TRAVEL EASY

A PROJECT REPORT

Submitted by

Rajashree S 2018115085

Arthi A 2018115016

Priya J 2018115078

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

MONTH YEAR

ANNA UNIVERSITY

CHENNAI - 600 025

BONA FIDE CERTIFICATE

Certified that this project report titled TRAVEL EASY is the bona fide work of RAJASHREE S,ARTHI A,PRIYA J who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:
DATE:

DR.S.BAMA
ASSISTANT PROFESSOR
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

Dr. SASWATI MUKHERJEE
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

TRAVEL LITE is a mobile application where customers can buy tickets via mobile using Qr code and helps to make social distancing possible in bus . Every ticket purchased will be given QR code for verification process.

The QR code is essential during the boarding process where customer need to show their QR code before on board to the bus.The Otp(one time password) will be sent to the User's registered mobile number and from that otp,QR code gets generated accordingly.Then, Scanner in the bus scan customer's QR code to update the availability of the passenger.

The seats will be booked and the Qr code will be generated only after count checking where crowd monitoring is done to ensure safety as well as to check the availability of the seats.

The objective of our project is to develop an application that utilizes QR-Code that supports express bus integrated e-ticket and to improves the management process in generating reports easily and systematically.

And Make passengers to travel more comfortably with peace environment.This new system will support the eco green campaign that urged by the government with using less paper.

ACKNOWLEDGEMENT

Acknowledgement should be brief and should not exceed one page when typed in one and a half line spacing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

OTP	One Time Password
SHA	Secure Hash Algorithm
JSON	JavaScript Object Notation
XML	Extensible Markup Language
UI	User Interface
SDK	Software Development Kit

CHAPTER 1

INTRODUCTION

Travel Easy is a mobile application where customers can buy tickets via mobile using QR code, Every ticket purchased will be given QR code for verification process. QR code must be shown to the conductor, Conductor scans the code to update the availability.

1.1 PURPOSE

We have chosen this project because this new system will support the eco-green campaign that urged by the government with using less paper and also this makes customer to know the availability and buy tickets online. It also reduces the contact between customer and conductors which is more helpful for our current situation.[**?**].

1.2 MODULES

We have 5 Modules in our project

1.2.1 USER

User Module is the one where the customers are able to check their availability of buses and seats and able to book the tickets online.

1.2.2 ADMIN

Admin Module will basically add buses for the respective driver. Admin can also delete,update the details of bus and Admin will be able to send the mail for the respective driver.

1.2.3 DRIVER

In Driver Module, the drivers will be able to login and check their work for the day.

1.2.4 QR SCANNER

Qr Scanner will be able to scan the Qr code generated after the payment of the customers for their tickets

1.2.5 QR CODE GENERATOR

Qr Code Generator generates the Qr code when the customer enters their OTP

1.3 PLATFORM

Our mobile application is developed in android studio using Java and XML. This is compatible for android users. We are using Firebase as Backend.Firebase is Google's mobile platform which enables us to create high quality apps easily.

CHAPTER 2

EXISTING WORK

We have gone through interesting thesis, websites and other media contents for developing this project and our knowledge. The citation of those are given in this chapter as refernces.

We have also discussed the limitations of the existing work and how our idea is advantageous over them in this chapter.

2.1 LIMITATIONS OF EXISTING PROJECT

There are some mobile applications which are partially similar to our mobile application. Redbus,MakemyTrip,TNSTC. These are the applications which allows the user to book the tickets for their trips.Our project is designed mainly for the government buses for which no specific mobile application has been introduced so far.

Also, In day-to-day travelling,the system will not provide facility to buy the ticket online by means of credit card/ Net banking. User has to pay the money manually from the conductor for getting the ticket

2.2 ADVANTAGES OVER THE EXISTING PROJECT

i) Our Ultimate goal is for the people who are dependent on the government buses everyday.

ii) We simplify their process of rushing and taking the tickets as booking the tickets using a mobile phone.

iii) They need not have to move inside the bus for getting their tickets.

iv) Also we have crowd monitoring which stops the booking when a limited number of people had already booked.

v) So it is also helpful for the people to maintain social distancing inside the bus.

2.3 METHOD OF CITATION

Following are a few methods of citation.

2.3.1 Citation of Websites

We followed many websites to get various information regarding firebase .Their references are given as reference. For QR generation[1] , QR scanner[2] , OTP Authentication[3] , Firebase[4] and [5].

2.3.2 Citation of Journals

Method for citing journals can be seen in References [6] and [7] .

CHAPTER 3

DESIGN OF YOUR WORK

We have several modules associated with our project. They are discussed in this chapter.

3.1 MODULES

Basically,it consists of three modules which are listed below:

1) User

2)Admin

3)Driver

3.1.1 USER MODULE

The User has to register into our travel easy app if he is a new user. After the registration, User can enter his current location and his destination to check buses. After entering it he/she will be able to check the number of buses taking his/her route and can check the availability of seats. If there is a seat available for the user, he/she can able to book the ticket and make their payment through our app.

After the payment the user will get an OTP to his/her registered mobile phone. The User will be entering that OTP in our app to get his/her

Qr Code. Before getting into the bus user should scan his/her Qr code using the Qr Scanner available in the bus and can enter the bus.

3.1.2 DRIVER MODULE

In this module, the driver will fill his information and gets in into our app using the id and password he was given during registration. Then he will be able to see his trip for the day (bus number, route, and timing) that the admin allocated to him.

3.1.3 ADMIN MODULE

The Admin will be signed in first into the app. And the admin plays the major role in our app. Admin is the person who will be able to add the buses for the corresponding routes and he assigns a driver for the respective buses that he had added. In addition the admin also sends the e-mail for the respective driver about his work of the day. This information can be viewed by the Driver when he gets in from his side.

3.2 OTHER FEATURES

3.2.1 OTP AUTHENTICATION

The user must enter his or her mobile number after completing the booking and payment process in the OTP module, and then the otp number will be sent to their respective mobile number.

3.2.2 QR CODE

The user should enter his or her otp number, which is sent to his or her mobile number as mentioned above and the QR code will be created after entering the otp number.

3.2.3 QR SCANNER

The user can scan the QR code generated using the scanner available in the bus . He/she is allowed to board the bus only after scanning the QR code.

3.3 WORKFLOW

Figure 3.1

Figure 3.2

Figure 3.3

CHAPTER 4

IMPLEMENTATION OF YOUR WORK

4.1 SEAT BOOKING

To Avoid confusion in seat booking,we use code synchronization which restricts multiple threads to work on the same code at the same time.The code without synchronization confuses users and can have negative consequences for data consistency.

A race condition occurs when two or more threads or processes read or write the same data, and the outcome is determined by the threads' scheduling. Unpredictable outcomes and subtle software bugs can occur where race conditions exist. To prevent this, threads should be limited to acting in a queue.

Code synchronisation prevents several threads from operating on the same piece of code at the same time. The locking principle underpins the synchronisation process.Threads working on synchronised code should get the lock on the object or class file, as necessary.

If several locks are present, separate threads would have the ability to modify the data at the same time.As two threads operate on the two methods at the same time, they will be able to access the same data. As a result, the locks were implemented at the object level rather than at the process level, taking into account the data manipulation.

When one thread locks an object and another attempts to call a synchronised method on it, the second thread will block until the object is unlocked.So,Finally

code synchronisation has prevented both threads from entering the bookTickets Method at the same time, preventing the amount of tickets available for both of them from being displayed at the same time.

4.2 PLATFORM AND PROGRAMMING LANGUAGE

We developed our mobile application in the Java language using the Android Software Development Kit.

Based on IntelliJ IDEA, Android Studio is the official Integrated Development Environment (IDE) for Android app development.Android Studio, in addition to IntelliJ's strong code editor and development tools, includes additional features that help us build Android apps faster, such as flexible Gradle-based build system and a unified development environment for all Android devices.

Initially, Java was the official language for developing Android apps (though it has since been superseded by Kotlin), and as a result, it is also the most widely utilised. Java is used in a lot of the apps in the Play Store, and it's also Google's most popular programming language. In addition, in the event that any problems arise, Java has a fantastic online community to help.

4.3 SETTING UP FIREBASE REALTIME DATABASE

We used firebase as the backend of our project as it is excellent for real-time data, and you can simply handle it because to Google's numerous APIs.Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

we first built a Firebase project to connect to our Android app before we integrated it with Firebase.After creating it,we registered our app by giving package name which will be mentioned in the androidManifest.xml file and also by giving debug signing certificate SHA-1.we need to give SHA-1 certificate key only when we use authentication for our application like phone number verification, e-mail verification,etc.,The authentication here indicate the OTP which will be sent to the registered mobile number after successful booking.

To add a Firebase configuration ,download google-services.json and move it to the module directory of the project.To enable Firebase products in our app,we added the following plugin:

classpath 'com.google.gms:google-services:4.3.8'

We added following firebase SDK's to declare the desired dependencies:

i)implementation platform('com.google.firebase:firebase-bom:28.0.1')

ii) implementation 'com.google.firebase:firebase-analytics'

The following dependency is added to ensure firebase authentication:

implementation 'com.google.firebase:firebase-auth:11.8.0'

After giving all the packages and dependencies,sync the app with the firebase.On successful completion,the gradle files get synched into the project without any errors.

To implement firebase in the code,we used DatabaseReference.In order to connect the DatabaseReference with the firebase database, we need

to get the reference from FirebaseDatbase using getInstance() method.Once any update happens at a certain node,we will receive a DataSnapshot..Data from a Firebase Database location is stored in a DataSnapshot object. When we read data from a database, we get it as a DataSnapshot.They are immutable copies of data at a Firebase Database location that have been efficiently constructed. They can't and won't be changed.

4.4 USER INTERFACE

To design the UI,we used XML layout to make it much more neat and presentable.The widget package includes (mainly visual) UI elements for the Application's screen.

The actual UI of our application is defined by layout xml files. It contains all of the pieces (views) or tools that we intend to employ in our programme. TextViews, Button,Navigation View,Recycler View and other UI elements are examples.

Manifest xml File is used to specify all of our application's components. It contains the names of our application packages, activities, receivers, services, and the permissions required by our application like Access network state,Internet and Access wifi state.

Like this,there are numerous files such as styles.xml,colors.xml,strings.xml and drwables which supports to build an attractive UI for our application.

4.5 SCREENSHOTS

Our app screenshots are added in the below subsections.

4.5.1 ADMIN

Figure 4.1: splashscreen

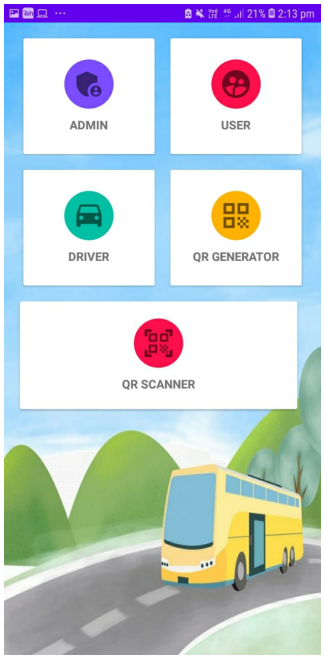

Figure 4.2: Home

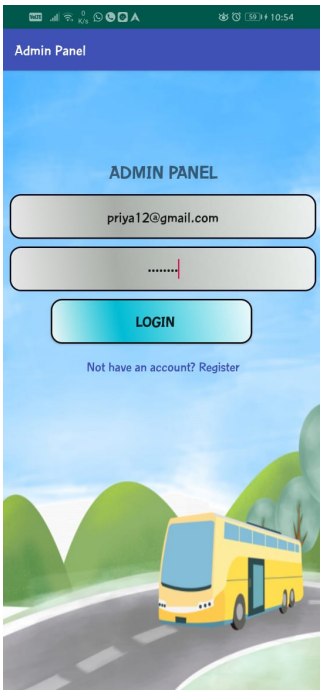

Figure 4.3: Login

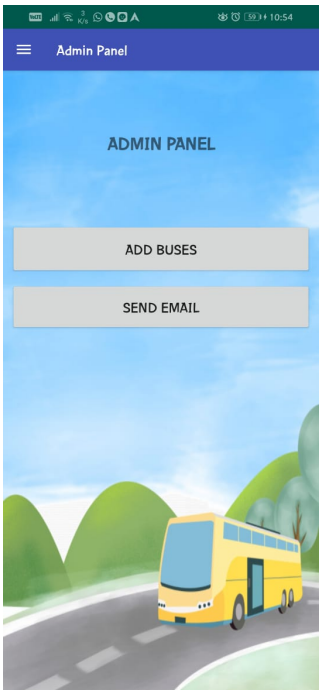

Figure 4.4: Add Buses

Figure 4.5: Bus Details

Figure 4.6: BusList

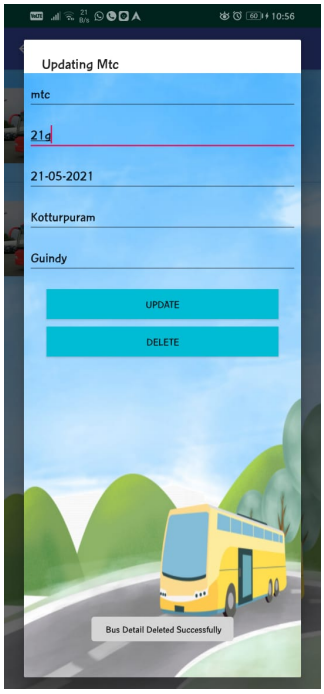

Figure 4.7: Update

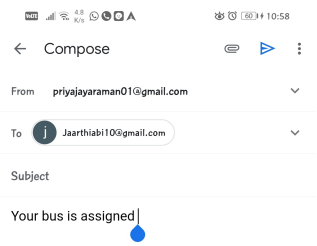

Figure 4.8: SendingMail

4.5.2 USER

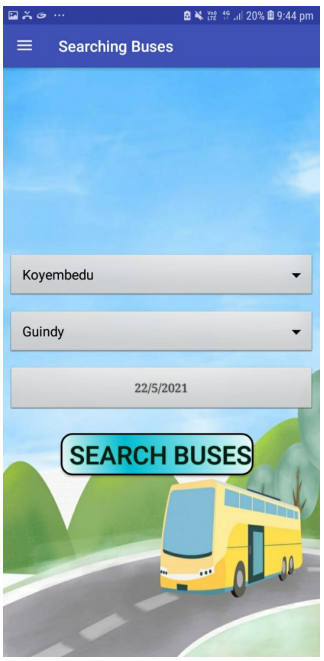

Figure 4.9: SearchBus

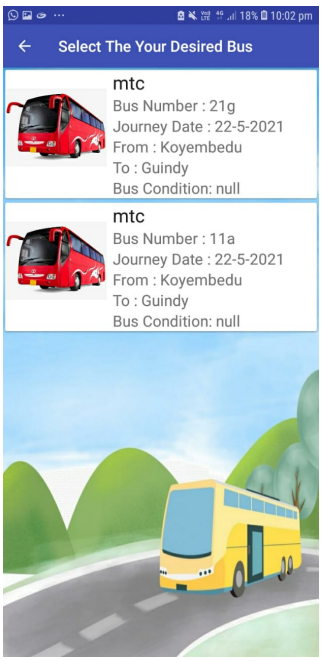

Figure 4.10: BusList

Figure 4.11: SeatBooking

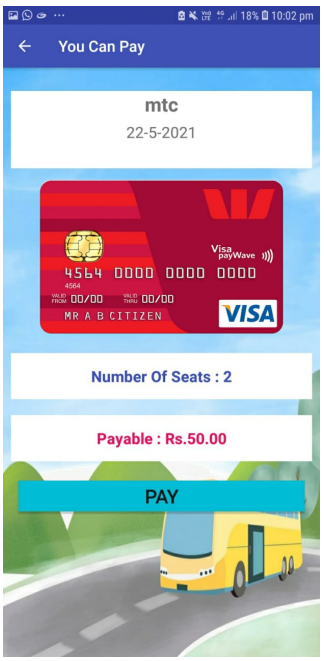

Figure 4.12: Cost

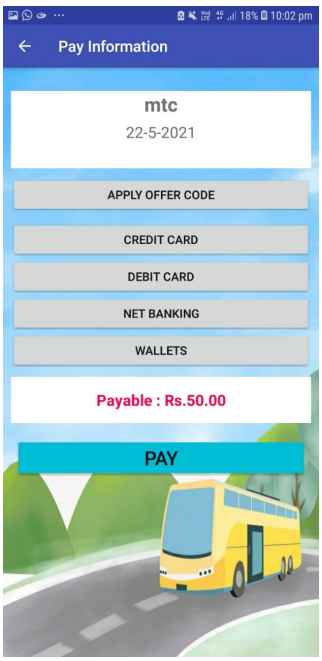

Figure 4.13: Payment

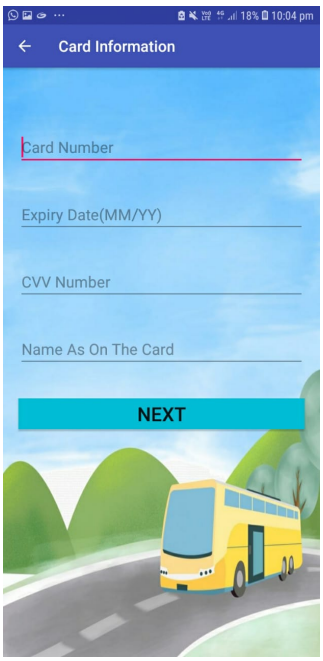

Figure 4.14: CardDetails

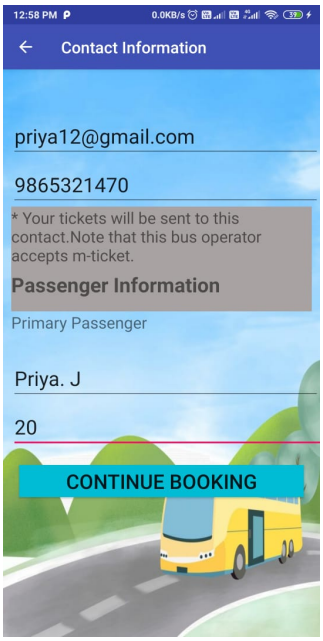

Figure 4.15: Contact

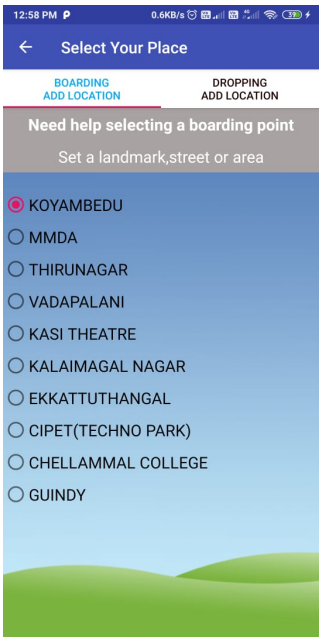

Figure 4.16: Pickup

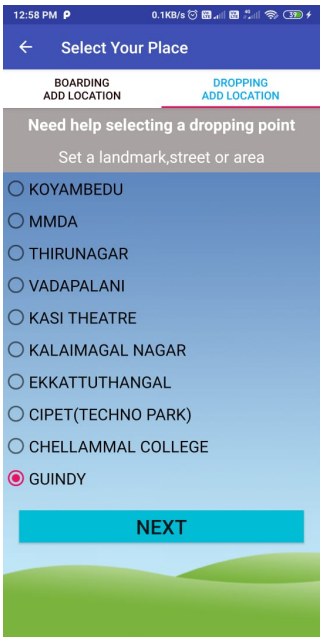

Figure 4.17: Drop

Figure 4.18: Booked

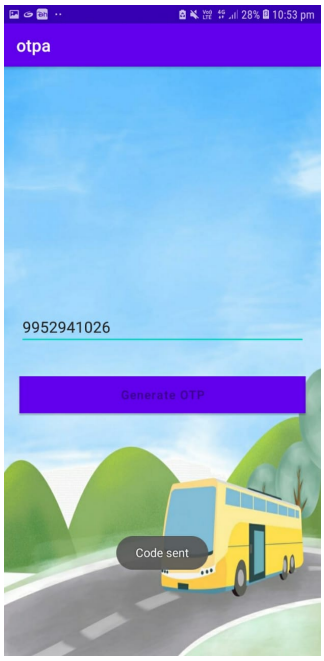

Figure 4.19: OtpSent

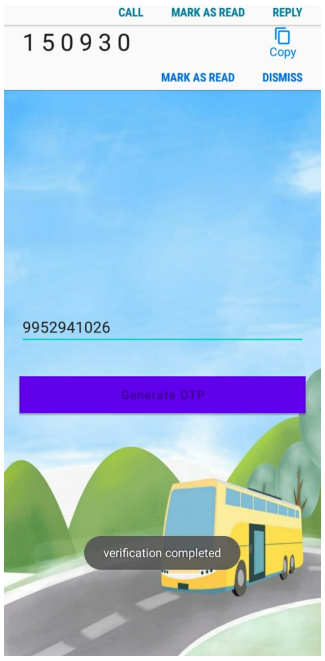

Figure 4.20: OtpReceived

4.5.3 DRIVER

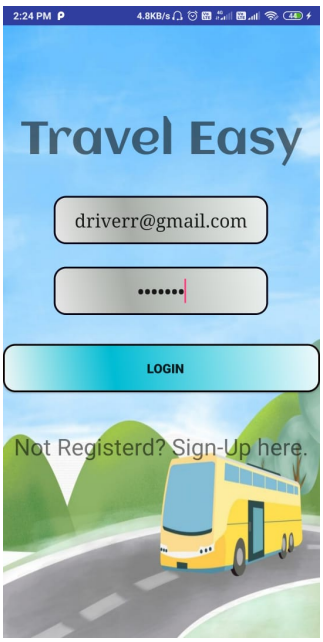

Figure 4.21: Login

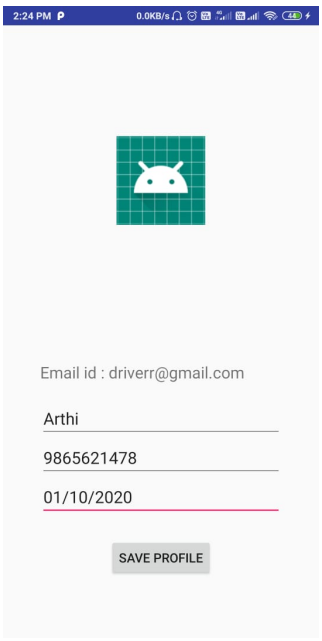

Figure 4.22: Profile

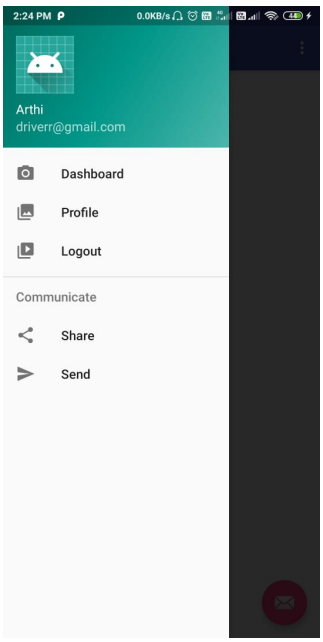

Figure 4.23: DashBoard

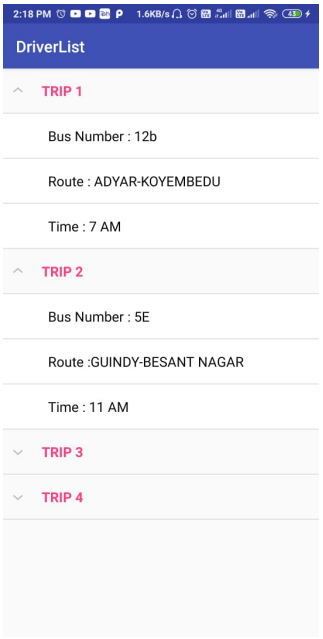

Figure 4.24: TripList-1

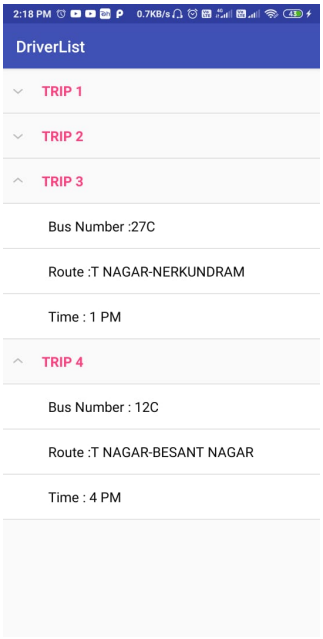

Figure 4.25: TripList-2

4.5.4 QRCODE

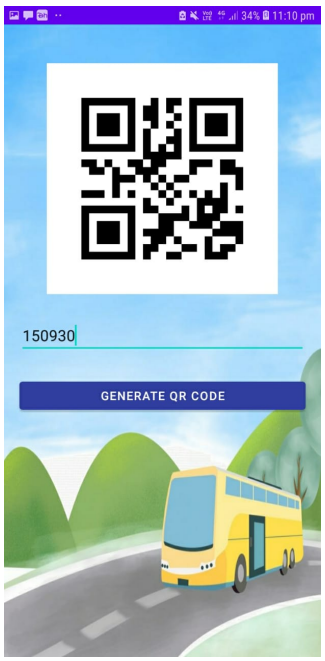

Figure 4.26: Generator

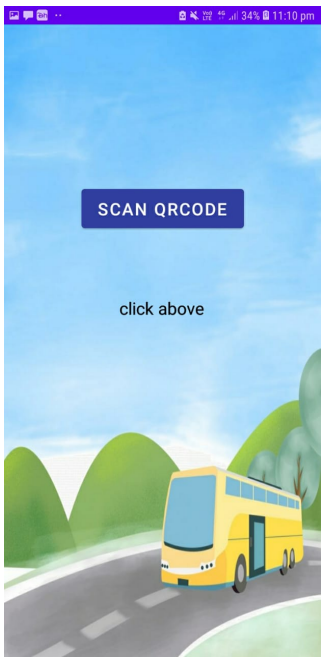

Figure 4.27: Scan

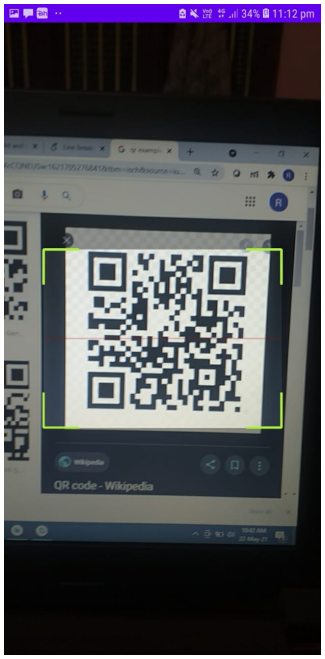

Figure 4.28: Scanning

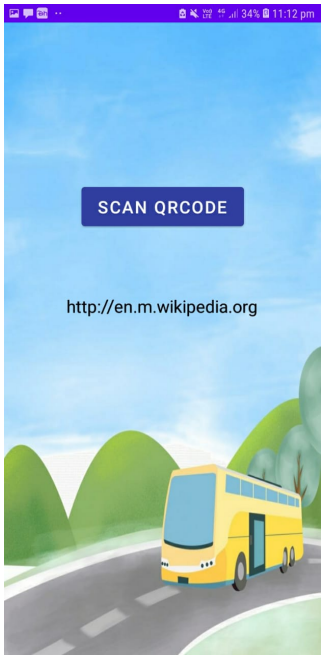

Figure 4.29: Result

CHAPTER 5

CONCLUSION

In this we have discussed how our mobile application is useful for the people travelling through bus everyday. Though there were many mobile applications for the bus ticket booking we have designed a mobile application in which we make the process easier for the people travelling by government buses. This is the user friendly application and as we already mentioned this new system will support the eco-green campaign that urged by the government with using less paper.

In our project we have created a mobile application in which a user books the ticket through his/her mobile phone and the user is acknowledged about his/her payment by the OTP sent to their mobile phone and user will be able to get his/her Qr code with the help of OTP.

Also, we have introduced crowd monitoring in our mobile application which helps the user to maintain social distancing inside the bus which is very important in this pandemic situation.

5.0.1 Future enhancement

We have designed our project with minimal number of bus routes. This can be further developed with even more number of bus routes, So that it would be even more better way of making the application useful to the people and can also be implemented in each districts.

APPENDIX A

TRAVEL EASY

A.1 Firebase

Firebase is a backend platform for building Web, Android and IOSapplications.

A.2 OTP

Firebase is a backend platform for building Web, Android and IOSapplications.

A.3 SHA

A family of popular cryptographic hash algorithms used to create digital signatures and hashes for blockchain transactions.

A.4 JSON

Json is an open standard file format and data interchange format that stores and transmits data objects consisting of attribute–value pairs and arrays using human-readable text.

A.5 UI

The user interface (UI) is the set of screens, pages, and visual elements—such as buttons and icons—that allow a person to engage with a product or service at its most basic level.

A.6 SDK

A software development kit is a bundle of software development tools that enable us to create apps for a certain software package or software framework.

REFERENCES

[1] QR GENERATOR. https://www.c-sharpcorner.com/article/how-to-generate-qr-code-in-android. Accessed: 20 Feb 2020.

[2] QR SCANNER. https://learntodroid.com/how-to-create-a-qr-code-scanner-app-in-android/. Accessed: 2021.

[3] OTP AUTHENTICATION. https://www.geeksforgeeks.org/firebase-authentication-with-phone-number-otp-in-android/. Accessed: 31 Dec 2020.

[4] FIREBASE. https://www.geeksforgeeks.org/firebase-authentication-with-phone-number-otp-in-android/. Accessed: 21 Jan 2020.

[5] FIREBASE. https://youtu.be/DD32pAVpKHA. Accessed: 17 August 2018.

[6] T. Winsolin. Bus ticket granting service in mtc buses using android application with the help of qr code. *International Journal Of Engineering And Computer Science*, 04 2017.

[7] M. Anuradha, M. V. D. Devi, K. Keerthana, and K. Dhanasree. Smart bus ticket system using qr code in android app. *International Journal Of Engineering And Computer Science*, 2018.