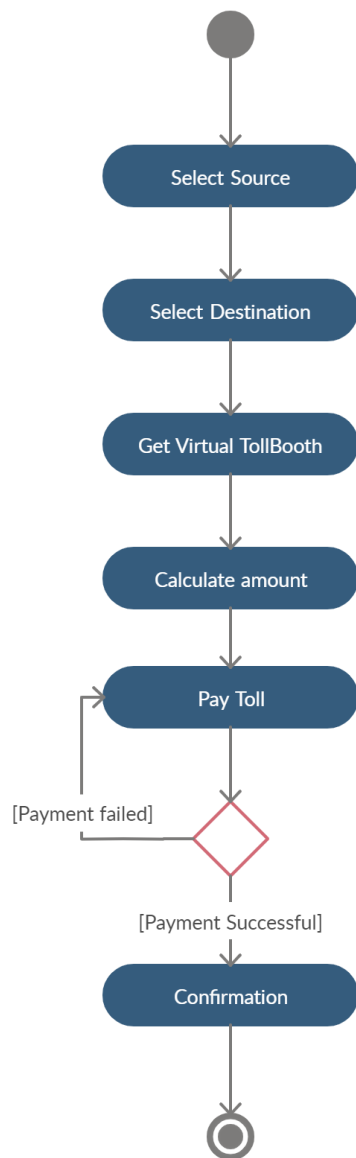
Sequence diagrams:

For Make-Payment :

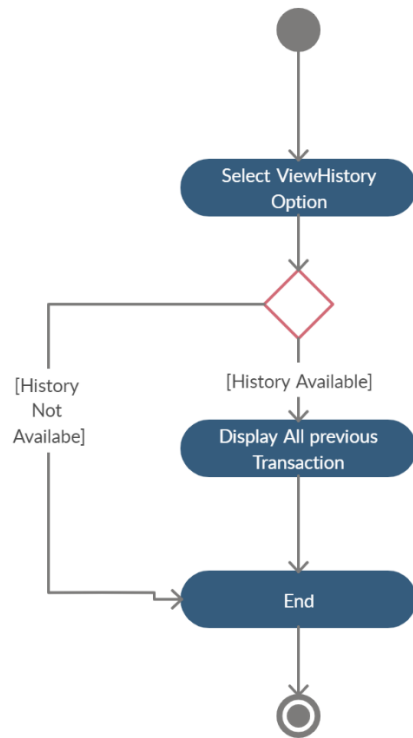


Activity diagrams:

For Tollpay:



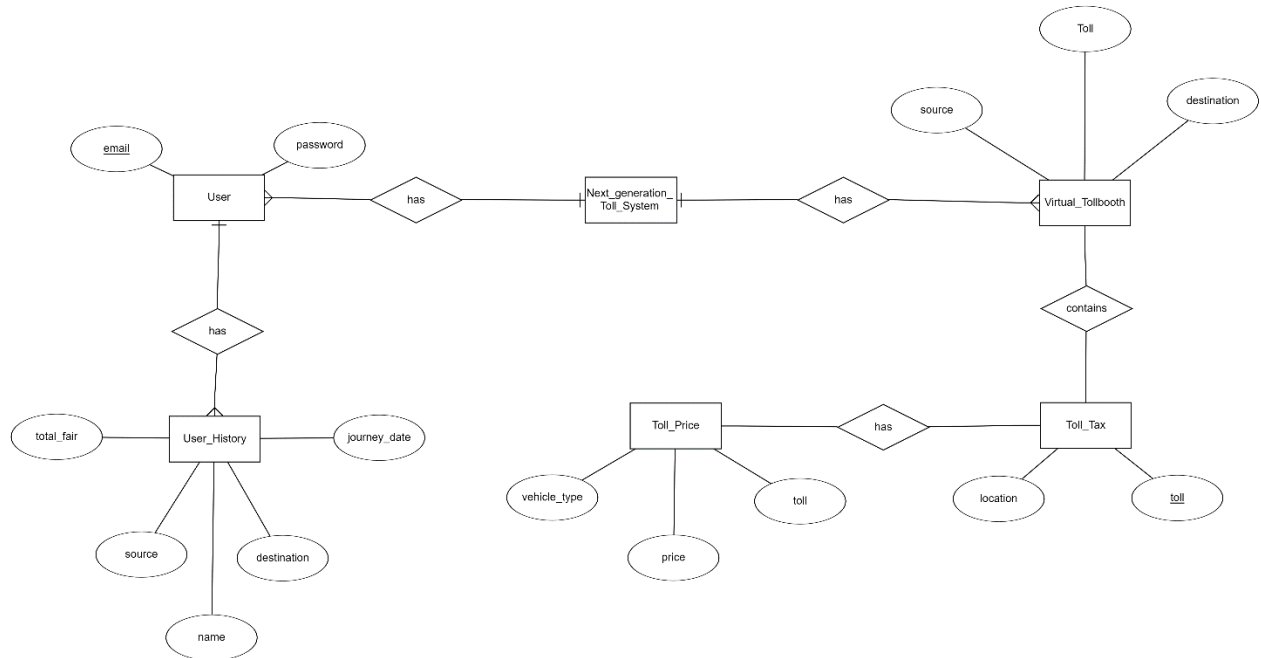
For view history:



State diagrams:

*State diagram is not applicable for this system because system doesn't contain any kind of state.

E-R Diagram:



Data dictionary :**Data Dictionary for Restaurant Billing System**

| User | | | | | |
|----------|--------------|----------|--------|-------|-----------------|
| Field | Type | Required | Unique | PK/FK | Reference Table |
| email | varchar (30) | Yes | Yes | PK | - |
| password | varchar (50) | Yes | No | - | - |

| Virtual_Tollbooth | | | | | |
|-------------------|--------------|----------|--------|-------|-----------------|
| Field | Type | Required | Unique | PK/FK | Reference Table |
| source | varchar (20) | Yes | No | - | - |
| destination | varchar (20) | Yes | No | - | - |
| toll | array | Yes | Yes | PK | - |

| Toll_Tax | | | | | |
|----------|--------------|----------|--------|-------|-----------------|
| Field | Type | Required | Unique | PK/FK | Reference Table |
| toll | varchar (10) | Yes | Yes | PK | - |
| location | geopoint | Yes | No | - | - |

| Toll_Price | | | | | |
|--------------|--------------|----------|--------|-------|-----------------|
| Field | Type | Required | Unique | PK/FK | Reference Table |
| toll | varchar (10) | Yes | Yes | FK | Toll_Tax |
| price | float | Yes | No | - | - |
| vehicle_type | varchar (10) | Yes | No | - | - |

| User_History | | | | | |
|--------------|--------------|----------|--------|-------|-----------------|
| Field | Type | Required | Unique | PK/FK | Reference Table |
| name | varchar (30) | Yes | No | - | - |
| source | varchar (50) | Yes | No | - | - |
| destination | varchar (50) | Yes | No | - | - |
| journey_date | date | Yes | No | - | - |
| total_fair | float | Yes | No | - | - |

5. IMPLEMENTATION DETAIL

I). Modules

➤ Main User Module :

This module contains the main homepage of the system which provides basic information about the website. It contains different functionalities like view history, pay toll etc. This module does require authentication.

➤ Login Module :

This module takes user credentials and then verifies it with registered users, if user have entered incorrect credentials then alert with "Incorrect details" , otherwise user will be redirected to the home page.

➤ Registration Module :

This module is used to create the account on system. All fields are containing some types of validations. Then user can login to our system using his/her credentials.

➤ Map route Module :

This module allows driver to set source and destination , it helps to display tollbooth between two points and based on car type and finds total toll amount of journey. It has been implemented by use of Maps SDK for Android and geo-location API.

➤ Payment Module :

This module is build for pay the toll by using Razorpay API. Razorpay is a payment gateway service. It accept and validate internet payments via credit card, debit card, Net banking modes, also other wallets like Paytm , PhonePe, fast charge.

➤ Previous transaction Module :

This module is displaying all previous transaction for user of the system. It display all information regarding previous journey.

II). Function prototypes

➤ Login :

```
Future signInwithemailandpassword(String email, String password)
```

➤ Register :

```
Future registerwithemailandpassword(String email,String password)
```

➤ Load google map :

```
void _onMapCreated(GoogleMapController controller)
```

➤ Set source & destination :

```
searchandnavigatesource()
```

```
searchandnavigatedest()
```

➤ Mark tollbooth :

```
tollmark(toll,tolllength)
```

➤ Payment handle methods :

```
void _handlePaymentSuccess(PaymentSuccessResponse response)
```

```
void _handlePaymentError(PaymentFailureResponse response)
```

```
void _handleExternalwallet(ExternalWalletResponse response)
```

- Add history in database:

```
Future <void> addhistory()
```

- Show toll amount :

```
showtollamount() async
```

- Sign out :

```
Future signOut() async
```

6. TESTING

In this system we have used Black Box testing.

The main focus of black box testing is on the validation of your functional requirements.

Here are the generic steps followed to carry out any type of Black Box Testing.

- ✓ Initially, the requirements and specifications of the system are examined.
- ✓ Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- ✓ Tester determines expected outputs for all those inputs.
- ✓ Software tester constructs test cases with the selected inputs.
- ✓ The test cases are executed.
- ✓ Software tester compares the actual outputs with the expected outputs.
- ✓ Defects if any are fixed and re-tested.

Different test cases :

For Register & Login :

If user doesn't enter valid information then system not allowed for register in the app. It displays error message "INVALID USERNAME OR PASSWORD".

If user doesn't provide valid details at login time system displays error message like "Enter Valid Email" , "Enter a Password 6+ chars long."

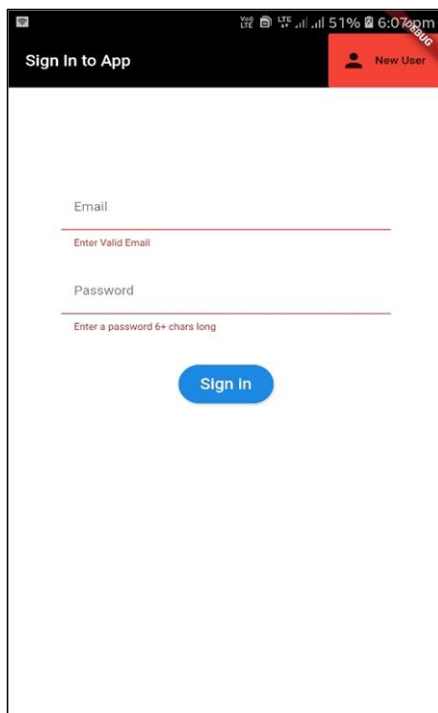
For source & destination between tollbooths :

If user enter source then pin pointed to that place and as destination then click on the icon that shows the virtual toll booths between based on source and destination.

For add details of journey :

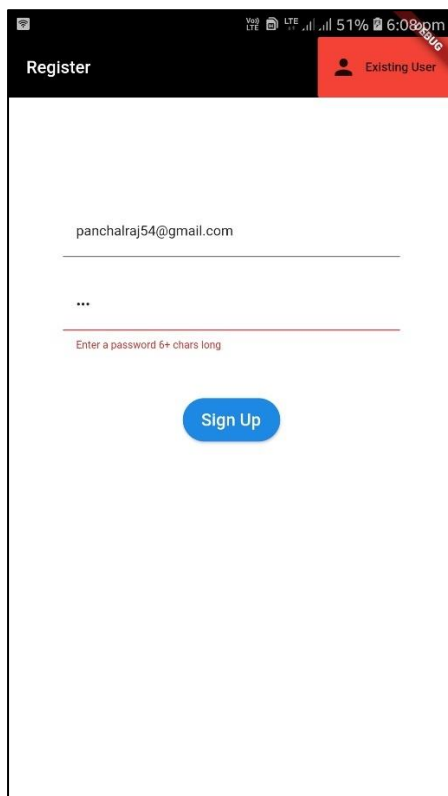
If user enter all details correctly only then he allows to pay otherwise it display validation message.

For Sign in:



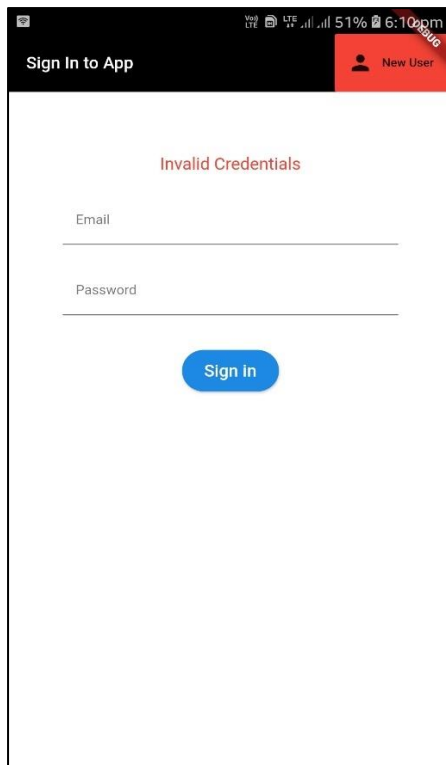
A mobile app interface for signing in. The status bar at the top shows 'LTE', signal strength, 51% battery, and 6:07pm. The app header is black with 'Sign In to App' in white and a red button labeled 'New User' with a person icon. The main area is white with two input fields: 'Email' with a red underline and the hint 'Enter Valid Email', and 'Password' with a red underline and the hint 'Enter a password 6+ chars long'. A blue 'Sign In' button is centered below the fields.

For Register:

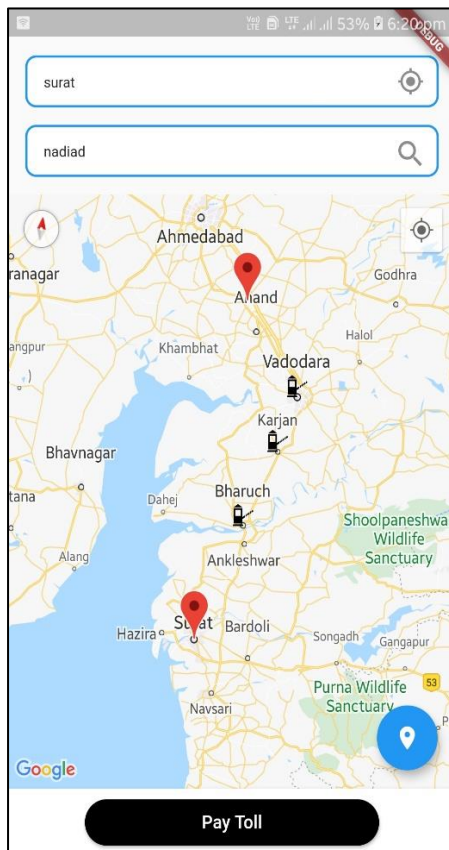


A mobile app interface for registering. The status bar at the top shows 'LTE', signal strength, 51% battery, and 6:08pm. The app header is black with 'Register' in white and a red button labeled 'Existing User' with a person icon. The main area is white with two input fields: the first contains 'panchalraj54@gmail.com' with a red underline, and the second contains '***' with a red underline and the hint 'Enter a password 6+ chars long'. A blue 'Sign Up' button is centered below the fields.

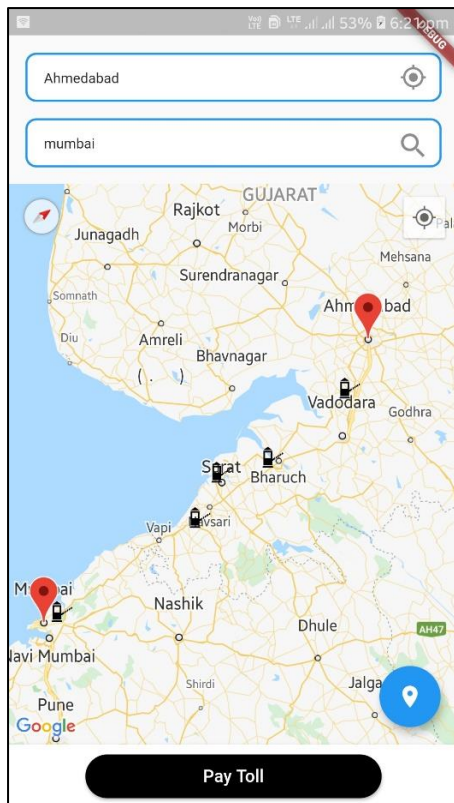
For Wrong Credentials:



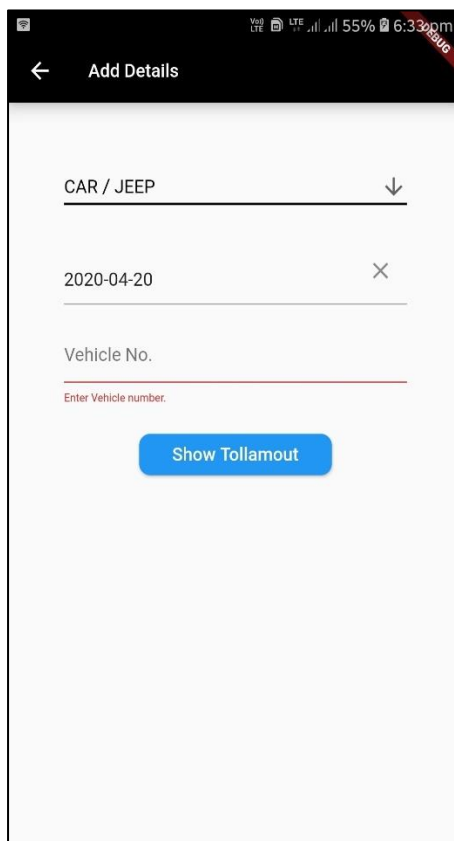
For source : Surat & destination : Nadiad



For source : Ahmedabad & destination : Mumbai

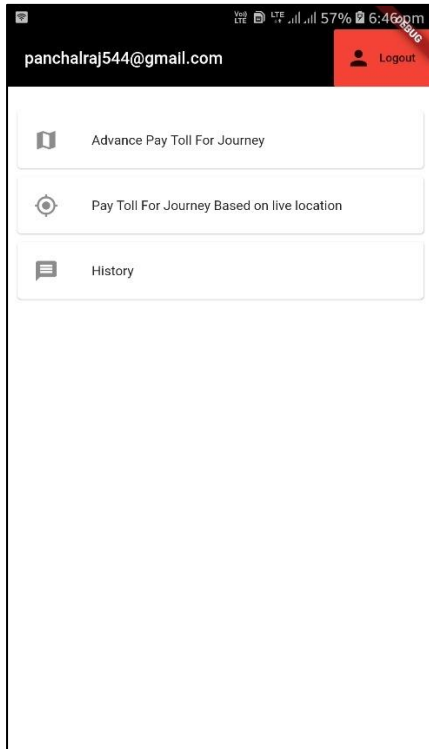


For Add details of journey :

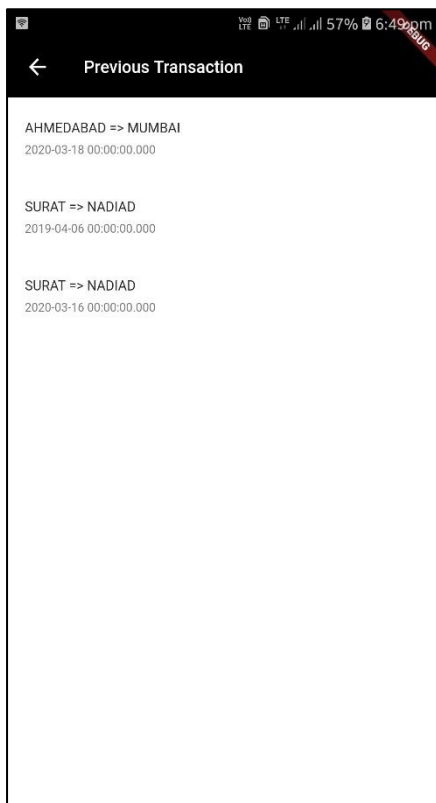


7. SCREEN-SHOTS

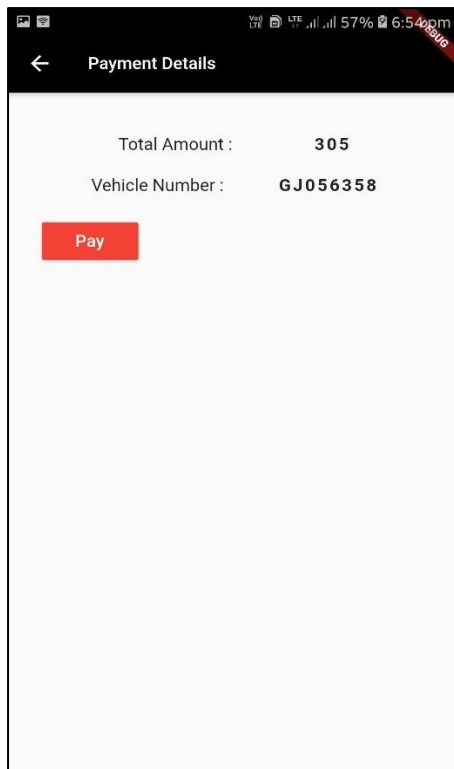
Home screen for user:



Previous transaction page :



For payment Details:



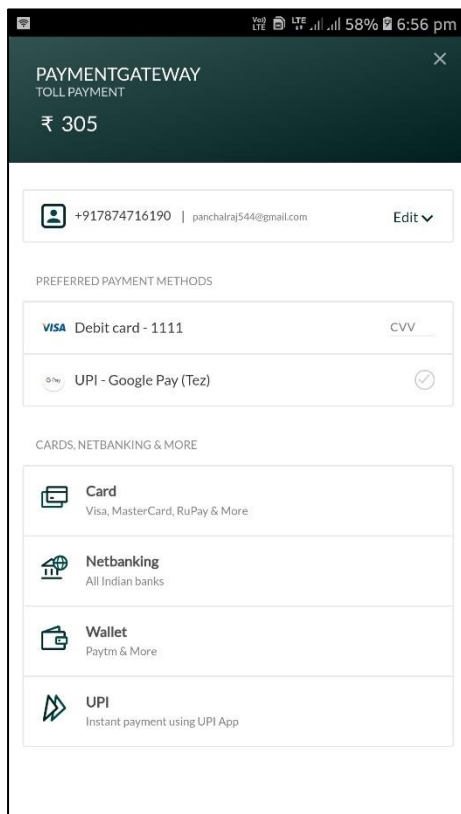
Payment Details

Total Amount : 305

Vehicle Number : GJ056358

Pay

For payment gateway:



PAYMENTGATEWAY
TOLL PAYMENT

₹ 305

+917874716190 | panchalraj544@gmail.com Edit

PREFERRED PAYMENT METHODS

VISA Debit card - 1111 CVV

UPI - Google Pay (Tez)

CARDS, NETBANKING & MORE

Card
Visa, MasterCard, RuPay & More

Netbanking
All Indian banks

Wallet
Paytm & More

UPI
Instant payment using UPI App

8. CONCLUSION

Functionality Successfully Implemented :

- Sign in / Login
- Sign Up / Register
- Set source and destination
- Get all virtual tollbooths between source and destination
- Calculate toll amount according to requirement
- Payment method
- View history
- Add details of journey

Above all functionality has been successfully implemented by us.

9. LIMITATION AND FUTURE EXTENSION

Limitation of System :

This system only provides facility of pay toll in advanced and view history. This system doesn't display routes between source and destination because for this task direction API necessary but it is not available free so we can't able to use it.

Also, we have not implemented functionality that provides facility to pay toll based on live location because of some limitation. System display only virtual toll booth not original located toll booth. How to get original located toll booth that is question for us.

We want to fit this system in car but we have currently only implemented as mobile app. It is not possible to implement system in all cars. So, if system is not including in car then user need to follow original process.

Functionality which was not implemented:

- Pay toll based on live location:
This means that whenever toll comes in journey amount automatically deducted from individual user's or traveller's account.
- Add virtual tollbooth:
Though we add virtual location from directly through the database we haven't implement functionality that add virtual toll booth directly through the app.
- Monitoring user activity :
We did not implement functionality that track the record of user's all activities like log In, log out , transaction record , search record etc.

Possible future extension to project:

In future we want to remove limitation from project that we have described above and also,

- Try to build better optimized system.
- Setup the system that can be used in as many cars as possible.

10. BIBLIOGRAPHY

Reference Link :

- ✓ <https://console.firebase.google.com>
- ✓ <https://dashboard.razorpay.com/#/app>
- ✓ <https://cloud.google.com/docs/authentication/api-keys>
- ✓ <https://www.youtube.com/>
- ✓ <https://medium.com/@rajesh.muthyala/flutter-with-google-maps-and-google-place-85ccee3f0371>
- ✓ <https://pub.dev/>
- ✓ <https://github.com/>
- ✓ <https://console.cloud.google.com/home/dashboard?project=tollmap-266314>