"QR CODE ENABLED TOLL COLLECTION WITH ENCRYPTION AND PAYEMENT GATEWAY INTEGRATION"

Submitted to the

SCHOOL OF COMPUTER SCIENCE

In partial fulfilment of the

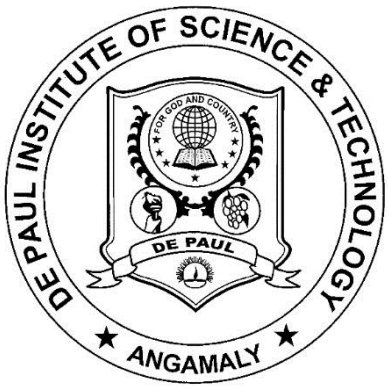
DUAL DEGREE MASTER OF COMPUTER APPLICATIONS

Under the guidance of

Asst. Prof DENNY P FRANCIS

Research Report Done by

JOYS JOHNEY
(Reg no: 150208)



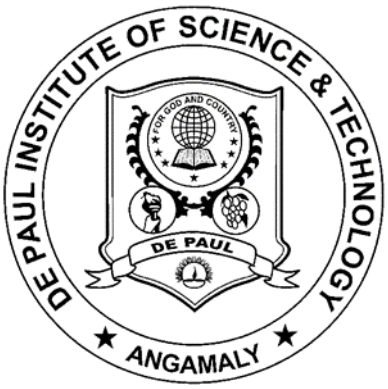
SCHOOL OF COMPUTER SCIENCE

DE PAUL INSTITUTE OF SCIENCE & TECHNOLOGY

ANGAMALY

MARCH – 2020

DE PAUL INSTITUTE OF SCIENCE & TECHNOLOGY
ANGAMALY



BONAFIDE CERTIFICATE

"QR CODE ENABLED TOLL COLLECTION WITH ENCRYPTION AND PAYEMENT GATEWAY INTEGRATION"

is a bonafide work done by

JOYS JOHNEY

In partial fulfilment of the requirement for the Award of

DUAL DEGREE MASTER OF COMPUTER APPLICATIONS
Degree From

Mahatma Gandhi University, Kottayam

(2015 - 2020)

Head of Department Research Guide

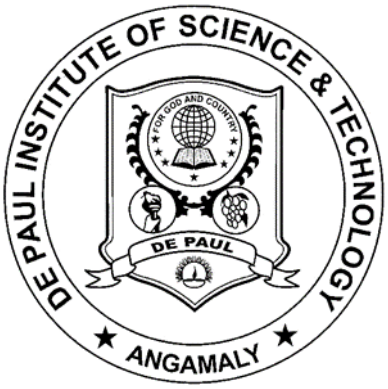
Submitted for the Viva-Voce Examination held on……...

External Examiner1
(Name & Signature)

ExternalExaminer2
(Name & Signature)

DE PAUL INSTITUTE OF SCIENCE & TECHNOLOGY

ANGAMALY



CERTIFICATE

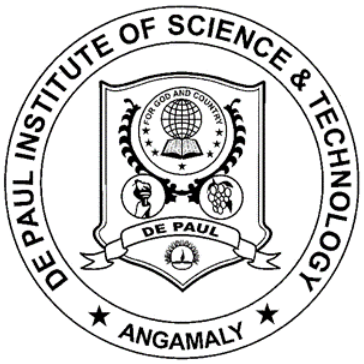
This is to certify that the project entitled **"QR CODE ENABLED TOLL COLLECTION WITH ENCRYPTION AND PAYEMENT GATEWAY INTEGRATION"** has been successfully carried out by **JOYS JOHNEY** (Reg no: 150208) in partial fulfilment of the Course **DUAL DEGREE MASTER OF COMPUTER APPLICATIONS**.

HEAD OF DEPARTMENT **RESEARCH GUIDE**

Date:

DE PAUL INSTITUTE OF SCIENCE & TECHNOLOGY

ANGAMALY



DECLARATION

I, **JOYS JOHNEY**, hereby declare that the project work entitled **"QR CODE ENABLED TOLL COLLECTION WITH ENCRYPTION AND PAYEMENT GATEWAY INTEGRATION"** is an authenticated work carried out by me under the guidance of **Asst. Prof DENNY P FRANCIS** for the partial fulfilment of the course **DUAL DEGREE MASTER OF COMPUTER APPLICATIONS.** This work has not been submitted for similar purpose anywhere else except to **DE PAUL INSTITUTE OF SCIENCE & TECHNOLOGY ANGAMALY**, affiliated to **M.G.UNIVERSITY, KOTTAYAM.**

I understand that detection of any such copying is liable to be punished in any way the school deems fit.

Date: **JOYS JOHNEY**

Place:

Signature

ABSTRACT

Today, all most all toll plazas are operated manually. We often encounter traffic jams at almost all toll plazas due to this toll collection and the in-efficient training provided to the toll plaza workers. The time we spend at toll traffic is a waste of time. To avoid such delays we are developing a system using the latest technology QR code. This system will reduce the burden of collecting tolls and also reduces the man power required. In this system, scanners are used to capture the QR code and tag each vehicle as it passes through the toll gate. The QR code will be decoded by the central database and if validates the toll amount for the particular vehicle is deducted from the registered account of the user. The QR code is encrypted using algorithm. In the present day RFID (Radio Frequency Identification) has been used for this purpose. In order to overcome the issues of RFID tags QR code is proposed.

Android based application is used for the usage of generating the unique QR code which is encrypted using the algorithm. The application allows for the user to register and login using personal information and generate QR code by providing the vehicle details hence a unique QR code is generated for that vehicle, any number of vehicle can be added which belongs to that person. The application *integrates a payment gateway* which is approved by the government hence enables an ease of transaction. We have included many additional features like *emergency SOS* call button, *SMS system* which regularly updates the customer On each toll passage along with the balance amount. *24/7 support and road side assistance* can be initiated within the application. With the payment gateway integration the user can recharge the wallet within the application hence enhancing the digital India scheme.

The RFID automatic toll gate system can automatically discover the vehicles of the identities, reading items in motion and tracing of the vehicles can be done by accurately by RFID. In this paper they have executed a framework which will punish for infringement of toll entryway and they believe it will prompt to a fastidious activity. The framework developed will help in reducing the number of mischance. The framework integrates the RFID, AVR microcontroller, the database creation and GUI outline.

QR code that was seen as new age technology and allows us to greatly reduce the manpower and reduces traffic congestion. QR code allows for the toll collection system to be much better than the RFID tags used in today's vehicle. QR code gives authorities to set variable pricing for toll services and allows for a fair tax collection.

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank ***Fr. George Pottayil****,****Director*** *of DiST and* ***Dr.Unni C J,Principal****,De Paul Institute of Science & Technology(DiST) giving me all necessary facilities in the college for successfully completion of this dissertation.*

I record my gratitude to ***Mr Joseph Paul, HOD, Dept. of MCA*** *for his support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although he had busy schedule.*

I owe my deep gratitude to my project guide ***Mr Denny P Francis****,* ***Assistant Professor****,* ***Guide of this project*** *who took keen interest on my project work and guided me all along, till the completion of my project work by providing all the necessary information for developing a good system.*

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all teaching staffs of Department of Computer Science which helped me in successfully completing my project work. Also, I would like to extend my sincere esteems to all staff in laboratory for their timely support.

NAME: **JOYS JOHNEY**

TABLE OF CONTENTS

CHAPTER I
INTRODUCTION

A toll road which fee is assessed for passages have been implemented to recoup the money spends for construction and maintenance of the road. Transportation has become a human life daily routine hence there is a probability that everyone would encounter a toll plaza at one point of their life. Manual toll collection system became out-dated due to the manpower required and number of drawbacks. Indian government has recently pushed for automatic toll collection using RFID technology this has several drawbacks. RFID uses radio waves to communicate with the receiver. An IR receiver is used to receive the pulse and a sends it to the controller, which transmits the vehicle through the RF transmitter located in the vehicle. Some of the existing system uses RFID tags while others use the GSM module and INFRARED technology. We have designed the system keeping in mind to reduce the cost and also to find a solution to the existing traffic congestions at the toll booths. We suggested the use of QR technology for the *payment of toll gate processing*. Recent trends in the market have shown an increase in the use of QR codes than can be scanned and read by camera in a smartphone. The QR code allows storing data, numbers and even using it for payment and transfer of money.

Android based application is used for the usage of generating the unique QR code which is encrypted using the algorithm. The application allows for the user to register and login using personal information and generate QR code by providing the vehicle details hence a unique QR code is generated for that vehicle, any number of vehicle can be added which belongs to that person. The application integrates a payment gateway which is approved by the government hence enables an ease of transaction. We have included many additional features like emergency SOS call button, SMS system which regularly updates the customer On each toll passage along with the balance amount. 24/7 support and road side assistance can be initiated within the application. With the payment gateway integration the user can recharge the wallet within the application hence enhancing the digital India scheme.

QR CODE

QR Code is a matrix barcode that can be used to store data. It consists of black modules arranged in a square grid which is then read by the camera. Data is extracted from patterns in both directions of the image. QR code offers numerous benefits such as [6]

- Cost effective
- Readable from distance
- Structured appending

On a grid of 6x6 mask pattern is defined which is necessary to cover the whole symbol. Patterns in data area such as blank areas or misleading features that look like locator marks can confuse scanner, hence masking is used. QR code has become a focus of advertising strategy, since it provides a way to access a brand's website more quickly than by manually entering a URL.

ANDRIOD

QR Code based toll collection uses different system from the hardware components to the user component in this paper we assume that all the pre-defined hardware for the real world systems are implemented as we already have existing toll plaza only minor upgrades have to be done in order to fit the scanners. This paper talks about the android application. Android is a mobile operating system by Google. It is based on the Linux kernel and other open source software. Android is basically for the touch and type interface that can be
Implemented in a smartphone and other devices. Google has further developed android TV, Wear OS, and Android Auto that is used in cars.

Android is designed to keep the processes consume the battery charge at a minimum rate. Android has enabled the market to be flooded with smartphones and hence opened up a whole new digital world to the different sections of the society. This revolution in smartphones has enabled more people to go online and use different technologies that are available to them hence it important to keep updating the technologies that can have a greater importance in people's lives.

PAYMENT GATEWAY

Payment gateway is a merchant service provided by an e-commerce application service provider that authorizes credit card or direct payment processing gateway may be provided by a bank to its customers, but can be provided by a specialised financial service provider as a separate service, such as a payment service provider. A payment gateway like Razor pay that we are integrating with the application allows the user to securely transact their payment an HTTPS protocol based transaction take place. Virtual payer authentication is something that the acquirers, issuers and the payment gateways are backing to secure the process even

more. PCI-DSS makes it secure enough to allow the user to store their personal data in the portal or gateway for recurring payments. The most significant advantage of a payment gateway is the fact that it allows millions of users to use it at the same time, making it possible for you to purchase or sell goods and
Services whenever you want.

SMS

SMS (Short Message Service) is a text messaging service component of most telephone, internet and mobile device systems. The protocols used in this service allow users to send and receive messages to and from GSM mobiles. SMS although commonly seen in the mobile to mobile services it can be expanded into technologies that offer services and support. SMS is a stateless communication protocol in which every SMS message is considered entirely independent of other messages. In this system SMS is used to send alert, balance and other toll based reports directly to the user mobile. This enables the user to have a regular update on the main account balance of the wallet.

CHAPTER Ⅰ Ⅰ
LITERATURE REVIEW

Automated toll collection and check post system using Radio Frequency Identification (RFID) and Global System for Mobile Communication (GSM) module. The recognition is succeeded with the guidance of passive radio frequency. In this paper [1], vehicle particulars like unique ID is saved in RFID tag which is attached in the vehicle. Image process and GPS is combined with RFID and GSM module to create the system a lot reliable and secure. The hardware design of the paper consists of transmitter and receiver module. Transmitter module is fixed in the vehicle as an active tag. Receiver is the automated check post and e-toll control. ATmega328 Arduino controller is a 28 pin Arduino controller which has 32 bit natural working registers. The servo Motor SG90 type is used in this setup. The servo Motor is used for automatic gate operation whenever the motor receives the signal from the controller. SIM800 GSM is a communication device designed for global market. RFID uses radio wave to process the information from the devices. The bi-directional connection network with endpoints has been designed to use RFID to produce an electronic product code. An experimental study result has also been given at the end of the paper.

In paper [2], the toll collection system is designed primary for the use by GSM and GPS systems. The GPS is used to find the position of the vehicle. The GPRS kit should be installed in the vehicle in track the vehicle. Each GPRS system has a subscriber identity module. This system also incorporates geo fences of the toll plaza to get the information regarding the location of the toll plaza. The GPS and GPRS are integrated into an ARM microcontroller. In order for the GPRS system to work the system requires 1 GB of data and a stable network connection which should be switched on all the time. The system also requires a mobile tower to be located at the toll plaza to process the information of the vehicle in the rage of the tower. The system compares the position of the vehicle and when the vehicle is within the 5 meter radius of the toll plaza the database is updated and the amount is debited from the user's account which will be immediately followed by an SMS to the customer. The position of the vehicle or the GPRS SIM module is compared using the haversine formula, and if the vehicle is within the range of toll plaza, amount will be taken from the account.

3D environment modelling for the toll collection which directly affects the decision making part, in paper [3] the new toll gate approach is divided into perception, decision making and motion control. A deep natural network is used in Perception. Virtual lanes are generated from the 3D environment result and an optional is selected. In motion control a collision free path is planned and transmitted to maneuver the

vehicle. The automation driving vehicle receives the localization and mapping (SLAM) algorithm estimates the motion state of the vehicle. The traffic patterns are further understood and scanned to detect passable electronic toll collection gate. These virtual lines are generated only for the types of vehicle were this system can be implemented.

Convolutional neural network (CNN) based algorithm is used for object detection to sense other vehicle, and ETC gates. YOLO algorithm a type of CNN with structure of Google Net, is divided into 24 convolutional layers with connected layers of 2. The determination of total queue length is calculated by first detect the neighbouring vehicle and correlate the distance of vehicle to the gate. The SLAM used to update or construct maps on an unknown environment. LSD-SLAM (Large Scale Direct Monocular) is applied to track and map by image intensities divided into three steps: tracking, depth map estimation and map optimization. The system takes a while to scan all the gates as the gates could be occluded by trucks or buses. The coordinates are interpreted and the best cubic coordinate origin and the ETC gate are selected. The motion control adopts an adaptive method. It receives a collision free trajectory data from path planning using curvature, yaw rate and velocity. In this approach LQR-PID algorithm is used. LQR can exactly pave a way to the optimum pole. LQR predicts an expectation as inputs to PID controller. PID is a classic control with strong adaptation and robustness. YOLO algorithm can detect most of the vehicles. ETC signs can also be detected using this algorithm. The proposed system is designed to be more universally usable, even without HD map and V2X. The V2X will make visual classification of ETC gate obsolete.

In paper [4], a proposed system is a web application and android application. The system is designed primarily for devices like smartphones, personal computers and all other devices which support web services. The main objective behind this paper is to design the application which provides an effective and easier way to payment of road toll. Keeping in mind the Indian condition the application contains QR code for recognition with centralized availability of data. Throughout the system it enables user to pay from the account created after reaching the toll booth. The system gives many advantages and assures and accurate collection of toll amount. This paper uses GUI for collection of toll, the real time monitoring and management is done. The architecture of the system uses user application to generate QR code which can be connected by GPS for the connection of toll plaza receiver. The data base of the system saves the details of the vehicle connected and other different toll collection. The server coordinates all the different activates of the application.

RFID is a dependable technology in paper [5], the RFID automatic toll gate system can automatically discover the vehicles of the identities, reading items in motion and tracing of the vehicles can be done by accurately by RFID. In this paper they have executed a framework which will punish for infringement of toll entryway and they believe it will prompt to a fastidious activity. The framework developed will help in reducing the number of mischance. The framework integrates the RFID, AVR microcontroller, the database creation and GUI outline. RFID tags are fixed and attached and through this the reader reads the data. The main aim of this paper is addressing the prevention of motorists and toll authorities manually perform ticket payments and also check driving without documents. This system proposes to identify theft vehicles. When the gate is automated it requires minimum human intervention hence efficiency can be improved. The framework expands wellbeing. The paper also looks at the adequacy of toll stations and the road developments that limited to the toll road. RFID technologies implemented in the system adopt a kind of frequency chips which authenticates and authorizes protocol model used to guarantee system security. Accordingly, electronic toll collection system deserves deeply research whether from technology, economy or environment protection.

CHAPTER Ⅲ

THEORETICAL BACKGROUND / CONCEPTS / METHODOLOGY / DESIGN / MODELING

INTRODUCTION

My FASTTag is an application created by the government of India for the recharge of the fast tag cards issued to the user. The FASTTAG system widely uses the RFID (radio frequency Identification). FASTTag is affixed to the windscreen of the vehicle and can be driven through the toll plazas without stopping for a payment. The amount for that particular drive will be deducted from the user bank account issued through the participating banks which has issued the FASTTag.

The RFID uses electromagnetic fields to automatically identify and tracks tags attached to the object. An RFID tag consists of a tiny radio transponder; a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader.

The application My FASTTag allows the fasttag users to recharge their device by entering information. The application is only being used for recharge and not for any other purpose.

My FASTTag FEATURES:

Instant FASTag recharge through Unified Payments Interface (UPI)

Any issuer bank's FASTag can be now recharged through My FASTag App using UPI payment. You can create a New UPI ID instantly and pay/recharge through any of your bank accounts, or pay using any BHIM UPI App active on your mobile. Money gets transferred directly from your UPI linked bank account to your FASTag Wallet/Account. Only the FASTag issuer banks which are live on UPI platform to accept FASTag reloads will appear in the list of banks in the App.

• Customers who have purchased an IHMCL FASTag can download this myFastag App and 'Link IHMCL FASTag' with their bank account.

• When user clicks on 'Link IHMCL FASTag,' system will ask for a reference number received on Email/SMS at the time of buying the tag from IHMCL FASTag vendor.

• The mobile number & email ID provided by the customer at the time of IHMCL FASTag purchase should be active and available to complete the tag linking.

• Customer's bank must be live on NETC platform for linking with IHMCL FASTag

CHAPTER Ⅳ
PROJECT IMPLEMENTATION

INTRODUCTION

In this study, we focussed on collecting toll according to a vehicle and provide uniform toll collection system. The approach of automatically toll collection helps to avoid unnecessary delay in collection of toll and provide safe, secure, effective strong system in real world transport system. For an effective and fast collection of toll on toll plaza, we developed QR code based toll collection system. QR code mounted on the vehicle used to read vehicle with the help of QR code reader. In this system when the vehicle reaches near the toll plaza, the camera captures the image; send it to QR code reader for the decoding process. It will retrieve the vehicle database. Depending on this information appropriate toll tax will be deducted from the owner's prepaid account and boom will open automatically and the vehicle can pass without any delay. If the balance in the owner's prepaid account is low or if the vehicle is not equipped with a tag, then the owner has to pay toll manually.

ARCHITECTURE

First vehicle owner needs to download the application and has to register by giving basic details such as username, vehicle details and then registering an account pertaining to the owner either prepaid. During the registration, the owner needs to deposit ascertain amount in his account. The registration provides QR code to the vehicle. The second module is toll management.

The gate needs to register for the various toll amount levied on different vehicle types that approach a toll gate. As the vehicle approaches the gate, QR code reader captures the QR code and decodes, authenticates and the respective amount will be deducted from owner account.

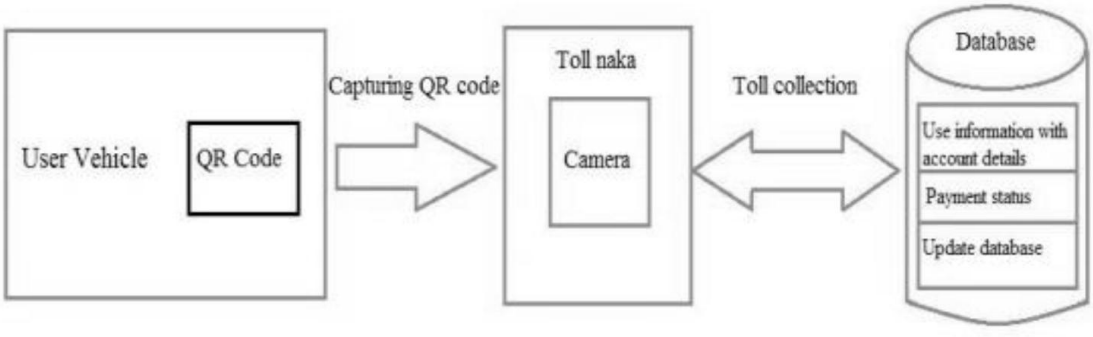


Figure 4.1

FLOWCHART:

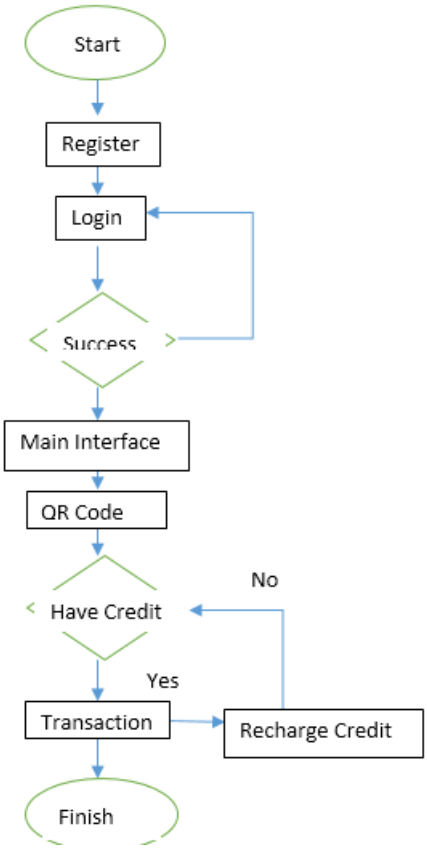


Figure 4.2

Generally the selected operating system is android to over all the features. SMS is used for communicating with the user whenever the user uses one of the application features. It will also allow the user to stay updated with the balance and QR code tag. The application developed to make a user-friendly approach which makes it easier and simpler to use.

The road user has the application where the user can register to login. The application allows the user to add and view vehicles and thereby generate QR codes for each vehicle. The QR code generated is unique for each vehicle added by the user. Hence, a clear and transparent system is being created. The system is developed where the road users has the complete knowledge about the toll plazas and their rules. The mobile application allows the user to add vehicle details using the RC details of the vehicle and the camera modules is activated to upload photos. User can view different toll plazas in between his/her routes and find the distance and what kind of toll plaza it is. Today in the modern world emergencies can occur at any time of the journey. We have included a special emergency tab where the user can select different category of emergency from the application tab and a call module will be invoked to the dial.

Travel during toll routes can be daunting task when the road we travel on is built in a poor quality and also the toll plaza management can sometimes show bad behaviour. This has been looked into and we have added a feature to report issues with the highway or toll. The report and issue will have the consolidated features which are divided into four and these will have camera module and the upload module where the user can upload the photos of issues from the camera.

Feedback is important to update and also to improve for the developers to have the road user's life more enjoyable. The feedback will have the user to enter a comment into the section and also rate the application with a five star. The application also has a profile tab where the user details can be viewed and edited. These have been included after the research and analysis of the existing project. The very basic of the project is to have a transparent view from the existing system and also a great implementation of the proposed system. The system will be a top notch system when compared to other government applications. The system also removes the need for any third party in the system.

The SMS facility is included where the system will notify the user whenever a change in the attribute is being made. The history for the user is important as to know where and amount that has been deducted from. The date attribute will allow the user to know when the user account has been debited. For better result a good mobile connectivity, broadband connectivity and location service should be enabled.

We have divided our project into different modules and they are:

- Vehicle
- Nearby Toll
- Emergency
- Report an Issue
- History
- Camera
- SMS
- Feedback
- Recharge

The main module of this project is the vehicle module where the QR code can be established and the vehicles can be added or deleted. The generated QR code can be downloaded from two options which are being placed for the user convenience.

AREA OF WORKING:

In this project my **area of working** involves Recharge and main balance, Emergency, Feedback and Scanner app.

One of the main area of the application is the recharge option where the user would be able to recharge an amount in to the main balance of the application. The recharge is done with the help of the payment gateway integration. The payment gateway helps us with the authentication and the user experience with the banking system.

The emergency module in the application is where the user can access the emergency service dials from the same page. A different emergency services had been brought together under one page so the user can dial different services according to the need.

The feedback page allows for the submission of feedback as the name suggests to the controlling authority of the application.

SCANNER

Scanner application created is a mock up model of the real life scanner. The substitute of the application will be able to scan and decrypt the QR code which show cases what the actual system might be able to do. The scanner even though it's an application created in android. The system actually represent the whole of toll plaza.

The toll plaza cannot be made as it require a lot of hardware and software complexities to be taken into. Our area of research included the automatic toll collection and we have focused mainly on the user application rather than the toll plaza side. When we talk about introducing a new technology into the already existing technology there we will be required to take into account the hardware requirements of the system.

The scanner being a mockup will allow the user to establish the connection of how the QR code gets scanned. The scanner is being used to decrypt the encrypted QR code so that the security of the QR code and user data is protected. The scanner system is simple at its best, when the system if one day is implemented into the real life then the application of scanner can be used to scan the QR code as a backup if the scanners placed in the toll systems doesn't work. The application is a standalone attribute of the main application.

RECHARGE

A main account is being added into the application so that when a QR code is scanned for the system then the amount will be deducted from this account. A centralized balance account is designed so that there is no need to create a balance account for each and every vehicle that has been added to the account.

The recharge is done when the user clicks on the add fund button from the home screen. The page will be directed to the system where the user can enter the amount she/he wishes to recharge for. Once an ok is clicked the integrated payment gateway system is takes over and the user is directed to the Razor Pay pay page where the user can select different options ranging from Credit cards to UPI payments.

Once the user selects the payment option the authentication of the payment is done by the interface and the user will be redirected to the home page. The main balance tab will updated if the payment was successful and a message regarding the same is also initiated to be send to the user phone number which the user used to register to the application. The payment if failed then there won't be a message rather there will be a toast displayed.

Payment Gateway Integration.

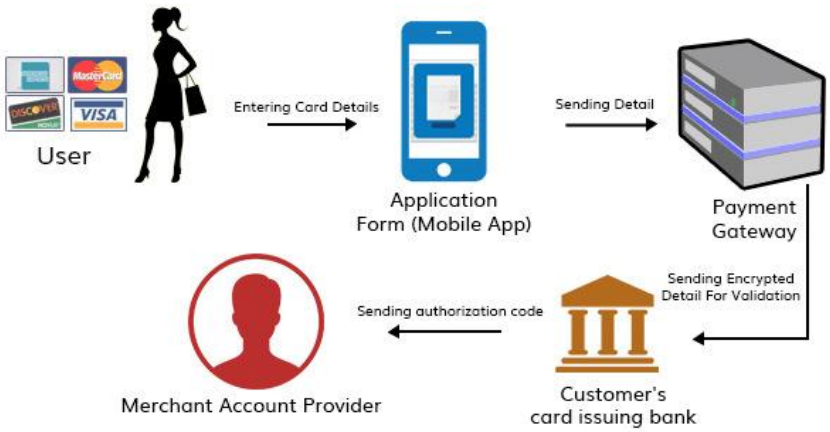


Figure 4.3

A payment gateway is an ecommerce service that processes online payments for online as well as offline businesses. Payment gateways help accept payments by transferring key information from their merchant websites to issuing banks, card associations and online wallet players.

Payment gateways play a vital role in the online transaction process, which is the realisation of value, and hence are seen as an important pillar of ecommerce. Hence integrating payment gateway into the system is an unavoidable part of the development of the application. This gateway helps us with the authorisation, capture, void, refund and in sale related graphics of the complex banking system.

Emergency is a page where the user can access the emergency service dial systems. The user will be able to call the services from the different options available. The system has options such as Police, Fire, Ambulance, ChildLine and Highway emergency number.

The number will be routed to the service dials and the service dispatcher will be able to pick up and register the complaint and do appropriate actions. This page is just a mere middle party where the user can use the page to access the services.

Feedback is just a page where the user can submit the feedback. The Name, Phone number, comment and a star rating can also be given. The comment section is where the user can give an elaborate reason why she/he love or hate the application.

AES DECRYPTION

Cryptography enables the confidentiality of communication through an insecure channel and hence protects against unauthorized parties by preventing unauthorized alteration of use. The Advanced Encryption Standard, AES (Rijndael) algorithm is implemented as described in the NIST (National Institute of Standards and Technology) Federal Information Processing Standard.

The AES Decryption process works as follows:

[1]. Processor writes the key data to AES Key register. The key setup operation is requested by writing the KEY_SETUP command to the AES Control register. The completion of the key setup operation is flagged by the interrupt signal and the busy bit in the AES Status register.

[2]. Processor writes encrypted data to AES Data register. The 128 bit input data block is delivered by four subsequent write commands to the AES_DATA register. The decoding operation is started by writing the AES_DECRYPT command to the AES Control register.

The completion of the decoding operation is flagged by the interrupt signal and the busy bit in the AES Status register. The key remains in the module until a new key is written or a reset command is performed.

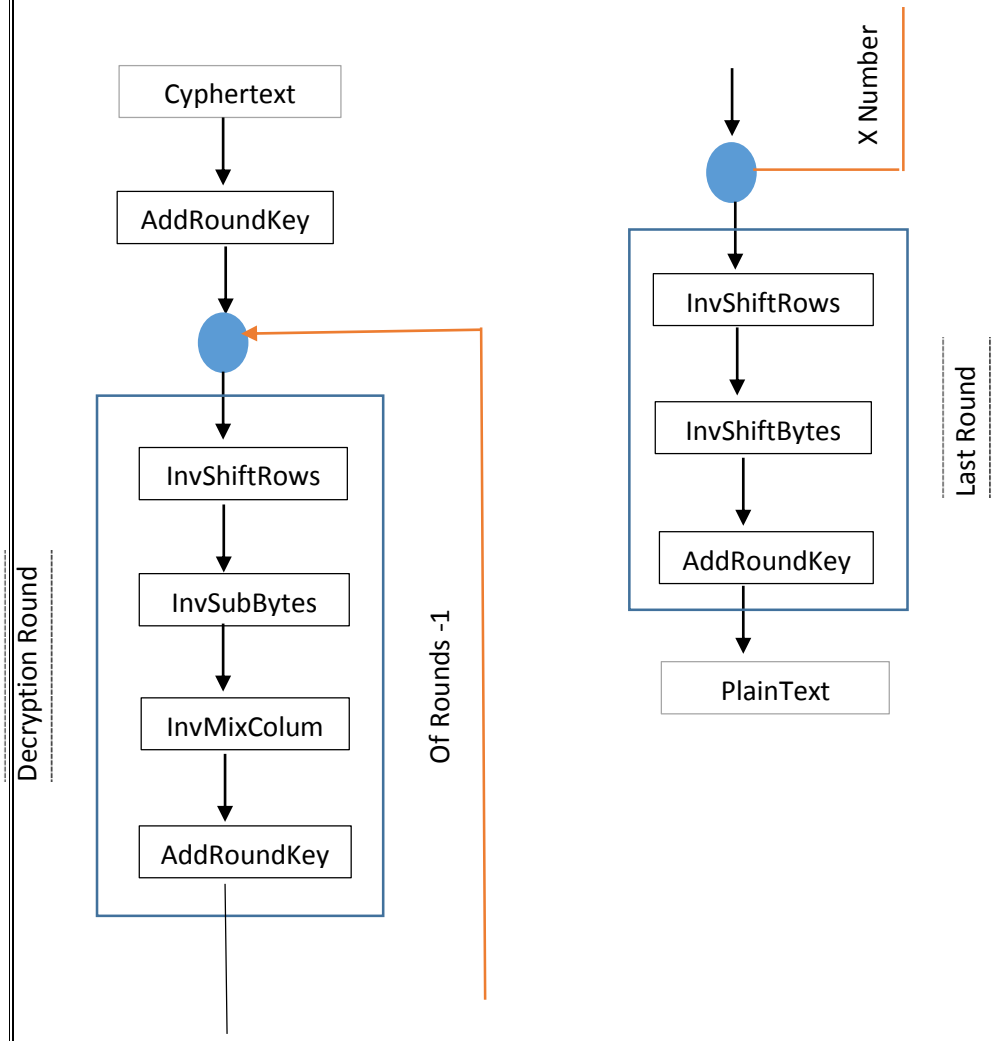


Figure 4.4

CHAPTER V

RESULT OF ANALYTICAL DESIGN

In this we evaluate the efficiency of the application in terms of time of execution, vehicle scanned and tagged and the through put of the system.

In this section, we look into the already existing system the fast tag model and the proposed system the QR Code tag. Although similar in many ways there is huge difference in how these two systems act and work. Their differences can be measured through different system dynamics or attributes.

Time of execution:

The time of execution by the two systems are being projected here during two separate times. One during Normal hours and other by peak Traffic hours. The systems here are the fast tag and the QR code tag. The execution is clearly mentioned allows for a comprehensive understanding of how QR code tag works during heavy traffic flow. The execution means the transfer of tagging to amount time. This is much larger for QR code tag when compared.

Figure 5.1

Vehicle Scanner Range:

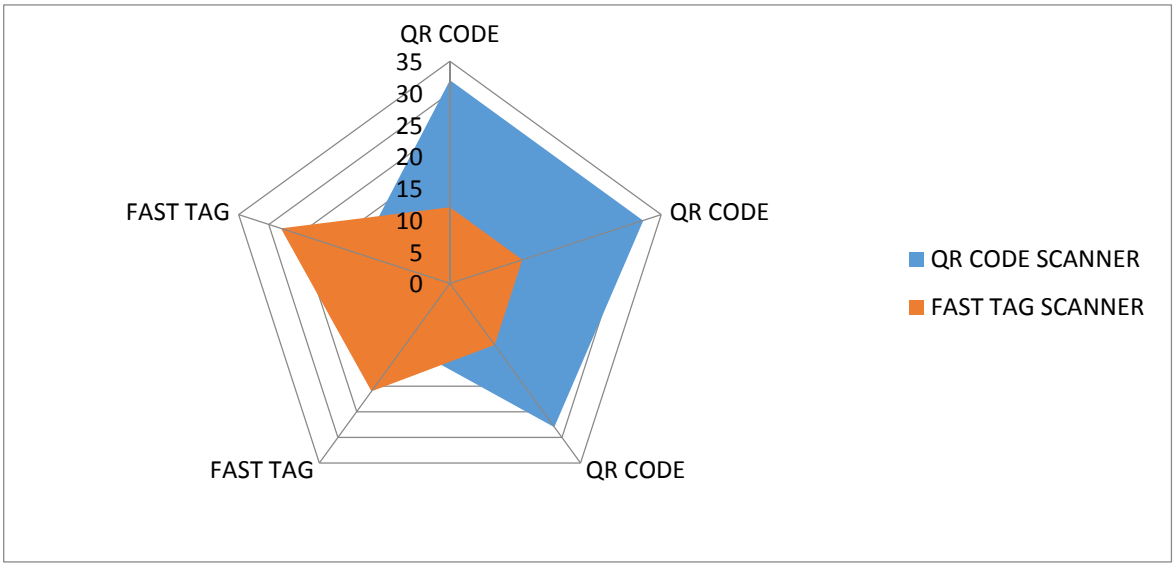


Figure 5.2

Each system has their own specific scanners in each toll plazas for their purpose. Each scanner has their own limitations and range. When the scanner for QR code tag is compared with the fast tag scanner. The fast tag had limited range and also the scanner used to multi-tag that same fast tag vehicle. The QR code tag scanner although supported by additional tag systems have much greater range and have clear tagging system in place.

Choice of people for toll payment:

The commuters when given a choice to choose between what is a great option in accordance with the government law or as an alternative to the fast tag system. The people choose QR code system over the fast tag upon knowing its vast implications and familiarity with the concept through QR codes used for the payments. The QR code system also has a user friendly application which let users directly control their tags rather than a bank or third person in between.

Cash and no toll payment was also popular choice among the users as they wanted a smoother ride through the roads.

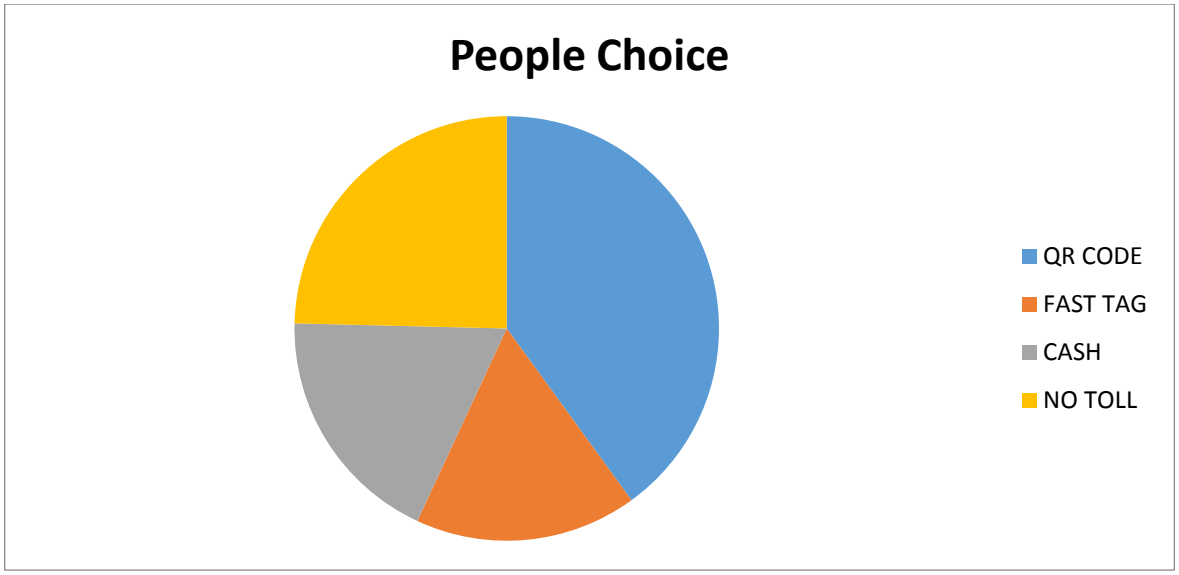


Figure 5.3

Throughput of System:

The vehicles passing through the tag lane is calculated here. From the approximate data collected through various sources the tag lane has an initial tag system and the final through the gate time. The vehicle when it approaches a fast tag lane is mostly held up by up to 2 vehicles due to the inefficiency in scanning the tags fast. While in QR code it is projected that the system will have a greater throughput through the gates due to the combined effects of the scanner and other tagging techniques.

Only delay that can be caused is when the user has forgotten to recharge the amount but it's made easier to recharge through the user friendly application available.

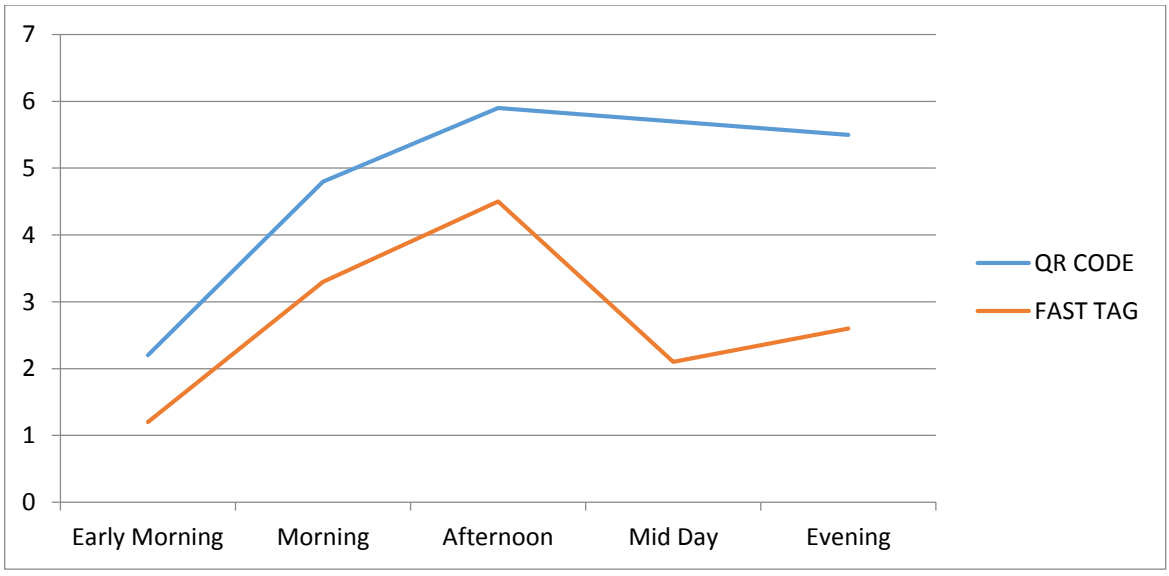


Figure 5.4

RESULT

India is a rising automotive market in the world and the amount of vehicle is said to increase and overtake China by the next two years. Due to these increase and the new government schemes which normalizes the use of fast tags and the fast lanes on toll plazas, the traffic is going to increase and a much efficient system which cater to both users is needed. The toll collection should be automated and a system which is cheap and requires very little money to maintain should be implemented in the country.

The roads build by the private contractors are being levied for toll collection and the road users should be able to pay for these without waiting in lane or traffic while at the same time these toll plazas should also make money.

QR code enabled toll collection makes it simple for the road user to charge up their balance and also to issue a QR code tag which is government approved and can only be collected through the RTO offices. The payment through QR code tag is transparent as user always has a report of it.

The various features that made QR code an instant success with the common people for digital transfer can also be found in this scheme too. Due to the preexisting knowledge about the QR code technology it’s easier and more trustable for the road users to use our application.

CHAPTER VI

CONCLUSION & FUTURE SCOPE

The approach of automatic toll collection using QR code that was seen as new age technology and allows us to greatly reduce the manpower and reduces traffic congestion. QR code allows for the toll collection system to be much better than the RFID tags used in today's vehicle. QR code gives authorities to set variable pricing for toll services and allows for a fair tax collection. Due to the technology, the QR code is generated after recharge and allows to get the QR code can be shown if the scanner is unable to scan the code. Due to the encryption algorithm used we try to integrate a payment gateway into the android application. This technology has high securable and low amount of speed cost. The messaging system build into the application allows for a regular update within the usage of toll. The implementation of this technology takes very little money and allows for updated driving dynamics as different toll gates will have different scanners that will tag each and every vehicle which very efficient as there are multiple scanners placed at regular points that tags the vehicle hence the scanning can never be avoided. The android application supports the QR code in building a strong network of commuters that help to keep the Indian roads less congested by the traffic. The android application is very user-friendly and anyone can easily use the application without the need of any help. The android application removes the need for a third party in order to proceed with the toll tag system that is been currently used today. The QR code tag is generated by the user for his/her vehicle hence the need of a bank account as seen today is eliminated. The android application allows different emergency and other conveniently added features to improve the road users experience and allows them to have a comfortable journey through out there travel on the highways knowing that they can always report any issue regarding anything on the highway to the government directly. This system is far more superior to the existing system in terms of usage and other activities.

FUTURE SCOPE

Every research application has its own merits and demerits .This research work has exposed almost all the necessities. Further improvements can easily be done since the coding is mainly structured or modular in nature .Changing the existing modules along with new technique or adding new research modules can append improvements.

The system itself being on the very basic idea of the QR code technology we can integrate a lot of features into the application as future scope. The QR code technology is being used in every digital payment gateway system and by integrating our QR code present in the vehicle with the Central governments BHIM application we can

improve the users experience and become a one stop QR code. The addition of QR code can be exploited by allowing this code tag to be used for fuel recharges as fuel stations can be found along the highways. This system allows for the road user to only recharge the main account balance in the application and not to worry about the payment of the fuel as they can easily scan the tag after fuel recharge.

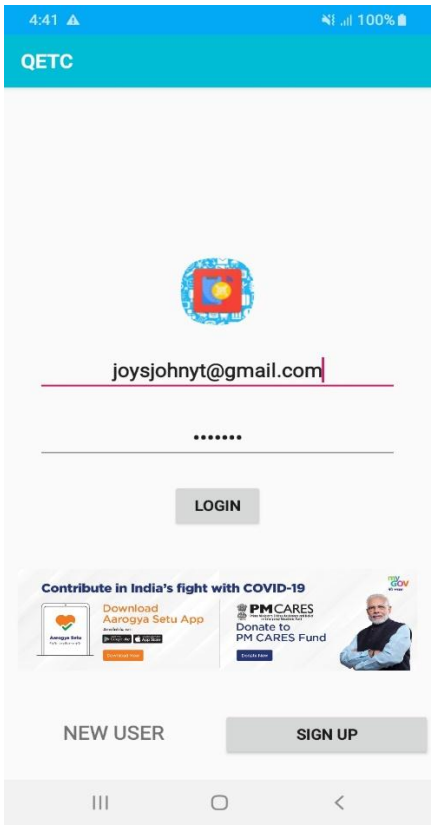
The system is easy to work with and allows the user to have added features such as tab where the nearby repair workshops along with tyre shops can be seen which are rated by the government and open twenty four hours.

This application has a future if implemented correctly. Then this could become the global standard in the toll collection which is automated.

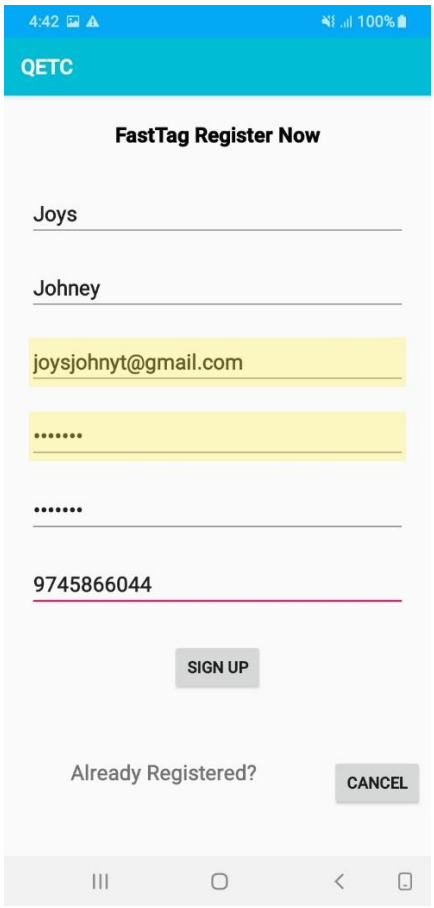
CHAPTER Ⅶ
APPENDICES

I. SCREENSHOTS

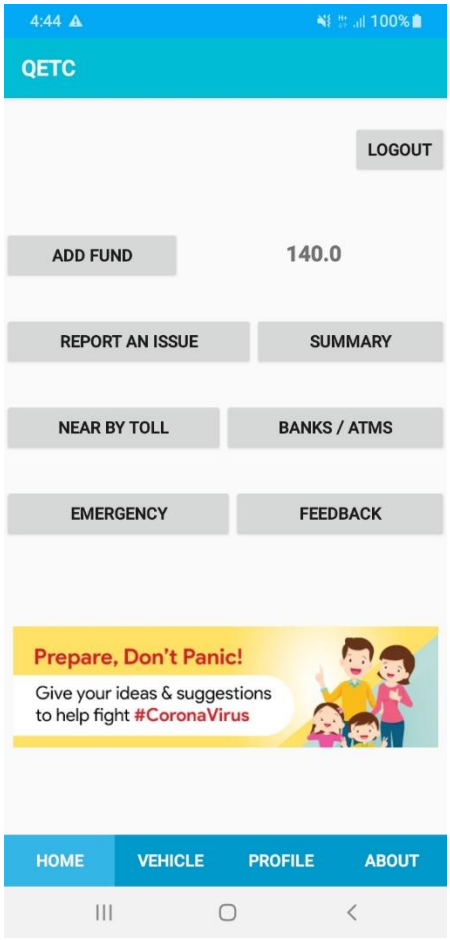
Login



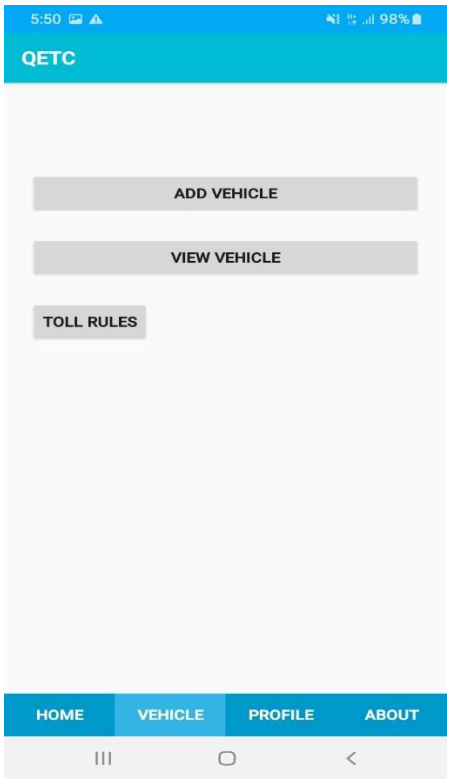
Register

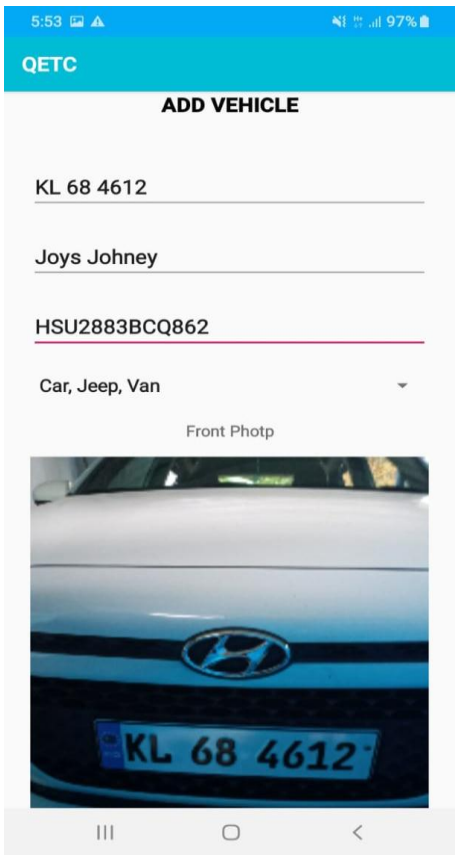


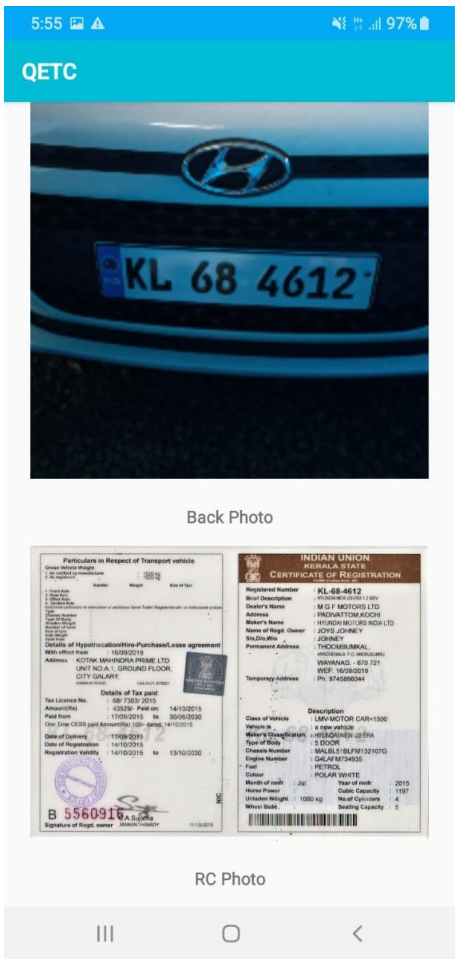
Home Page

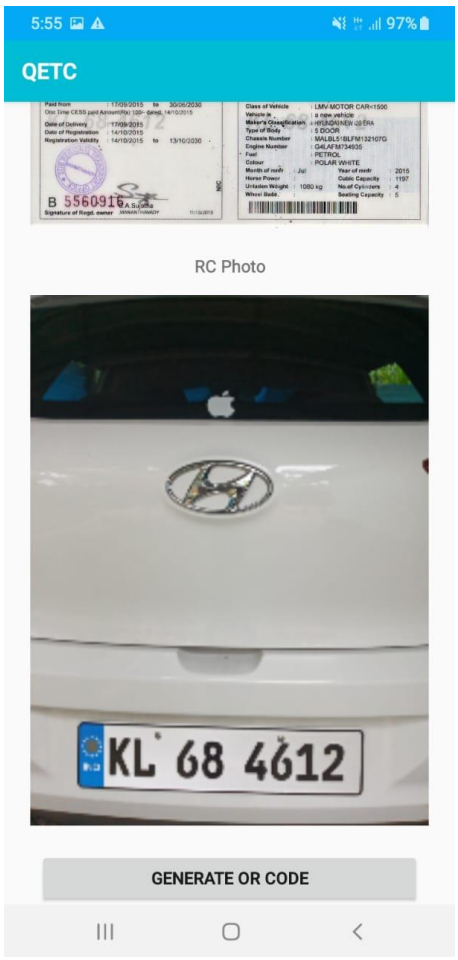


Vehicle

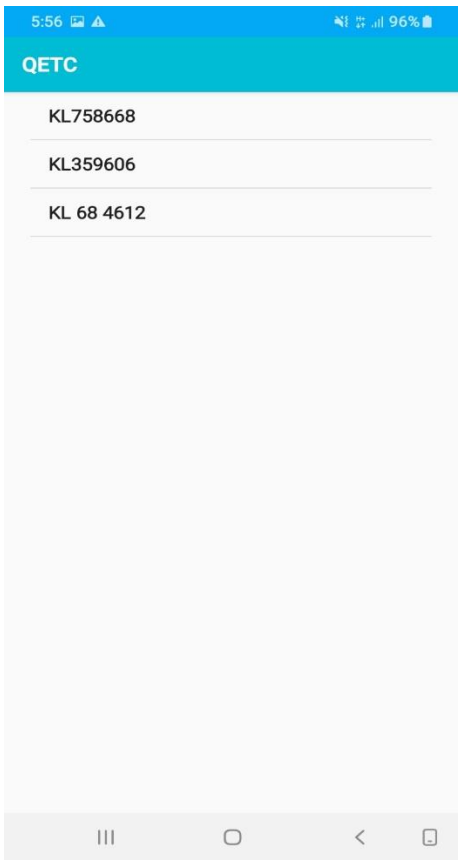


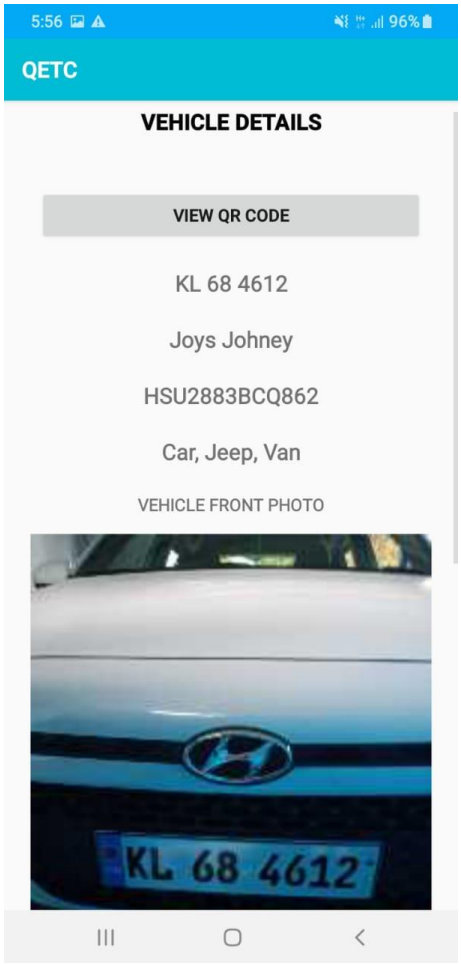

5:53
97%
QETC
ADD VEHICLE
KL 68 4612
Joys Johney
HSU2883BCQ862
Car, Jeep, Van
Front Photp
KL 68 4612


5:55
97%
QETC
KL 68 4612
Back Photo
INDIAN UNION
KERALA STATE
CERTIFICATE OF REGISTRATION
RC Photo


5:55
97%
QETC
RC Photo
KL 68 4612
GENERATE OR CODE

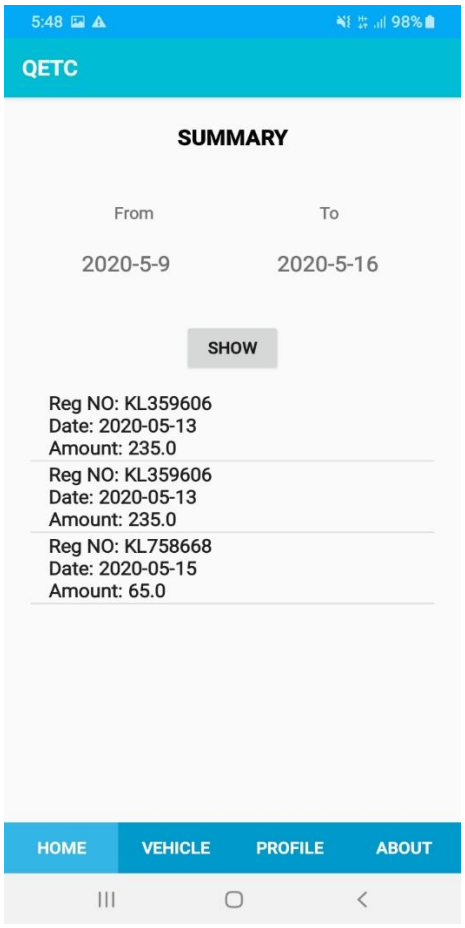

5:55
97%
QETC
DOWNLOAD QR


5:56
96%
QETC
KL758668
KL359606
KL 68 4612

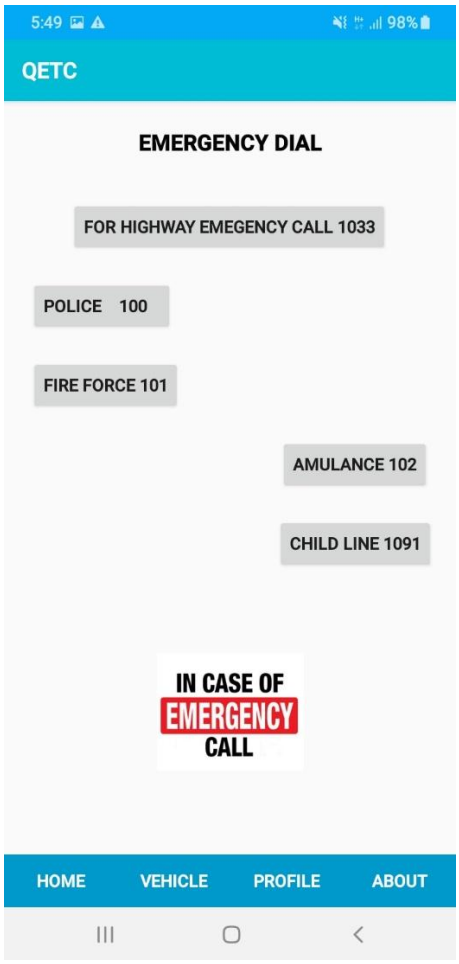

5:56
96%
QETC
VEHICLE DETAILS
VIEW QR CODE
KL 68 4612
Joys Johney
HSU2883BCQ862
Car, Jeep, Van
VEHICLE FRONT PHOTO
KL 68 4612


5:55
97%
QETC
DOWNLOAD QR

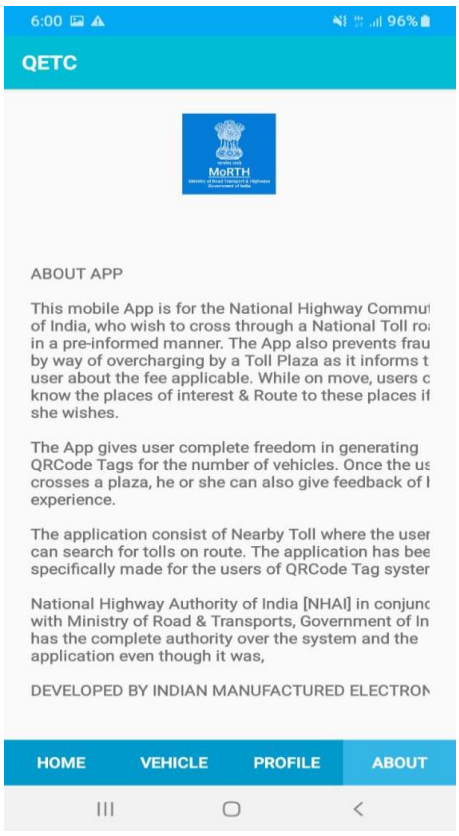
Summary


5:48
98%
QETC
SUMMARY
From
To
2020-5-9
2020-5-16
SHOW
Reg NO: KL359606
Date: 2020-05-13
Amount: 235.0
Reg NO: KL359606
Date: 2020-05-13
Amount: 235.0
Reg NO: KL758668
Date: 2020-05-15
Amount: 65.0
HOME
VEHICLE
PROFILE
ABOUT

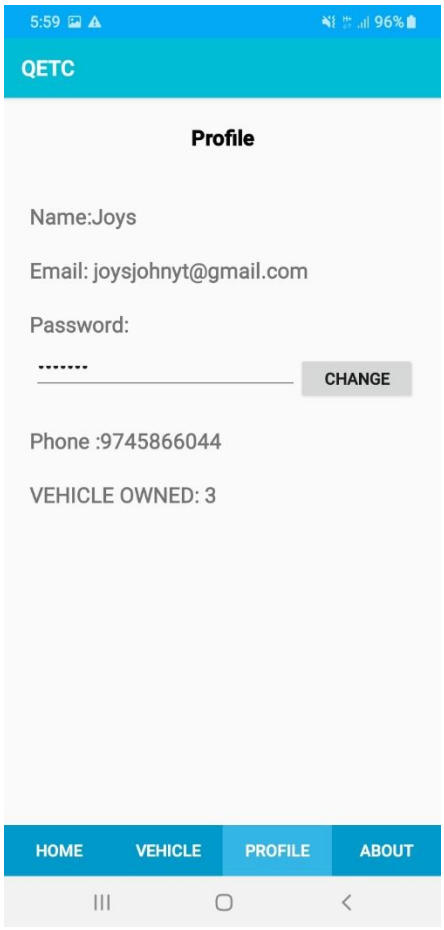
Emergency Dial



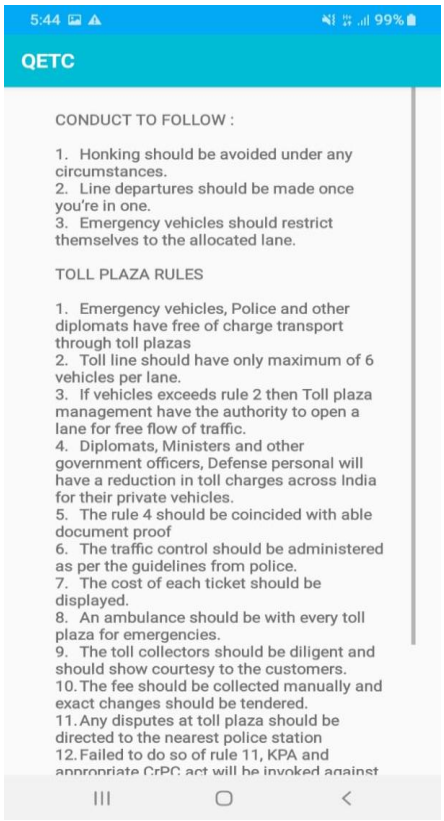
About



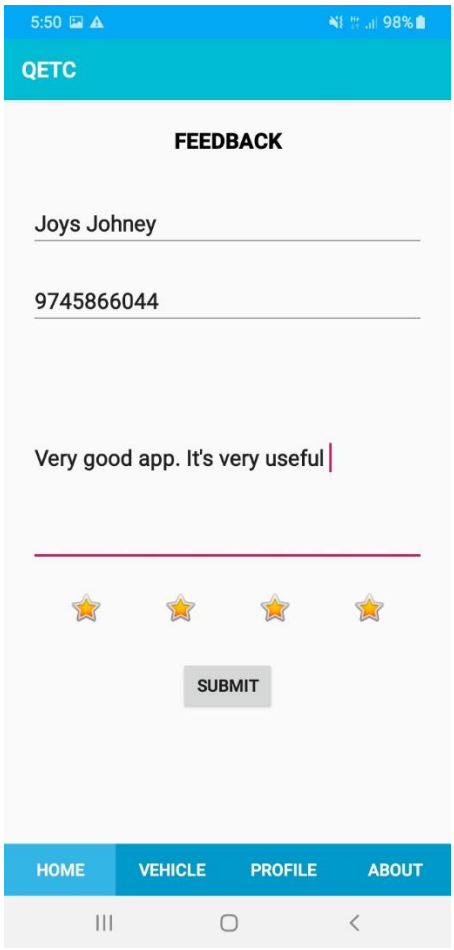
Profile



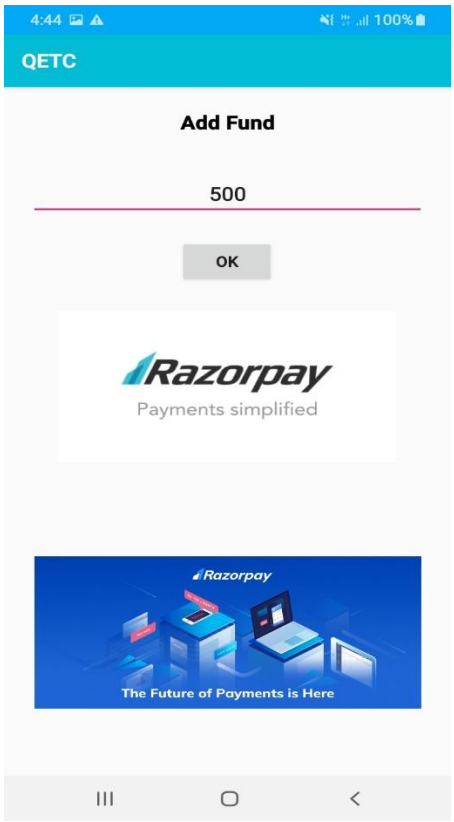
Toll Rules

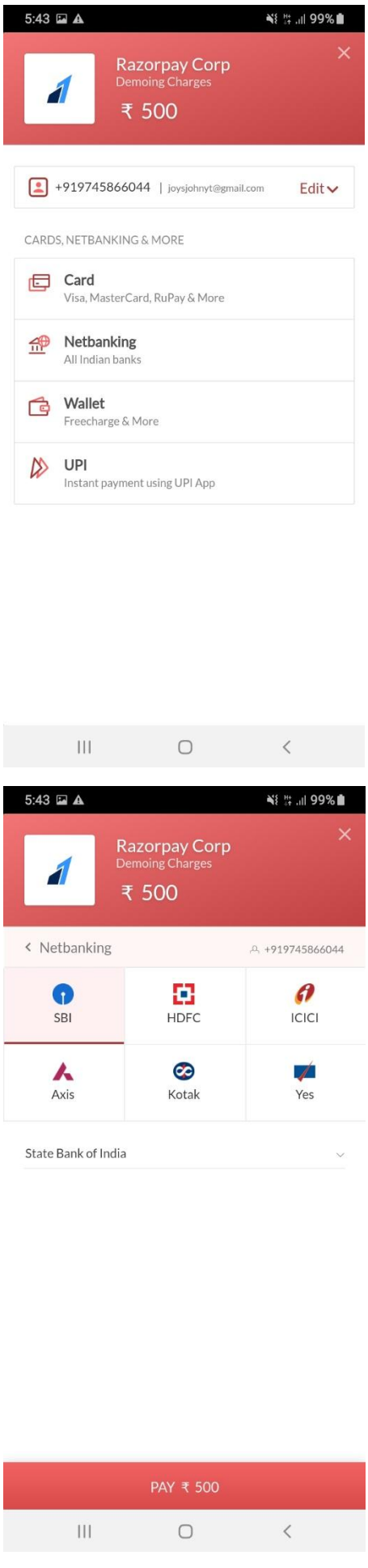


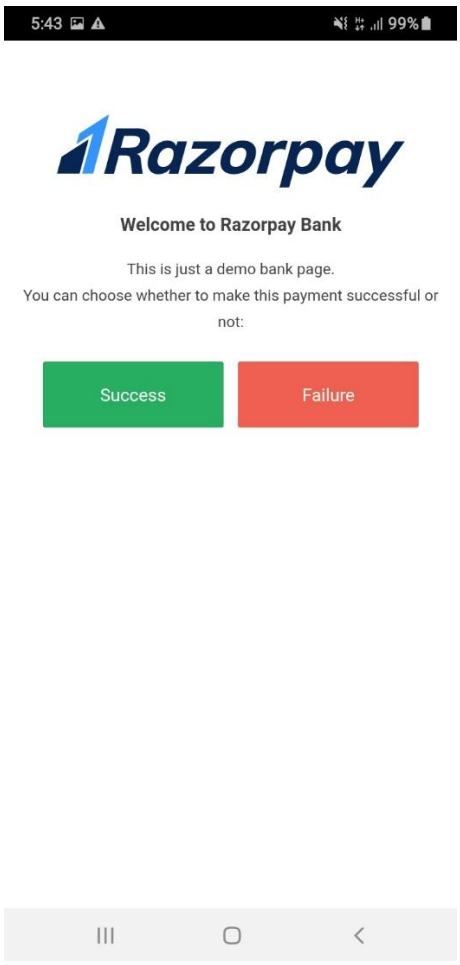
Feedback



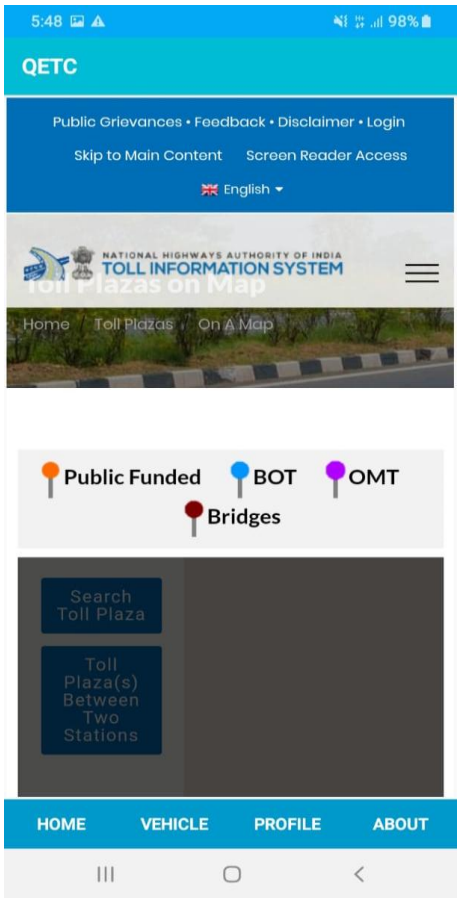
Add Fund/ Recharge



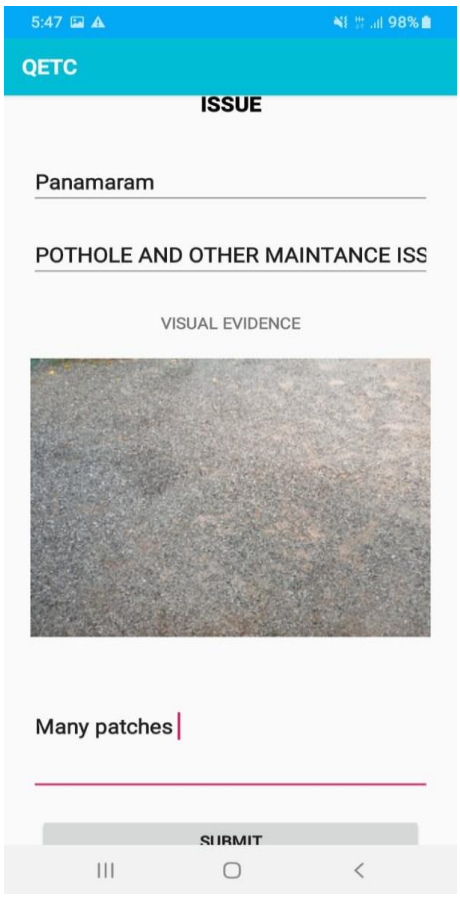

5:43
99%
Razorpay Corp
Demoing Charges
₹ 500
+919745866044 | joysjohnyt@gmail.com
Edit
CARDS, NETBANKING & MORE
Card
Visa, MasterCard, RuPay & More
Netbanking
All Indian banks
Wallet
Freecharge & More
UPI
Instant payment using UPI App
5:43
99%
Razorpay Corp
Demoing Charges
₹ 500
Netbanking
+919745866044
SBI
HDFC
ICICI
Axis
Kotak
Yes
State Bank of India
PAY ₹ 500

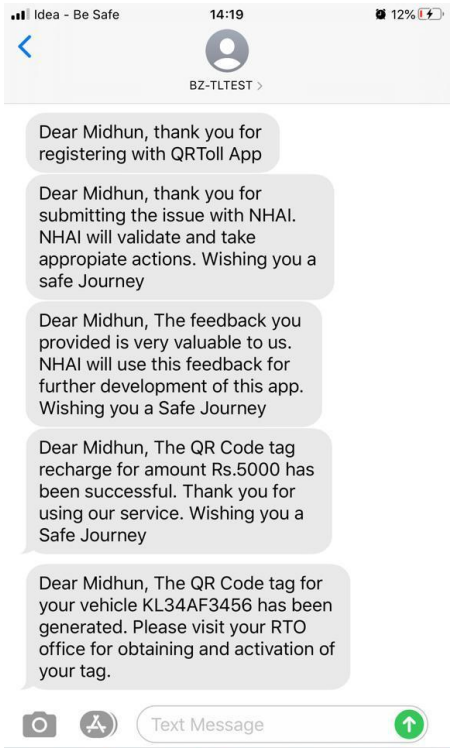


Near By Tolls

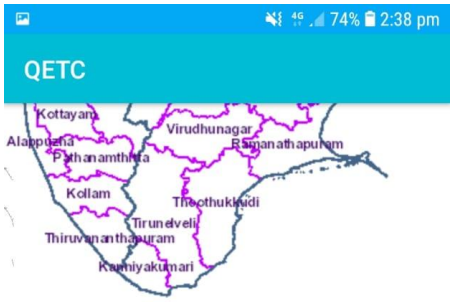


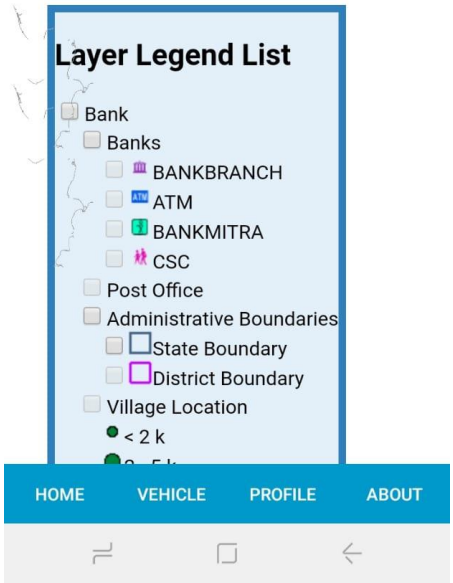
Report an Issue



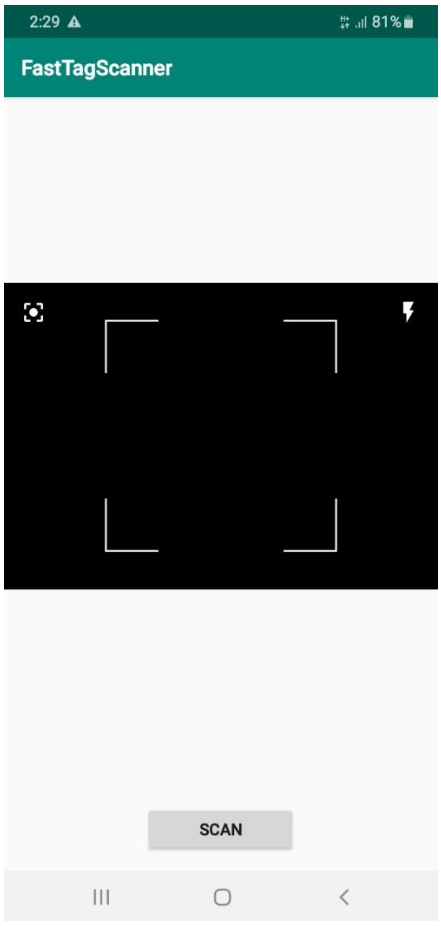

Idea - Be Safe
14:19
12%
BZ-TLTEST
Dear Midhun, thank you for registering with QRToll App
Dear Midhun, thank you for submitting the issue with NHAI. NHAI will validate and take appropiate actions. Wishing you a safe Journey
Dear Midhun, The feedback you provided is very valuable to us. NHAI will use this feedback for further development of this app. Wishing you a Safe Journey
Dear Midhun, The QR Code tag recharge for amount Rs.5000 has been successful. Thank you for using our service. Wishing you a Safe Journey
Dear Midhun, The QR Code tag for your vehicle KL34AF3456 has been generated. Please visit your RTO office for obtaining and activation of your tag.
Text Message

Bank/Atm


74% 2:38 pm
QETC
Kottayam
Virudhunagar
Alappuzha
Ramanathapuram
Pathanamthitta
Kollam
Thoothukkudi
Tirunelveli
Thiruvananthapuram
Kanniyakumari


Layer Legend List
Bank
Banks
BANKBRANCH
ATM
BANKMITRA
CSC
Post Office
Administrative Boundaries
State Boundary
District Boundary
Village Location
< 2 k
HOME
VEHICLE
PROFILE
ABOUT

Scanner


2:29
81%
FastTagScanner
SCAN


2:30
81%
FastTagScanner
Thank You!
SCAN

II. SAMPLE CODE

ABOUT

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class About extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
        Button btn_home =findViewById(R.id.btn_home);
        Button btn_vehicle =findViewById(R.id.btn_vehicle);
        Button btn_profile =findViewById(R.id.btn_profile);

        TextView about =findViewById(R.id.textView5);


        about.setText("ABOUT APP\n\n" +
                "This mobile App is for the National Highway Commuters of
India, who wish to cross through a National Toll road in a pre-informed manner.
The App also prevents frauds by way of overcharging by a Toll Plaza as it
informs the user about the fee applicable. While on move, users can know the
places of interest & Route to these places if he/she wishes.\n" +
                "\n" +
                "The App gives user complete freedom in generating QRCode Tags
for the number of vehicles. Once the user crosses a plaza, he or she can also
give feedback of his experience.\n" +
                "\n" +
                "The application consist of Nearby Toll where the user can
search for tolls on route. The application has been specifically made for the
users of QRCode Tag system.\n"+
               "\n"+
                "National Highway Authority of India [NHAI] in conjunction with
Ministry of Road & Transports, Government of India, has the complete authority
over the system and the application even though it was,\n"+
                "\n"+
                "DEVELOPED BY INDIAN MANUFACTURED ELECTRONICS \n");

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(About.this, MainActivity.class);
                startActivity(intent);
                finish();

            }
        });
        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(About.this, Vehicle.class);
```

```
                startActivity(intent);
                finish();

            }
        });

        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(About.this, Profile.class);
                startActivity(intent);
                finish();

            }
        });

    }
}
```

Add Vehicle

```
package com.mtlz.qetc;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;
import java.util.ArrayList;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;


public class AddVehicle extends AppCompatActivity {
    ImageView img_front,img_back,current_img,img_rc;
    TextView reg_number,owner_name,chasis_number;
```

```
    Spinner spinner_type;
    ProgressDialog progressDialog;
    int img_flag=0;
    //Uri to store the image uri
    Uri filePath;
    String FilePath1,FilePath2;
    String class_of_vehicle="CAR";
    Sms sms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_vehicle);

        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Uploading...");

        reg_number=findViewById(R.id.editTextVehicleReg);
        owner_name=findViewById(R.id.editText_owner);
        chasis_number=findViewById(R.id.editText_ch_no);


        Button generate_qr=findViewById(R.id.btn_generate_qr);
        img_front=findViewById(R.id.imageViewFront);
        img_back=findViewById(R.id.imageViewBack);
        img_rc=findViewById(R.id.imageViewRC);
        spinner_type = findViewById(R.id.spinner_vehicle_class);

        ArrayList<String> arrayList = new ArrayList<>();
        arrayList.add("Car, Jeep, Van");
        arrayList.add("Light Commercial Vehicle (LCV)");
        arrayList.add("Bus, Truck");
        arrayList.add("Multi axle");
        ArrayAdapter<String> arrayAdapter = new
ArrayAdapter<String>(this,android.R.layout.simple_spinner_item, arrayList);

arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_i
tem);
        spinner_type.setAdapter(arrayAdapter);

        img_front.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                current_img=img_front;
                selectImage(AddVehicle.this);
            }
        });

        img_back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                current_img=img_back;
                selectImage(AddVehicle.this);
            }
        });


        img_rc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                current_img=img_rc;
                selectImage(AddVehicle.this);
            }
```

```
        });


        spinner_type.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
                class_of_vehicle =
parent.getItemAtPosition(position).toString();


            }
            @Override
            public void onNothingSelected(AdapterView <?> parent) {
            }
        });

        generate_qr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.i("nnn","Class "+class_of_vehicle);
                Log.i("nnn","Uses "+GlobalVariables.EMAIL);


                uploadMultipleFiles();
            }
        });

    }

    private void selectImage(Context context) {
        final CharSequence[] options = {"Take Photo", "Choose from Gallery",
"Cancel"};

        AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setTitle("Choose Picture");

        builder.setItems(options, new DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int item) {

                if (options[item].equals("Take Photo")) {
                    Intent takePicture = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                    startActivityForResult(takePicture, 0);

                } else if (options[item].equals("Choose from Gallery")) {
                    Intent pickPhoto = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                    startActivityForResult(pickPhoto, 1);//one can be replaced
with any action code

                } else if (options[item].equals("Cancel")) {
                    dialog.dismiss();
                }
            }
        });
        builder.show();
    }
```

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode != RESULT_CANCELED) {
            switch (requestCode) {
                case 0:
                    if (resultCode == RESULT_OK && data != null) {
                        Bitmap selectedImage = (Bitmap)
data.getExtras().get("data");
                        current_img.setImageBitmap(selectedImage);
                        Uri tempUri = getImageUri(getApplicationContext(),
selectedImage);

                        String picturePath= new
File(getRealPathFromURI(tempUri)).getPath();

                        if(current_img==img_front)
                        {
                            FilePath1=picturePath;
                        }
                        else if(current_img==img_back)
                        {
                            FilePath2=picturePath;
                        }


                        Log.i("nnn", "111"+picturePath );
                    }

                    break;
                case 1:
                    if (resultCode == RESULT_OK && data != null) {

                        filePath = data.getData();
                        Log.i("nnn", "222");

                        Uri selectedImage = data.getData();
                        String[] filePathColumn =
{MediaStore.Images.Media.DATA};
                        if (selectedImage != null) {
                            Cursor cursor =
getContentResolver().query(selectedImage,
                                    filePathColumn, null, null, null);
                            if (cursor != null) {
                                cursor.moveToFirst();


                                int columnIndex =
cursor.getColumnIndex(filePathColumn[0]);
                                String picturePath =
cursor.getString(columnIndex);


                                if(current_img==img_front)
                                {
                                    FilePath1=picturePath;

                                    Log.i("nnn","Front !"+picturePath);
                                }
                                else if(current_img==img_back)
```

```
                                {
                                    FilePath2=picturePath;
                                    Log.i("nnn","Back !"+picturePath);
                                }


current_img.setImageBitmap(BitmapFactory.decodeFile(picturePath));
                                cursor.close();
                            }
                        }

                    }
                    break;
                }
            }
        }


    public Uri getImageUri(Context inContext, Bitmap inImage) {
        String path =
MediaStore.Images.Media.insertImage(inContext.getContentResolver(), inImage,
"VFD", null);
        return Uri.parse(path);
    }

    public String getRealPathFromURI(Uri uri) {
        String path = "";
        if (getContentResolver() != null) {
            Cursor cursor = getContentResolver().query(uri, null, null, null,
null);
            if (cursor != null) {
                cursor.moveToFirst();
                int idx =
cursor.getColumnIndex(MediaStore.Images.ImageColumns.DATA);
                path = cursor.getString(idx);
                cursor.close();
            }
        }
        return path;
    }
    private void uploadMultipleFiles() {
        //  progressDialog.show();


        Log.i("nnn", "Path1= " + FilePath1);
        Log.i("nnn", "Path2= " + FilePath2);

        String reg = reg_number.getText().toString();
        String ownr_name = owner_name.getText().toString();
        String ch_no= chasis_number.getText().toString();

        GlobalVariables.REG_NO=reg;



        if (reg.isEmpty()|| ownr_name.isEmpty()|| ch_no.isEmpty()) {
            Toast.makeText(AddVehicle.this, "Fill all fields!",
Toast.LENGTH_SHORT).show();

        } else
```

```
        {
            if (FilePath1 != null && FilePath2 != null ) {
                Log.i("nnn", "OK");

                // Map is used to multipart the file using okhttp3.RequestBody
                File file1 = new File(FilePath1);
                File file2 = new File(FilePath2);

                // Parsing any Media type file
                RequestBody requestBody1 =
RequestBody.create(MediaType.parse("*/*"), file1);
                RequestBody requestBody2 =
RequestBody.create(MediaType.parse("*/*"), file2);

                MultipartBody.Part fileToUpload1 =
MultipartBody.Part.createFormData("file1", file1.getName(), requestBody1);
                MultipartBody.Part fileToUpload2 =
MultipartBody.Part.createFormData("file2", file2.getName(), requestBody2);

                ApiConfig getResponse =
AppConfig.getRetrofit().create(ApiConfig.class);
                Call<ServerResponse> call =
getResponse.uploadMulFile(fileToUpload1, fileToUpload2, GlobalVariables.EMAIL,
reg, ownr_name,ch_no, class_of_vehicle);
                call.enqueue(new Callback<ServerResponse>() {
                    @Override
                    public void onResponse(Call<ServerResponse> call,
Response<ServerResponse> response) {
                        ServerResponse serverResponse = response.body();



                        if (serverResponse != null) {
                            if (serverResponse.getSuccess()) {

                                sms = new Sms(AddVehicle.this);
                                sms.send(GlobalVariables.PHONE, "Dear " +
GlobalVariables.user + ", The QR Code tag for your vehicle
"+GlobalVariables.REG_NO+ " has been generated. Please visit your RTO office
for obtaining and activation of your tag.");


                                Toast.makeText(getApplicationContext(),
serverResponse.getMessage(), Toast.LENGTH_SHORT).show();
                                Log.i("nnn", "Response1:
"+serverResponse.getMessage());
                                Intent intent = new Intent(AddVehicle.this,
ViewQr.class);
                                startActivity(intent);
                                finish();

                            } else {
                                Toast.makeText(getApplicationContext(),
serverResponse.getMessage(), Toast.LENGTH_SHORT).show();
                                Log.i("nnn", "Response2:
"+serverResponse.getMessage());
                            }
                        } else {
                            assert serverResponse != null;
                            Log.i("nnn", "Response3:
"+serverResponse.getMessage());
                        }
```

```
                        progressDialog.dismiss();
                    }

                    @Override
                    public void onFailure(Call<ServerResponse> call, Throwable
t) {
                        Log.i("nnn", "Failed");
                        Log.i("nnn","Response "+t.getMessage());
                    }
                });


            } else {
                Toast.makeText(AddVehicle.this, "Choose images!",
Toast.LENGTH_SHORT).show();
            }

        }



    }

}
```

BANK

```
package com.mtlz.qetc;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class Bank extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bank);

        Button btn_home =findViewById(R.id.btn_home);
        Button btn_vehicle =findViewById(R.id.btn_vehicle);
        Button btn_about =findViewById(R.id.btn_about);
        Button btn_profile =findViewById(R.id.btn_profile);

        WebView myWebView = findViewById(R.id.bank_view);
        WebSettings webSettings = myWebView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        myWebView.getSettings().setJavaScriptCanOpenWindowsAutomatically(true);
        myWebView.loadUrl("http://findmybank.gov.in/FMB/");

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Bank.this, MainActivity.class);
                startActivity(intent);
                finish();
```

```
            }
        });
        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Bank.this, Vehicle.class);
                startActivity(intent);
                finish();

            }
        });


        btn_about.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Bank.this, About.class);
                startActivity(intent);
                finish();

            }
        });
        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Bank.this, About.class);
                startActivity(intent);
                finish();

            }
        });
    }
}
```

EMERGENCY

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

public class Emergency extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_emergency);
```

```
        Button btn_home = findViewById(R.id.btn_home);
        Button btn_vehicle = findViewById(R.id.btn_vehicle);
        Button btn_about = findViewById(R.id.btn_about);
        Button btn_profile = findViewById(R.id.btn_profile);


        Button btn_call_1033 = findViewById(R.id.btn_call_highway);
        Button btn_call_100 = findViewById(R.id.btn_call_police);
        Button btn_call_112 = findViewById(R.id.btn_call_fire);
        Button btn_call_102 = findViewById(R.id.btn_call_ambulance);
        Button btn_call_1254 = findViewById(R.id.btn_call_child);

        btn_call_1033.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View v) {

                Log.i("nnn", "Calling");
                Intent intent = new Intent(Intent.ACTION_CALL);
                intent.setData(Uri.parse("tel:+919207179161"));
                if (ActivityCompat.checkSelfPermission(Emergency.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                    // TODO: Consider calling
                    //    ActivityCompat#requestPermissions
                    // here to request the missing permissions, and then
overriding
                    //   public void onRequestPermissionsResult(int
requestCode, String[] permissions,
                    //                                          int[]
grantResults)
                    // to handle the case where the user grants the permission.
See the documentation
                    // for ActivityCompat#requestPermissions for more details.
                    Log.i("nnn", "Enable permission");
                    return;
                }
                startActivity(intent);

            }
        });

        btn_call_100.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View v) {

                Log.i("nnn", "Calling");
                Intent intent = new Intent(Intent.ACTION_CALL);
                intent.setData(Uri.parse("tel:+919207179161"));
                if (ActivityCompat.checkSelfPermission(Emergency.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                    // TODO: Consider calling
                    //    ActivityCompat#requestPermissions
                    // here to request the missing permissions, and then
overriding
                    //   public void onRequestPermissionsResult(int
requestCode, String[] permissions,
                    //                                          int[]
grantResults)
                    // to handle the case where the user grants the permission.
See the documentation
```

```
                    // for ActivityCompat#requestPermissions for more details.
                    Log.i("nnn", "Enable permission");
                    return;
                }
                startActivity(intent);

            }
        });
        btn_call_112.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View v) {

                Log.i("nnn", "Calling");
                Intent intent = new Intent(Intent.ACTION_CALL);
                intent.setData(Uri.parse("tel:+919207179161"));
                if (ActivityCompat.checkSelfPermission(Emergency.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                    // TODO: Consider calling
                    //    ActivityCompat#requestPermissions
                    // here to request the missing permissions, and then
overriding
                    //   public void onRequestPermissionsResult(int
requestCode, String[] permissions,
                    //                                          int[]
grantResults)
                    // to handle the case where the user grants the permission.
See the documentation
                    // for ActivityCompat#requestPermissions for more details.
                    Log.i("nnn", "Enable permission");
                    return;
                }
                startActivity(intent);

            }
        });
        btn_call_102.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View v) {

                Log.i("nnn", "Calling");
                Intent intent = new Intent(Intent.ACTION_CALL);
                intent.setData(Uri.parse("tel:+919207179161"));
                if (ActivityCompat.checkSelfPermission(Emergency.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                    // TODO: Consider calling
                    //    ActivityCompat#requestPermissions
                    // here to request the missing permissions, and then
overriding
                    //   public void onRequestPermissionsResult(int
requestCode, String[] permissions,
                    //                                          int[]
grantResults)
                    // to handle the case where the user grants the permission.
See the documentation
                    // for ActivityCompat#requestPermissions for more details.
                    Log.i("nnn", "Enable permission");
                    return;
                }
                startActivity(intent);
```

```
            }
        });

        btn_call_1254.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View v) {

                Log.i("nnn", "Calling");
                Intent intent = new Intent(Intent.ACTION_CALL);
                intent.setData(Uri.parse("tel:+919207179161"));
                if (ActivityCompat.checkSelfPermission(Emergency.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                    // TODO: Consider calling
                    //    ActivityCompat#requestPermissions
                    // here to request the missing permissions, and then
overriding
                    //   public void onRequestPermissionsResult(int
requestCode, String[] permissions,
                    //                                          int[]
grantResults)
                    // to handle the case where the user grants the permission.
See the documentation
                    // for ActivityCompat#requestPermissions for more details.
                    Log.i("nnn", "Enable permission");
                    return;
                }
                startActivity(intent);

            }
        });

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Emergency.this, MainActivity.class);
                startActivity(intent);
                finish();

            }
        });
        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Emergency.this, Vehicle.class);
                startActivity(intent);
                finish();

            }
        });

        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Emergency.this, Profile.class);
                startActivity(intent);

            }
        });
```

```
        btn_about.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Emergency.this, About.class);
                startActivity(intent);
                finish();

            }
        });
    }
}
```

FEEDBACK

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;

import android.content.Intent;
import android.graphics.BitmapFactory;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

public class Feedback extends AppCompatActivity {
    ImageView R1,R2,R3,R4;
    Sms sms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_feedback);

        R1=findViewById(R.id.imageViewR1);
        R2=findViewById(R.id.imageViewR2);
        R3=findViewById(R.id.imageViewR3);
        R4=findViewById(R.id.imageViewR4);



        final EditText name=findViewById(R.id.editTextName);
        final EditText phone=findViewById(R.id.editTextPhone);
        final EditText com=findViewById(R.id.editTextComment);

        Button btn_submit =findViewById(R.id.buttonSubmit);


        Button btn_home =findViewById(R.id.btn_home);
        Button btn_about =findViewById(R.id.btn_about);
        Button btn_profile =findViewById(R.id.btn_profile);
        Button btn_vehicle =findViewById(R.id.btn_vehicle);



        R1.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View v) {
                String PACKAGE_NAME = getApplicationContext().getPackageName();
                int imgId =
getResources().getIdentifier(PACKAGE_NAME+":drawable/btn_star_big_on" , null,
null);
                R1.setImageResource(android.R.drawable.btn_star_big_on);
                R2.setImageResource(android.R.drawable.btn_star_big_off);
                R3.setImageResource(android.R.drawable.btn_star_big_off);
                R4.setImageResource(android.R.drawable.btn_star_big_off);
            }
        });

        R2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String PACKAGE_NAME = getApplicationContext().getPackageName();
                int imgId =
getResources().getIdentifier(PACKAGE_NAME+":drawable/btn_star_big_on" , null,
null);
                R1.setImageResource(android.R.drawable.btn_star_big_on);
                R2.setImageResource(android.R.drawable.btn_star_big_on);
                R3.setImageResource(android.R.drawable.btn_star_big_off);
                R4.setImageResource(android.R.drawable.btn_star_big_off);
            }
        });
        R3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String PACKAGE_NAME = getApplicationContext().getPackageName();
                int imgId =
getResources().getIdentifier(PACKAGE_NAME+":drawable/btn_star_big_on" , null,
null);
                R1.setImageResource(android.R.drawable.btn_star_big_on);
                R2.setImageResource(android.R.drawable.btn_star_big_on);
                R3.setImageResource(android.R.drawable.btn_star_big_on);
                R4.setImageResource(android.R.drawable.btn_star_big_off);
            }
        });

        R4.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String PACKAGE_NAME = getApplicationContext().getPackageName();
                int imgId =
getResources().getIdentifier(PACKAGE_NAME+":drawable/btn_star_big_on" , null,
null);
                R1.setImageResource(android.R.drawable.btn_star_big_on);
                R2.setImageResource(android.R.drawable.btn_star_big_on);
                R3.setImageResource(android.R.drawable.btn_star_big_on);
                R4.setImageResource(android.R.drawable.btn_star_big_on);
            }
        });


        btn_submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String head_txt =name.getText().toString();
                String loc_txt = phone.getText().toString();
                String com_txt = com.getText().toString();
```

```
                if(head_txt.isEmpty()|| loc_txt.isEmpty()||com_txt.isEmpty())
                {
                    Toast.makeText(Feedback.this, "Fill All !",
Toast.LENGTH_SHORT).show();
                }

                else
                {
                    Toast.makeText(Feedback.this, "Submitted !",
Toast.LENGTH_SHORT).show();
                    sms = new Sms(Feedback.this);
                    sms.send(GlobalVariables.PHONE, "Dear " +
GlobalVariables.user + ", The feedback you provided is very valuable to us.
NHAI will use this feedback for further development of this app. Wishing you a
Safe Journey");

                }

            }
        });

        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Feedback.this, Vehicle.class);
                startActivity(intent);
                finish();

            }
        });

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Feedback.this, MainActivity.class);
                startActivity(intent);
                finish();

            }
        });


        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Feedback.this, Profile.class);
                startActivity(intent);
                finish();

            }
        });
        btn_about.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Feedback.this, About.class);
                startActivity(intent);
                finish();
```

```
            }
        });
    }
}
```

GLOBAL VARIABLES

```
package com.mtlz.qetc;

import android.widget.TextView;

public class GlobalVariables {

    static String user="Friend";

  //  static String host="http://192.168.0.7/";
    static String host="http://qetc.mcompany.in/";
    static String URL_REG= host+ "reg.php";
    static String URL_LOGIN= host+ "login.php";
    static String URL_GET_BALANCE=host+"select_amount.php";
    static String URL_UPDATE_AMOUNT=host+"update_amount.php";

    static String URL_GET_PROFILE=host+"get_profile.php";

    static String URL_CHANGE_PASSWORD=host+"change_password.php";

    static String URL_GET_VEHICLE_LIST=host+"get_vehicle_list.php?email=";

    static String URL_GET_SUMMARY=host+"get_summary.php?email=";

    static String
URL_GET_VEHICLE_DETAILS=host+"get_vehicle_details.php?reg_no=";

    static String URL_DELETE_VEHICLE=host+"delete_vehicle.php?reg_no=";

    static String URL_SMS=host+"send_sms.php";



    static String REG_NO="";

    static String ISSUE_HEAD="";

    static String EMAIL="";

    static String PHONE="";



    static Double BALANCE=0.0;

    static String FROM_DATE="";


    public static TextView from,to;

    static int flag=0;

}
```

ISSUE

```
package com.mtlz.qetc;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import java.io.File;

public class Issue extends AppCompatActivity {
    ImageView img_visual,current_img;
    Uri filePath;
    Sms sms;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_issue);
        final EditText head=findViewById(R.id.editText_head);
        final EditText loc=findViewById(R.id.editText_loc);
        final EditText com=findViewById(R.id.editText_comment);
        head.setText(GlobalVariables.ISSUE_HEAD);

        img_visual=findViewById(R.id.imageViewVisual);

        Button submit=findViewById(R.id.buttonSubmit);
        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String head_txt =head.getText().toString();
                String loc_txt = loc.getText().toString();
                String com_txt = com.getText().toString();


                if(head_txt.isEmpty()|| loc_txt.isEmpty()||com_txt.isEmpty())
                {
                    Toast.makeText(Issue.this, "Fill All !",
Toast.LENGTH_SHORT).show();
                }

                else if (filePath == null  )
                {
                    Toast.makeText(Issue.this, "Choose Image !",
Toast.LENGTH_SHORT).show();
                }
```

```
                else
                {
                    Toast.makeText(Issue.this, "Submitted !",
Toast.LENGTH_SHORT).show();
                    sms = new Sms(Issue.this);
                    sms.send(GlobalVariables.PHONE, "Dear " +
GlobalVariables.user + ", thank you for submitting the issue with NHAI. NHAI
will validate and take appropiate actions. Wishing you a safe Journey");
                    Intent intent = new Intent(Issue.this, MainActivity.class);
                    startActivity(intent);
                    finish();

                }

            }
        });

        img_visual.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                current_img=img_visual;
                selectImage(Issue.this);
            }
        });

    }

    private void selectImage(Context context) {
        final CharSequence[] options = {"Take Photo", "Choose from Gallery",
"Cancel"};

        AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setTitle("Choose Picture");

        builder.setItems(options, new DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int item) {

                if (options[item].equals("Take Photo")) {
                    Intent takePicture = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                    startActivityForResult(takePicture, 0);

                } else if (options[item].equals("Choose from Gallery")) {
                    Intent pickPhoto = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                    startActivityForResult(pickPhoto, 1);//one can be replaced
with any action code

                } else if (options[item].equals("Cancel")) {
                    dialog.dismiss();
                }
            }
        });
        builder.show();
    }


    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
```

```
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode != RESULT_CANCELED) {
            switch (requestCode) {
                case 0:
                    if (resultCode == RESULT_OK && data != null) {
                        Bitmap selectedImage = (Bitmap)
data.getExtras().get("data");
                        current_img.setImageBitmap(selectedImage);
                        Uri tempUri = getImageUri(getApplicationContext(),
selectedImage);

                        String picturePath= new
File(getRealPathFromURI(tempUri)).getPath();



                        Log.i("nnn", "111"+picturePath );
                    }

                    break;
                case 1:
                    if (resultCode == RESULT_OK && data != null) {

                        filePath = data.getData();
                        Log.i("nnn", "222");

                        Uri selectedImage = data.getData();
                        String[] filePathColumn =
{MediaStore.Images.Media.DATA};
                        if (selectedImage != null) {
                            Cursor cursor =
getContentResolver().query(selectedImage,
                                    filePathColumn, null, null, null);
                            if (cursor != null) {
                                cursor.moveToFirst();


                                int columnIndex =
cursor.getColumnIndex(filePathColumn[0]);
                                String picturePath =
cursor.getString(columnIndex);



current_img.setImageBitmap(BitmapFactory.decodeFile(picturePath));
                                cursor.close();
                            }
                        }

                    }
                    break;
            }
        }
    }



    public Uri getImageUri(Context inContext, Bitmap inImage) {
        String path =
MediaStore.Images.Media.insertImage(inContext.getContentResolver(), inImage,
"VFD", null);
        return Uri.parse(path);
```

```
    }

    public String getRealPathFromURI(Uri uri) {
        String path = "";
        if (getContentResolver() != null) {
            Cursor cursor = getContentResolver().query(uri, null, null, null,
null);
            if (cursor != null) {
                cursor.moveToFirst();
                int idx =
cursor.getColumnIndex(MediaStore.Images.ImageColumns.DATA);
                path = cursor.getString(idx);
                cursor.close();
            }
        }
        return path;
    }




}
```

LOGIN

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class Login extends AppCompatActivity {
    EditText lpswd,lmail;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
```

```
        lmail = findViewById(R.id.txt_login_email);
        lpswd = findViewById(R.id.txt_login_password);

        Button login =findViewById(R.id.btn_login);
        Button signup=findViewById(R.id.btn_new_sign_up);


        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(!lmail.getText().toString().equals(" "))
                {

if(Patterns.EMAIL_ADDRESS.matcher(lmail.getText().toString()).matches() )
                    {
                        Toast.makeText(Login.this, "Email is VALID.",
Toast.LENGTH_SHORT).show();

                    }
                    else
                    {
                        Toast.makeText(Login.this, "Email is INVALID.",
Toast.LENGTH_SHORT).show();
                    }
                }
                else
                {
                    Toast.makeText(Login.this, "Email is Needed.",
Toast.LENGTH_SHORT).show();
                }

if((lmail.getText().toString().isEmpty())||(lpswd.getText().toString().isEmpty(
)))
                {
                    Toast.makeText(Login.this,"All fields are Mandatory!",
Toast.LENGTH_SHORT).show();
                }
                else
                {
                    login();
                }
            }
        });
        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent  intent = new Intent(Login.this, Register.class);
                startActivity(intent);

            }
        });
    }

    public void login()
    {
        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_LOGIN,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        Log.i("nnn","Login");
```

```
                        Log.i("nnn","Response="+response);
//If we are getting success from server
                        String sts="",phone1="";
                        try {
                            JSONObject json_obj= new JSONObject(response);
                            sts=json_obj.getString("status");
                           phone1=json_obj.getString("phone");
                           GlobalVariables.user=json_obj.getString("user");

                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                        if(sts.contains("login successful"))
                        {
                            GlobalVariables.EMAIL=lmail.getText().toString();
                            GlobalVariables.PHONE=phone1;
                            Toast.makeText(Login.this, sts,
Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(Login.this,
MainActivity.class);
                            startActivity(intent);


                            finish();
                        }

                        else

                        {

                            Toast.makeText(Login.this,"Invalid
Login",Toast.LENGTH_SHORT).show();
                        }

                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
//You can handle error here if you want
                    }

                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
//Adding parameters t o request

                params.put("email", lmail.getText().toString());
                params.put("password",lpswd.getText().toString());
//returning parameter
                return params;
            }
        };
//Adding the string request to the queue
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }


}
```

MAIN ACTIVITY

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {
    private ProgressDialog progress;
    TextView balance_amt_disp;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        balance_amt_disp=findViewById(R.id.balance_view);
        get_data();
        Button logout =findViewById(R.id.btn_logout);
        Button btn_add_fund =findViewById(R.id.btn_add_fund);

        Button btn_near_by =findViewById(R.id.near_by_btn);

        Button btn_bnk =findViewById(R.id.btn_bank);
        Button btn_report_issue =findViewById(R.id.btn_report_issue);
        Button btn_energency =findViewById(R.id.btn_emergency);

        Button btn_feedback =findViewById(R.id.buttonFeedback);
        Button btn_summary =findViewById(R.id.buttonSummary);


        Button btn_vehicle =findViewById(R.id.btn_vehicle);
        Button btn_about =findViewById(R.id.btn_about);
        Button btn_profile =findViewById(R.id.btn_profile);


        btn_summary.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, Summary.class);
                startActivity(intent);
                finish();
```

```
        }
    });


    btn_feedback.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, Feedback.class);
            startActivity(intent);

        }
    });



    btn_add_fund.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, Recharge.class);
            startActivity(intent);
            finish();
        }
    });



    btn_energency.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, Emergency.class);
            startActivity(intent);
            finish();
        }
    });


    logout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, Login.class);
            startActivity(intent);
            finish();
        }
    });

    btn_near_by.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intent = new Intent(MainActivity.this, NearBy.class);
            startActivity(intent);
            finish();

        }
    });

    btn_bnk.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intent = new Intent(MainActivity.this, Bank.class);
```

```
                startActivity(intent);
                finish();

            }
        });

        btn_report_issue.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(MainActivity.this,
ReportIssue.class);
                startActivity(intent);
                finish();
            }
        });




        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(MainActivity.this, Vehicle.class);
                startActivity(intent);

            }
        });

        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(MainActivity.this, Profile.class);
                startActivity(intent);

            }
        });
        btn_about.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(MainActivity.this, About.class);
                startActivity(intent);

            }
        });

    }


    public void get_data()
    {

        progress=new ProgressDialog(this);
        progress.setMessage("Please Wait..");
        progress.show();

        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_GET_BALANCE,
```

```
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        progress.hide();
//If we are getting success from server
                        Log.i("nnn",response);
                        String balance="";
                        try {
                            JSONObject json_obj= new JSONObject(response);
                            balance=json_obj.getString("amount");
                            Log.i("nnn","STS="+balance);

                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                        if(response.contains("amount"))
                        {
                            Log.i("nnn","Success");
                            try{

GlobalVariables.BALANCE=Double.parseDouble(balance);
                                balance_amt_disp.setText(balance);

                            } catch (Exception e) {
                                e.printStackTrace();
                            }

                            // Intent intent = new Intent(Login.this,
Balance.class);
                            // startActivity(intent);
                        }

                        else

                        {
                            Log.i("nnn","Error");

Toast.makeText(MainActivity.this,"Error",Toast.LENGTH_SHORT).show();
                        }

                    }
                },
                new Response.ErrorListener() {

                    @Override
                    public void onErrorResponse(VolleyError error) {
//You can handle error here if you want
                        progress.hide();
                    }

                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
//Adding parameters t o request

                params.put("email", GlobalVariables.EMAIL);
//returning parameter
                return params;
            }
        };
//Adding the string request to the queue
```

```
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }

}
```

NEARBY

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.Button;

public class NearBy extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_near_by);

        Button btn_home =findViewById(R.id.btn_home);
        Button btn_vehicle =findViewById(R.id.btn_vehicle);
        Button btn_about =findViewById(R.id.btn_about);
        Button btn_profile =findViewById(R.id.btn_profile);

        WebView myWebView = findViewById(R.id.near_by_view);
        WebSettings webSettings = myWebView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        myWebView.getSettings().setJavaScriptCanOpenWindowsAutomatically(true);
        myWebView.loadUrl("http://tis.nhai.gov.in/tollplazasonmap");

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(NearBy.this, MainActivity.class);
                startActivity(intent);
                finish();

            }
        });
        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(NearBy.this, Vehicle.class);
                startActivity(intent);
                finish();

            }
        });


        btn_about.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View v) {

                Intent intent = new Intent(NearBy.this, About.class);
                startActivity(intent);
                finish();

            }
        });
        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(NearBy.this, About.class);
                startActivity(intent);
                finish();

            }
        });
    }
}
```

PROFILE

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;
import org.w3c.dom.Text;

import java.util.HashMap;
import java.util.Map;

public class Profile extends AppCompatActivity {
    TextView name,email,phone,owned;
    EditText password;
    int flag=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);
```

```
        Button btn_home =findViewById(R.id.btn_home);
        Button btn_vehicle =findViewById(R.id.btn_vehicle);
        Button btn_about =findViewById(R.id.btn_about);
        final Button btn_change_password =findViewById(R.id.btn_change_pass);


        name=findViewById(R.id.textView_name);
        email=findViewById(R.id.textView_email);
        password=findViewById(R.id.editText_password);
        phone=findViewById(R.id.textView_phone);
        owned=findViewById(R.id.textView_owned);

        password.setFocusable(false);




        get_details();

        btn_change_password.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(flag==0)
                {
                    flag=1;
                    btn_change_password.setText("SAVE");
                    password.setFocusableInTouchMode(true);
                }
                else
                {
                    btn_change_password.setText("CHANGE");
                    flag=0;
                    change_pass();
                    password.setFocusable(false);
                }

            }
        });

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Profile.this, MainActivity.class);
                startActivity(intent);
                finish();

            }
        });
        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Profile.this, Vehicle.class);
                startActivity(intent);
                finish();

            }
        });


        btn_about.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Profile.this, About.class);
                startActivity(intent);
                finish();

            }
        });
    }



    public void get_details()
    {
        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_GET_PROFILE,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        Log.i("nnn","Response="+response);
                        try {
                            JSONObject json_obj= new JSONObject(response);

name.setText("Name:"+json_obj.getString("first_name"));
                            email.setText("Email: "+GlobalVariables.EMAIL);
                            password.setText(json_obj.getString("password"));
                            phone.setText("Phone
:"+json_obj.getString("phone"));
                            owned.setText("VEHICLE OWNED:
"+json_obj.getString("v_count"));



                        } catch (JSONException e) {
                            e.printStackTrace();
                        }



                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
//You can handle error here if you want
                    }

                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
//Adding parameters t o request

                params.put("email", GlobalVariables.EMAIL);
//returning parameter
                return params;
            }
        };
//Adding the string request to the queue
        RequestQueue requestQueue = Volley.newRequestQueue(this);
```

```
        requestQueue.add(stringRequest);
    }
    public void change_pass()
    {
        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_CHANGE_PASSWORD,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        String sts="";
                        try {
                            JSONObject json_obj= new JSONObject(response);
                            sts=json_obj.getString("status");

                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                        if(sts.contains("success"))
                        {
                            Toast.makeText(Profile.this,"Password Changed!",
Toast.LENGTH_LONG).show();

                        }



                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
//You can handle error here if you want
                    }

                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
//Adding parameters t o request

                params.put("email", GlobalVariables.EMAIL);
                params.put("password", password.getText().toString());
//returning parameter
                return params;
            }
        };
//Adding the string request to the queue
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }

}
```

RECHARGE

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.razorpay.Checkout;
import com.razorpay.PaymentResultListener;

import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class Recharge extends AppCompatActivity implements
PaymentResultListener {
    EditText amount_txt;
    Sms sms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recharge);
        amount_txt=findViewById(R.id.editText_amount);
        Button recharge =findViewById(R.id.btn_recharge);
        recharge.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startPayment();
            }
        });
    }


    public void startPayment() {
        /**
         * You need to pass current activity in order to let Razorpay create
CheckoutActivity
         */
        final Activity activity = this;

        final Checkout co = new Checkout();

        try {
            JSONObject options = new JSONObject();
            options.put("name", "Razorpay Corp");
            options.put("description", "Demoing Charges");
            //You can omit the image option to fetch the image from dashboard
            options.put("image", "https://rzp-
mobile.s3.amazonaws.com/images/rzp.png");
            options.put("currency", "INR");
            options.put("amount",
Integer.parseInt(amount_txt.getText().toString())*100);
            JSONObject preFill = new JSONObject();
            preFill.put("email", "sikander@gkmit.co");
```

```
            preFill.put("contact", "9680224241");
            options.put("prefill", preFill);

            co.open(activity, options);
        } catch (Exception e) {
            Toast.makeText(activity, "Error in payment: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
            e.printStackTrace();

        }


    }

    @Override
    public void onPaymentSuccess(final String razorpayPaymentID) {
        // Toast.makeText(this, "Payment successfully done! " +
razorpayPaymentID, Toast.LENGTH_SHORT).show();
        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_UPDATE_AMOUNT,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        Log.i("nnn","Response="+response);
//If we are getting success from server
                        if(response.contains("success"))
                        {
                            sms = new Sms(Recharge.this);
                            sms.send(GlobalVariables.PHONE, "Dear " +
GlobalVariables.user + ", The QR Code tag recharge for amount Rs."+
amount_txt.getText().toString() + " has been successful. Thank you for using
our service. Wishing you a Safe Journey");

                            Toast.makeText(Recharge.this,"Recharge
SuccessFully", Toast.LENGTH_SHORT).show();

GlobalVariables.BALANCE=GlobalVariables.BALANCE+Double.parseDouble(amount_txt.g
etText().toString());
                            Intent intent = new Intent(Recharge.this,
MainActivity.class);
                            startActivity(intent);
                            finish();



                        }
                        else
                        {
                            Toast.makeText(Recharge.this, "Payment Failed",
Toast.LENGTH_SHORT).show();
                        }
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
//You can handle error here if you want
                    }

                }){
            @Override
```

```
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
//Adding parameters t o request
                Double
amount=GlobalVariables.BALANCE+Double.parseDouble(amount_txt.getText().toString
());
                params.put("email",GlobalVariables.EMAIL);
                params.put("amount",amount.toString());

//returning parameter

                return params;
            }
        };

//Adding the string request to the queue
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);

    }

    @Override
    public void onPaymentError(int code, String response) {
        try {
            Toast.makeText(this, "Payment error please try again",
Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            Log.e("OnPaymentError", "Exception in onPaymentError", e);
        }
    }


}
```

REGISTER

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;
```

```
import java.util.HashMap;
import java.util.Map;
import java.util.regex.Matcher;
import java.util.regex.Pattern;


public class Register extends AppCompatActivity {

   EditText first_name,last_name,email,password,conpassword,phone;
    Sms sms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);


        Button signup = findViewById(R.id.btn_reg);
        Button cancel = findViewById(R.id.btn_cancel);

        first_name = findViewById(R.id.txt_first_name);
        last_name = findViewById(R.id.txt_last_name);
        email = findViewById(R.id.txt_email);
        password = findViewById(R.id.txt_password1);
        conpassword = findViewById(R.id.txt_password2);
        phone = findViewById(R.id.txt_phone);


        cancel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Register.this, Login.class);
                startActivity(intent);
                finish();
            }
        });

        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(!email.getText().toString().equals(" "))
                {

if(Patterns.EMAIL_ADDRESS.matcher(email.getText().toString()).matches() )
                    {
                        Toast.makeText(Register.this, "Email is VALID.",
Toast.LENGTH_SHORT).show();

                    }
                    else
                    {
                        Toast.makeText(Register.this, "Email is INVALID.",
Toast.LENGTH_SHORT).show();
                    }
                }
                else
                {
                    Toast.makeText(Register.this, "Email is Needed.",
Toast.LENGTH_SHORT).show();
                }

if((first_name.getText().toString().isEmpty())||(email.getText().toString().isE
mpty())||(password.getText().toString().isEmpty())||(conpassword.getText().toSt
```

```
ring().isEmpty())||(phone.getText().toString().isEmpty()))
                {
                    Toast.makeText(Register.this,"All Fields are Mandatory!",
Toast.LENGTH_SHORT).show();
                }
                else
if(!(password.getText().toString()).equals(conpassword.getText().toString()))
                {
                    Toast.makeText(Register.this, "Password Miss Matched",
Toast.LENGTH_SHORT).show();
                }
                else
                {
                    Register();
                }

            }
        });
    }

    public void Register() {
        StringRequest stringRequest = new StringRequest(Request.Method.POST,
GlobalVariables.URL_REG,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
//If we are getting success from server
                        Log.i("nnn","Respons" +response);
                        String status="0";
                        try {
                            JSONObject json_obj= new JSONObject(response);
                            status=json_obj.getString("status");

                            if(status.contains("registration successful")) {

                                sms = new Sms(Register.this);
                                sms.send(phone.getText().toString(), "Dear " +
first_name.getText().toString() + ", thank you for registering with QRToll
App");
                                Intent intent = new Intent(Register.this,
Login.class);
                                startActivity(intent);
                                finish();


                            }

                        } catch (JSONException e) {
                            e.printStackTrace();
                        }



                        Toast.makeText(Register.this, status,
Toast.LENGTH_SHORT).show();

                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
```

```
//You can handle error here if you want
                    }

                }) {
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String, String> params = new HashMap<>();
//Adding parameters to request
                params.put("first_name", first_name.getText().toString());
                params.put("last_name", last_name.getText().toString());
                params.put("email", email.getText().toString());
                params.put("password",password.getText().toString());
                params.put("phone", phone.getText().toString());
                params.put("amount", "0");
                params.put("status", "0");
//returning parameter
                return params;
            }
        };
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }

}
```

REPORT AN ISSUE

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class ReportIssue extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_report_issue);


        final Button btn_management=findViewById(R.id.btn_comp_toll);
        final Button btn_mint=findViewById(R.id.btn_comp_maint);
        final Button btn_saf=findViewById(R.id.btn_comp_safety);
        final Button btn_work=findViewById(R.id.btn_comp_work);
        Button btn_rules=findViewById(R.id.btn_rules);

        btn_management.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                GlobalVariables.ISSUE_HEAD=btn_management.getText().toString();
                Intent intent = new Intent(ReportIssue.this, Issue.class);
                startActivity(intent);

            }
        });
        btn_mint.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View v) {
                GlobalVariables.ISSUE_HEAD=btn_mint.getText().toString();
                Intent intent = new Intent(ReportIssue.this, Issue.class);
                startActivity(intent);
            }
        });
        btn_saf.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                GlobalVariables.ISSUE_HEAD=btn_saf.getText().toString();
                Intent intent = new Intent(ReportIssue.this, Issue.class);
                startActivity(intent);
            }
        });
        btn_work.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                GlobalVariables.ISSUE_HEAD=btn_work.getText().toString();
                Intent intent = new Intent(ReportIssue.this, Issue.class);
                startActivity(intent);
            }
        });
        btn_rules.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(ReportIssue.this, TollRules.class);
                startActivity(intent);

            }
        });



    }
}
```

SERVER RESPONSE

```
package com.mtlz.qetc;

import com.google.gson.annotations.SerializedName;

public class ServerResponse {
    // variable name should be same as in the json response from php
    @SerializedName("success")
    boolean success;
    @SerializedName("message")
    String message;
    String getMessage() {
        return message;
    }
    boolean getSuccess() {

        return success;
    }

}
```

SMS

```
package com.mtlz.qetc;

import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class Sms {
    Context mContext;

    public Sms(Context context){
        this.mContext = context;
    }

    public void send(final String phone, final String msg)
    {
        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_SMS,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        Log.i("nnn","Response="+response);

                        String sts="",message="";
                        try {
                            JSONObject json_obj= new JSONObject(response);
                            sts=json_obj.getString("status");
                            message=json_obj.getString("message");
                            Log.i("nnn","Message: "+message);


                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {

                    }

                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
```

```
                Map<String,String> params = new HashMap<>();


                Log.i("nnn","Phone: "+phone);
                Log.i("nnn","Phone: "+msg);
                params.put("mobile_no", phone);
                params.put("msg",msg);
                return params;
            }
        };

        RequestQueue requestQueue = Volley.newRequestQueue(this.mContext);
        requestQueue.add(stringRequest);
    }


}
```

SUMMARY

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.DialogFragment;

import android.annotation.SuppressLint;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.format.DateFormat;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Calendar;

public class Summary extends AppCompatActivity {
    ListView listView;
    ArrayList<String> summary_list = new ArrayList<>();
    ArrayAdapter<String> adapter;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_summary);

        listView=findViewById(R.id.summary_list);

        Button btn_home =findViewById(R.id.btn_home);
        Button btn_about =findViewById(R.id.btn_about);
        Button btn_profile =findViewById(R.id.btn_profile);
        Button btn_vehicle =findViewById(R.id.btn_vehicle);




        Button btn_show =findViewById(R.id.buttonShow);


        GlobalVariables.from=findViewById(R.id.textViewFrom);
        GlobalVariables.to=findViewById(R.id.textViewTo);


        btn_show.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                summary_list.clear();
                listView.setAdapter(null);
                getData();
            }
        });


        GlobalVariables.from.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.i("nnn","Clicked");
                GlobalVariables.flag=0;
                showDatePickerDialog();

            }
        });

        GlobalVariables.to.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.i("nnn","Clicked");
                GlobalVariables.flag=1;
                showDatePickerDialog();

            }
        });




        btn_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Summary.this, Vehicle.class);
```

```
                startActivity(intent);

            }
        });

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Summary.this, MainActivity.class);
                startActivity(intent);

            }
        });


        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Summary.this, Profile.class);
                startActivity(intent);

            }
        });
        btn_about.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Summary.this, About.class);
                startActivity(intent);

            }
        });
    }


    public void showDatePickerDialog() {
        DialogFragment newFragment = new DatePickerFragment();
        newFragment.show(getSupportFragmentManager(), "datePicker");
    }



    public static class DatePickerFragment extends DialogFragment
            implements DatePickerDialog.OnDateSetListener {

        @Override
        public Dialog onCreateDialog(Bundle savedInstanceState) {
            // Use the current date as the default date in the picker
            final Calendar c = Calendar.getInstance();
            int year = c.get(Calendar.YEAR);
            int month = c.get(Calendar.MONTH)+1;
            int day = c.get(Calendar.DAY_OF_MONTH);



            // Create a new instance of DatePickerDialog and return it
            return new DatePickerDialog(getActivity(), this, year, month, day);
        }

        public void onDateSet(DatePicker view, int year, int month, int day) {
```

```
            // Do something with the date chosen by the user

            if(GlobalVariables.flag==1)
            {
                GlobalVariables.to.setText(year+"-"+ month +"-"+day);
            }
            else {
                GlobalVariables.from.setText(year+"-"+ month +"-"+day);
            }



        }
    }



    private void getData()
    {

        String url =
GlobalVariables.URL_GET_SUMMARY+GlobalVariables.EMAIL+"&from="+GlobalVariables.
from.getText().toString()+"&to="+GlobalVariables.to.getText().toString();



        Log.i("nnn", "url="+url);
        StringRequest stringRequest = new StringRequest(url, new
Response.Listener<String>() {

            @Override
            public void onResponse(String response) {
                parseJsonData(response);

            }
        },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(Summary.this, error.getMessage(),
Toast.LENGTH_LONG).show();
                    }
                });

        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }

    public void parseJsonData(String response) {
        Log.i("nnn", "Response"+response);
        try {
            // Create the root JSONObject from the JSON string.

            JSONObject jsonRootObject = new JSONObject(response);

            //Get the instance of JSONArray that contains JSONObjects
            JSONArray jsonArray = jsonRootObject.optJSONArray("Result");

            Log.i("nnn", "Length : "+jsonArray.length());
            for (int i = 0; i < jsonArray.length(); i++)
            {
```

```
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                String log = "Reg NO: "+jsonObject.optString("reg_no")+"\nDate:
"+jsonObject.optString("date")+"\nAmount: "+jsonObject.optString("amount");
                summary_list.add(log);
            }

            adapter = new ArrayAdapter<String>(this,
                    android.R.layout.simple_list_item_1, android.R.id.text1,
summary_list);
            listView.setAdapter(adapter);



        } catch (JSONException e) {
            e.printStackTrace();
        }
    }


}
```

TOLLRULES

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class TollRules extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_toll_rules);




        TextView rules =findViewById(R.id.textView_rules);


        rules.setText(
                "CONDUCT TO FOLLOW :\n\n" +
                "1.\tHonking should be avoided under any circumstances. \n" +
                "2.\tLine departures should be made once you’re in one. \n" +
                "3.\tEmergency vehicles should restrict themselves to the
allocated lane. \n\n" +
                "TOLL PLAZA RULES\n\n" +
                "1.\tEmergency vehicles, Police and other diplomats have free
of charge transport through toll plazas\n" +
                "2.\tToll line should have only maximum of 6 vehicles per lane.
\n" +
                "3.\tIf vehicles exceeds rule 2 then Toll plaza management have
the authority to open a lane for free flow of traffic. \n" +
                "4.\tDiplomats, Ministers and other government officers,
Defense personal will have a reduction in toll charges across India for their
```

```
private vehicles. \n" +
                "5.\tThe rule 4 should be coincided with able document proof\n"
+
                "6.\tThe traffic control should be administered as per the
guidelines from police. \n" +
                "7.\tThe cost of each ticket should be displayed.\n" +
                "8.\tAn ambulance should be with every toll plaza for
emergencies. \n" +
                "9.\tThe toll collectors should be diligent and should show
courtesy to the customers. \n" +
                "10.\tThe fee should be collected manually and exact changes
should be tendered. \n" +
                "11.\tAny disputes at toll plaza should be directed to the
nearest police station\n" +
                "12.\tFailed to do so of rule 11, KPA and appropriate CrPC act
will be invoked against the toll management. \n" +
                "13.\tNot giving amble support the emergency vehicle will have
IPC act and KMVA proceedings against toll plaza. \n" +
                "14.\tAll govt. vehicles are free of charge to pass through the
toll plaza gates. \n\n\n");


    }
}
```

VEHICLE

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Vehicle extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_vehicle);
        Button btn_home =findViewById(R.id.btn_home);
        Button btn_about =findViewById(R.id.btn_about);
        Button btn_profile =findViewById(R.id.btn_profile);
        Button btn_view_vehicle =findViewById(R.id.btn_view_vhcl);
        Button btn_rules =findViewById(R.id.btn_toll_rule);



        Button btn_add_vehicle =findViewById(R.id.btn_add_vhcl);

        btn_home.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Vehicle.this, MainActivity.class);
                startActivity(intent);
                finish();

            }
```

```
        });


        btn_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Vehicle.this, Profile.class);
                startActivity(intent);
                finish();

            }
        });
        btn_about.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Vehicle.this, About.class);
                startActivity(intent);
                finish();

            }
        });



        btn_add_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Vehicle.this, AddVehicle.class);
                startActivity(intent);

            }
        });

        btn_view_vehicle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Vehicle.this, VehicleList.class);
                startActivity(intent);

            }
        });
        btn_rules.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Vehicle.this, TollRules.class);
                startActivity(intent);

            }
        });

    }
}
```

VEHICLE LIST

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class VehicleList extends AppCompatActivity {
    ListView listView;
    TextView textView;

    ArrayList<String> vehicle_list = new ArrayList<>();
    ArrayAdapter<String> adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_vehicle_list);


        getData();

        listView=findViewById(R.id.listView);
        textView=findViewById(R.id.textView);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int
position, long l) {
                // TODO Auto-generated method stub
                String reg=adapter.getItem(position);
                Log.i("nnn","Reg:"+reg);

                GlobalVariables.REG_NO=reg;


                Intent intent = new Intent(VehicleList.this,
ViewVehicle.class);
                startActivity(intent);
                finish();
```

```
            }
        });


    }


    private void getData()
    {

        String url =
GlobalVariables.URL_GET_VEHICLE_LIST+GlobalVariables.EMAIL;



        Log.i("nnn", "url="+url);
        StringRequest stringRequest = new StringRequest(url, new
Response.Listener<String>() {

            @Override
            public void onResponse(String response) {
                parseJsonData(response);

            }
        },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(VehicleList.this, error.getMessage(),
Toast.LENGTH_LONG).show();
                    }
                });

        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }

    public void parseJsonData(String response) {
        Log.i("nnn", "Response"+response);
        try {
            // Create the root JSONObject from the JSON string.

            JSONObject jsonRootObject = new JSONObject(response);

            //Get the instance of JSONArray that contains JSONObjects
            JSONArray jsonArray = jsonRootObject.optJSONArray("Result");


            for (int i = 0; i < jsonArray.length(); i++)
            {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                String reg_no = jsonObject.optString("reg_no");
                Log.i("nnn", "reg_no: "+reg_no);
                vehicle_list.add(reg_no);
            }

            adapter = new ArrayAdapter<String>(this,
                    android.R.layout.simple_list_item_1, android.R.id.text1,
vehicle_list);
            listView.setAdapter(adapter);
```

```
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

VIEW QR

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;

import java.io.File;
import java.security.MessageDigest;
import java.util.Timer;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;


public class ViewQr extends AppCompatActivity {
    public final static int QRcodeWidth = 500 ;
    Bitmap bitmap_qr;
    ImageView img_qr;
    Timer timer=new Timer();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_qr);

        img_qr=findViewById(R.id.imageViewQr);
        Button dowload=findViewById(R.id.buttonDownloadQR);



        try{

            String data=GlobalVariables.EMAIL+","+GlobalVariables.REG_NO;
```

```
            String encrypted_data=encrypt(data, "noufal");

            bitmap_qr= TextToImageEncode(encrypted_data);



            img_qr.setImageBitmap(bitmap_qr);


        } catch (Exception e) {
            e.printStackTrace();
        }

        dowload.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    Uri tempUri = getImageUri(getApplicationContext(),
bitmap_qr);
                    String picturePath= new
File(getRealPathFromURI(tempUri)).getPath();
                    Log.i("nnn","path"+picturePath);
                    Toast.makeText(getApplicationContext(), "Success!
\n"+picturePath, Toast.LENGTH_SHORT).show();
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });


    }


    public Uri getImageUri(Context inContext, Bitmap inImage) {
        String path =
MediaStore.Images.Media.insertImage(inContext.getContentResolver(), inImage,
GlobalVariables.REG_NO+"QR", null);
        return Uri.parse(path);
    }
    public String getRealPathFromURI(Uri uri) {
        String path = "";
        if (getContentResolver() != null) {
            Cursor cursor = getContentResolver().query(uri, null, null, null,
null);
            if (cursor != null) {
                cursor.moveToFirst();
                int idx =
cursor.getColumnIndex(MediaStore.Images.ImageColumns.DATA);
                path = cursor.getString(idx);
                cursor.close();
            }
        }
        return path;
    }


    Bitmap TextToImageEncode(String Value) throws WriterException {
        BitMatrix bitMatrix;
        try {
```

```
            bitMatrix = new MultiFormatWriter().encode(
                    Value,
                    BarcodeFormat.DATA_MATRIX.QR_CODE,
                    QRcodeWidth, QRcodeWidth, null
            );

        } catch (IllegalArgumentException Illegalargumentexception) {

            return null;
        }


        int bitMatrixWidth = bitMatrix.getWidth();

        int bitMatrixHeight = bitMatrix.getHeight();

        int[] pixels = new int[bitMatrixWidth * bitMatrixHeight];

        for (int y = 0; y < bitMatrixHeight; y++) {
            int offset = y * bitMatrixWidth;

            for (int x = 0; x < bitMatrixWidth; x++) {

                pixels[offset + x] = bitMatrix.get(x, y) ?
                        getResources().getColor(R.color.Black) :
getResources().getColor(R.color.White);
            }
        }
        Bitmap bitmap = Bitmap.createBitmap(bitMatrixWidth, bitMatrixHeight,
Bitmap.Config.ARGB_4444);

        bitmap.setPixels(pixels, 0, 500, 0, 0, bitMatrixWidth,
bitMatrixHeight);
        return bitmap;
    }




    private String encrypt(String user, String pass) throws Exception
    {
        SecretKeySpec k=gen(pass);
        Cipher cipher=Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE,k);
        byte[] en=cipher.doFinal(user.getBytes());
        String vall= Base64.encodeToString(en,Base64.DEFAULT);
        return vall;

    }

    private SecretKeySpec gen(String pass) throws Exception {
        final MessageDigest di= MessageDigest.getInstance("SHA-256");
        byte[] bytes=pass.getBytes("UTF-8");
        di.update(bytes,0,bytes.length);
        byte[] key=di.digest();
        SecretKeySpec secretKeySpec=new SecretKeySpec(key,"SHA");
        return secretKeySpec;

    }


}
```

VIEW VEHICLE

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.squareup.picasso.Picasso;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class ViewVehicle extends AppCompatActivity {

    TextView class_of_vehicle,reg_no_view,owner_name_view,chasis_no_view;
    ImageView img_rc,img_front,img_back,current_img,img_driver,img_licence;
Sms sms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_vehicle);


        reg_no_view=findViewById(R.id.textViewVehicleReg);
        owner_name_view=findViewById(R.id.textViewVehicleOwner);
        chasis_no_view=findViewById(R.id.textViewVehicleChasis);
        class_of_vehicle=findViewById(R.id.textViewVehicleClass);



        img_front=findViewById(R.id.imageViewFront);
        img_back=findViewById(R.id.imageViewBack);


        Button btn_delete=findViewById(R.id.button_delete);
        Button btn_view_qr=findViewById(R.id.buttonViewQR);




        getData();


        btn_view_qr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(ViewVehicle.this, ViewQr.class);
```

```
                startActivity(intent);
            }
        });
        btn_delete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {


                delete_vehicle();

            }
        });



    }




    private void getData()
    {

        String url =
GlobalVariables.URL_GET_VEHICLE_DETAILS+GlobalVariables.REG_NO;



        Log.i("nnn", "url="+url);
        StringRequest stringRequest = new StringRequest(url, new
Response.Listener<String>() {

            @Override
            public void onResponse(String response) {
                parseJsonData(response);

            }
        },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(ViewVehicle.this, error.getMessage(),
Toast.LENGTH_LONG).show();
                    }
                });

        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }

    public void parseJsonData(String response) {
        Log.i("nnnn", "Response"+response);
        try {
            // Create the root JSONObject from the JSON string.

            JSONObject jsonRootObject = new JSONObject(response);

            //Get the instance of JSONArray that contains JSONObjects
            JSONArray jsonArray = jsonRootObject.optJSONArray("Result");


            for (int i = 0; i < jsonArray.length(); i++)
```

```
            {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                String reg_no = jsonObject.optString("reg_no");
                String owner_name = jsonObject.optString("owner_name");
                String chasis_no = jsonObject.optString("chasis_no");
                String v_class = jsonObject.optString("type");
                String front_photo = jsonObject.optString("front_photo");
                String back_photo = jsonObject.optString("back_photo");


                reg_no_view.setText(reg_no);
                owner_name_view.setText(owner_name);
                chasis_no_view.setText(chasis_no);
                class_of_vehicle.setText(v_class);

                String imageUri;
                imageUri = GlobalVariables.host+front_photo;
                Picasso.with(ViewVehicle.this).load(imageUri).into(img_front);
                imageUri = GlobalVariables.host+back_photo;
                Picasso.with(ViewVehicle.this).load(imageUri).into(img_back);



            }

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }



    private void delete_vehicle()
    {

        String url = GlobalVariables.URL_DELETE_VEHICLE+GlobalVariables.REG_NO;

        Log.i("nnn", "url="+url);
        StringRequest stringRequest = new StringRequest(url, new
Response.Listener<String>() {

            @Override
            public void onResponse(String response) {
                Log.i("nnn","Response"+response);
                sms = new Sms(ViewVehicle.this);
                sms.send(GlobalVariables.PHONE, "Dear " + GlobalVariables.user
+ ", you have deleted the QR Code Tag for your vehicle"+GlobalVariables.REG_NO+
"from our system. Please note that the tag will become ineffective with in 2
minutes. \n" +
                        "If you havent done this , please visit your RTO
immediately or register a complaint to the nearest police station.");

                Toast.makeText(ViewVehicle.this,response,
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(ViewVehicle.this, Vehicle.class);
                startActivity(intent);
                finish();

            }
        },
                new Response.ErrorListener() {
```

```
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(ViewVehicle.this, error.getMessage(),
Toast.LENGTH_LONG).show();
                    }
                });

        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }
}
```

VIEW VEHICLES

```
package com.mtlz.qetc;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class ViewVehicles extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_vehicles);
    }
}
```

MAIN ACTIVITY SCANNER

```
package com.example.fasttagscanner;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.budiyev.android.codescanner.CodeScanner;
import com.budiyev.android.codescanner.CodeScannerView;
import com.budiyev.android.codescanner.DecodeCallback;
import com.google.zxing.Result;

import org.json.JSONException;
import org.json.JSONObject;

import java.security.MessageDigest;
```

```
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;


public class MainActivity extends AppCompatActivity {
    private CodeScanner mCodeScanner;
    CodeScannerView scannerView;
    private ProgressDialog progress;
    int flag=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        scannerView = findViewById(R.id.scanner_view);
        final Button scan=findViewById(R.id.scan_btn);

        scan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
               //
                if(flag==0) {
                    scan_code();
                    flag=1;
                }
                mCodeScanner.startPreview();
            }
        });

    }

    @Override
    protected void onResume() {
        super.onResume();
       // mCodeScanner.startPreview();
    }

    @Override
    protected void onPause() {
      //  mCodeScanner.releaseResources();
        super.onPause();
    }


    public void scan_code()
    {
        mCodeScanner = new CodeScanner(MainActivity.this, scannerView);
        mCodeScanner.setDecodeCallback(new DecodeCallback() {
            @Override
            public void onDecoded(@NonNull final Result result) {
                MainActivity.this.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {

                        String raw_data= result.getText();

                        String decrypted_data= null;
                        try {
                            decrypted_data = decrypt(raw_data, "noufal");
```

```
                            String[] values = decrypted_data.split(",");

                            System.out.println(Arrays.toString(values));
                            Log.i("nnn","email"+values[0]);
                            Log.i("nnn","reg"+values[1]);
                            GlobalVariables.EMAIL=values[0];
                            GlobalVariables.REG_NO=values[1];



                            get_data();



                        } catch (Exception e) {
                            Log.i("nnn","Error while decoding!");
                            e.printStackTrace();

                        }


                    }
                });
            }
        });
    }

    public void get_data()
    {

        progress=new ProgressDialog(this);
        progress.setMessage("Please Wait..");
        progress.show();

        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_GET_BALANCE,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        progress.hide();
//If we are getting success from server
                        Log.i("nnn",response);
                        String balance="",_fee="";
                        try {
                            JSONObject json_obj= new JSONObject(response);
                            balance=json_obj.getString("amount");
                            _fee=json_obj.getString("fee");
                            Log.i("nnn","STS="+balance);

                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                        if(response.contains("amount"))
                        {
                            Log.i("nnn","Success");


if(Double.parseDouble(balance)>=Double.parseDouble(_fee))
                            {

GlobalVariables.BALANCE=Double.parseDouble(balance);
                                GlobalVariables.FEE=Double.parseDouble(_fee);
```

```
                                update_balance();
                            }
                            else
                            {
                                Toast.makeText(MainActivity.this,"Insufficient
balance",Toast.LENGTH_SHORT).show();
                            }


                        }

                        else

                        {
                            Log.i("nnn","Error");

Toast.makeText(MainActivity.this,"Error",Toast.LENGTH_SHORT).show();
                        }

                    }
                },
                new Response.ErrorListener() {

                    @Override
                    public void onErrorResponse(VolleyError error) {
//You can handle error here if you want
                        progress.hide();
                    }

                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
//Adding parameters t o request

                params.put("email", GlobalVariables.EMAIL);
                params.put("reg_no", GlobalVariables.REG_NO);
//returning parameter
                return params;
            }
        };
//Adding the string request to the queue
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }



    public void update_balance() {
        // Toast.makeText(this, "Payment successfully done! " +
razorpayPaymentID, Toast.LENGTH_SHORT).show();
        StringRequest stringRequest = new
StringRequest(Request.Method.POST,GlobalVariables.URL_UPDATE_AMOUNT,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        Log.i("nnn","Response="+response);
//If we are getting success from server
                        if(response.contains("success"))
                        {
```

```
                            Toast.makeText(MainActivity.this,"Thank You!",
Toast.LENGTH_SHORT).show();


                        }
                        else
                        {
                            Toast.makeText(MainActivity.this, "Update Failed",
Toast.LENGTH_SHORT).show();
                        }
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
//You can handle error here if you want
                    }

                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String,String> params = new HashMap<>();
//Adding parameters t o request
                Double amount=GlobalVariables.BALANCE-GlobalVariables.FEE;
                params.put("email",GlobalVariables.EMAIL);
                params.put("amount",amount.toString());
                params.put("reg_no",GlobalVariables.REG_NO);
                params.put("fee",GlobalVariables.FEE.toString());

//returning parameter

                return params;
            }
        };

//Adding the string request to the queue
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);

    }
    private String decrypt(String out, String microtechlabz) throws Exception {
        SecretKeySpec k=gen(microtechlabz);
        Cipher cipher=Cipher.getInstance("AES");
        cipher.init(Cipher.DECRYPT_MODE,k);
        byte[] dv= Base64.decode(out,Base64.DEFAULT);
        byte[] deco= cipher.doFinal(dv);
        String n=new String(deco);
        return n;

    }




    private SecretKeySpec gen(String pass) throws Exception {
        final MessageDigest di= MessageDigest.getInstance("SHA-256");
        byte[] bytes=pass.getBytes("UTF-8");
        di.update(bytes,0,bytes.length);
        byte[] key=di.digest();
        SecretKeySpec secretKeySpec=new SecretKeySpec(key,"SHA");
        return secretKeySpec;
```

```
    }

}
```

GLOBAL VARIABLES SCANNER

```
package com.example.fasttagscanner;

public class GlobalVariables {

  //  static String host="http://192.168.0.8/";
    static String host="http://qetc.mcompany.in/";


    static String URL_GET_BALANCE=host+"select_amount_type.php";
    static String URL_UPDATE_AMOUNT=host+"update_amount_summary.php";

    static String EMAIL="";
    static String REG_NO="";
    static Double BALANCE=0.0;
    static Double FEE=0.0;

}
```

Ⅲ. PUBLICATION

QR Code Enabled Toll Collection With Encryption And Payment Gateway Integration

Denny P Francis[1], Joys Johney[2], Leo Lonappan[3], M.D.Damodaran[4], Midhun Benny[5]
[1]Associate Professor, Dept of Computer Science
[2, 3, 4, 5]Dept of Computer Science
[1, 2, 3, 4, 5] DePaul Institute of Science and Technology, Angamaly

Abstract- *Today, all most all toll plazas are operated manually. We often encounter traffic jams at almost all toll plazas due to this toll collection and the in-efficient training provided to the toll plaza workers. The time we spend at toll traffic is a waste of time. To avoid such delays we are developing a system using the latest technology QR code. This system will reduce the burden of collecting tolls and also reduces the man power required. In this system, scanners are used to capture the QR code and tag each vehicle as it passes through the toll gate. The QR code will be decoded by the central database and if validates the toll amount for the particular vehicle is deducted from the registered account of the user. The QR code is encrypted using algorithm. In the present day RFID (Radio Frequency Identification) has been used for this purpose. In order to overcome the issues of RFID tags QR code is proposed.*

Keywords- Toll Plaza, QR code, Encryption, RFID.

I. INTRODUCTION

A toll road which fee is assessed for passages have been implemented to recoup the money spends for construction and maintenance of the road. Transportation has become a human life daily routine hence there is a probability that everyone would encounter a toll plaza at one point of their life. Manual toll collection system became out-dated due to the manpower required and number of drawbacks.

Indian government has recently pushed for automatic toll collection using RFID technology this has several drawbacks. RFID uses radio waves to communicate with the receiver. An IR receiver is used to receive the pulse and a sends it to the controller, which transmits the vehicle through the RF transmitter located in the vehicle.

Some of the existing system uses RFID tags while others use the GSM module and INFRARED technology. We have designed the system keeping in mind to reduce the cost and also to find a solution to the existing traffic congestions at the toll booths. We suggested the use of QR technology for the *payment of toll gate processing*. Recent trends in the market have shown an increase in the use of QR codes than can be scanned and read by camera in a smartphone. The QR code allows storing data, numbers and even using it for payment and transfer of money.

Android based application is used for the usage of generating the unique QR code which is encrypted using the algorithm. The application allows for the user to register and login using personal information and generate QR code by providing the vehicle details hence a unique QR code is generated for that vehicle, any number of vehicle can be added which belongs to that person. The application *integrates a payment gateway* which is approved by the government hence enables an ease of transaction. We have included many additional features like *emergency SOS* call button, *SMS system* which regularly updates the customer On each toll passage along with the balance amount. *24/7 support and road side assistance* can be initiated within the application. With the payment gateway integration the user can recharge the wallet within the application hence enhancing the digital India scheme.

1.1 QR CODE

QR Code is a matrix barcode that can be used to store data. It consists of black modules arranged in a square grid which is then read by the camera. Data is extracted from patterns in both directions of the image. QR code offers numerous benefits such as [6]

- Cost effective
- Readable from distance
- Structured appending


Fig 1.1 QR Code

On a grid of 6x6 mask pattern is defined which is necessary to cover the whole symbol. Patterns in data area such as blank areas or misleading features that look like locator marks can confuse scanner, hence masking is used. QR code has become a focus of advertising strategy, since it provides a way to access a brand's website more quickly than by manually entering a URL.

1.2 ANDRIOD

QR Code based toll collection uses different system from the hardware components to the user component in this paper we assume that all the pre-defined hardware for the real world systems are implemented as we already have existing toll plaza only minor upgrades have to be done in order to fit the scanners. This paper talks about the android application. Android is a mobile operating system by Google. It is based on the Linux kernel and other open source software. Android is basically for the touch and type interface that can be implemented in a smartphone and other devices. Google has further developed android TV, Wear OS, and Android Auto that is used in cars. Android is designed to keep the processes consume the battery charge at a minimum rate. Android has enabled the market to be flooded with smartphones and hence opened up a whole new digital world to the different sections of the society. This revolution in smartphones has enabled more people to go online and use different technologies that are available to them hence it important to keep updating the technologies that can have a greater importance in people's lives.

1.3 PAYMENT GATEWAY

Payment gateway is a merchant service provided by an e-commerce application service provider that authorizes credit card or direct payment processing gateway may be provided by a bank to its customers, but can be provided by a specialised financial service provider as a separate service, such as a payment service provider.

A payment gateway like Razor pay that we are integrating with the application allows the user to securely transact their payment an HTTPS protocol based transaction take place. Virtual payer authentication is something that the acquirers, issuers and the payment gateways are backing to secure the process even more. PCI-DSS makes it secure enough to allow the user to store their personal data in the portal or gateway for recurring payments.

The most significant advantage of a payment gateway is the fact that it allows millions of users to use it at the same time, making it possible for you to purchase or sell goods and services whenever you want.

1.4 SMS

SMS (Short Message Service) is a text messaging service component of most telephone, internet and mobile device systems. The protocols used in this service allow users to send and receive messages to and from GSM mobiles. SMS although commonly seen in the mobile to mobile services it can be expanded into technologies that offer services and support.

SMS is a stateless communication protocol in which every SMS message is considered entirely independent of other messages. In this system SMS is used to send alert, balance and other toll based reports directly to the user mobile. This enables the user to have a regular update on the main account balance of the wallet.

II. LITERATURE REVIEW

Automated toll collection and check post system using Radio Frequency Identification (RFID) and Global System for Mobile Communication (GSM) module. The recognition is succeeded with the guidance of passive radio frequency. In this paper [1], vehicle particulars like unique ID is saved in RFID tag which is attached in the vehicle. Image process and GPS is combined with RFID and GSM module to create the system a lot reliable and secure. The hardware design of the paper consists of transmitter and receiver module. Transmitter module is fixed in the vehicle as an active tag. Receiver is the automated check post and e-toll control. ATmega328 Arduino controller is a 28 pin arduino controller which has 32 bit natural working registers. The servo Motor SG90 type is used in this setup. The servo Motor is used for automatic gate operation whenever the motor receives the signal from the controller. SIM800 GSM is a communication device designed for global market. RFID uses radio wave to process the information from the devices. The bi-directional connection network with endpoints has been designed to use

RFID to produce an electronic product code. An experimental study result has also been given at the end of the paper.

In paper [2], the toll collection system is designed primary for the use by GSM and GPS systems. The GPS is used to find the position of the vehicle. The GPRS kit should be installed in the vehicle in track the vehicle. Each GPRS system has a subscriber identity module. This system also incorporates geo fences of the toll plaza to get the information regarding the location of the toll plaza. The GPS and GPRS are integrated into an ARM microcontroller. In order for the GPRS system to work the system requires 1 GB of data and a stable network connection which should be switched on all the time. The system also requires a mobile tower to be located at the toll plaza to process the information of the vehicle in the rage of the tower. The system compares the position of the vehicle and when the vehicle is within the 5 meter radius of the toll plaza the database is updated and the amount is debited from the user's account which will be immediately followed by an SMS to the customer. The position of the vehicle or the GPRS SIM module is compared using the haversine formula, and if the vehicle is within the range of toll plaza, amount will be taken from the account.

3D environment modelling for the toll collection which directly affects the decision making part, in paper [3] the new toll gate approach is divided into perception, decision making and motion control. A deep natural network is used in perception. Virtual lanes are generated from the 3D environment result and an optimal lane is selected. In motion control a collision free path is planned and transmitted to maneuver the vehicle. The automation driving vehicle receives the localization and mapping (SLAM) algorithm estimates the motion state of the vehicle. The traffic patterns are further understood and scanned to detect passable electronic toll collection gate. These virtual lines are generated only for the types of vehicle were this system can be implemented. Convolutional neural network (CNN) based algorithm is used for object detection to sense other vehicle, and ETC gates. YOLO algorithm a type of CNN with structure of GoogleNet, is divided into 24 convolutional layers with connected layers of 2. The determination of total queue length is calculated by first detect the neighbouring vehicle and correlate the distance of vehicle to the gate. The SLAM used to update or construct maps on an unknown environment. LSD-SLAM (Large Scale Direct Monocular) is applied to track and map by image intensities divided into three steps: tracking, depth map estimation and map optimization. The system takes a while to scan all the gates as the gates could be occluded by trucks or buses. The coordinates are interpreted and the best cubic coordinate origin and the ETC gate are selected. The motion control adopts an adaptive method. It receives a collision free trajectory data from path planning using curvature, yaw rate and velocity. In this approach LQR-PID algorithm is used. LQR can exactly pave a way to the optimum pole. LQR predicts an expectation as inputs to PID controller. PID is a classic control with strong adaptation and robustness. YOLO algorithm can detect most of the vehicles. ETC signs can also be detected using this algorithm. The proposed system is designed to be more universally usable, even without HD map and V2X. The V2X will make visual classification of ETC gate obsolete.

In paper [4], a proposed system is a web application and android application. The system is designed primarily for devices like smartphones, personal computers and all other devices which support web services. The main objective behind this paper is to design the application which provides an effective and easier way to payment of road toll. Keeping in mind the Indian condition the application contains QR code for recognition with centralized availability of data. Throughout the system it enables user to pay from the account created after reaching the toll booth. The system gives many advantages and assures and accurate collection of toll amount. This paper uses GUI for collection of toll, the real time monitoring and management is done. The architecture of the system uses user application to generate QR code which can be connected by GPS for the connection of toll plaza receiver. The data base of the system saves the details of the vehicle connected and other different toll collection. The server coordinates all the different activates of the application.

RFID is a dependable technology in paper [5], the RFID automatic toll gate system can automatically discover the vehicles of the identities, reading items in motion and tracing of the vehicles can be done by accurately by RFID. In this paper they have executed a framework which will punish for infringement of toll entryway and they believe it will prompt to a fastidious activity. The framework developed will help in reducing the number of mischance. The framework integrates the RFID, AVR microcontroller, the database creation and GUI outline. RFID tags are fixed and attached and through this the reader reads the data. The main aim of this paper is addressing the prevention of motorists and toll authorities manually perform ticket payments and also check driving without documents. This system proposes to identify theft vehicles. When the gate is automated it requires minimum human intervention hence efficiency can be improved. The framework expands wellbeing. The paper also looks at the adequacy of toll stations and the road developments that limited to the toll road. RFID technologies implemented in the system adopta kind of frequency chips which authenticates and authorizes protocol model used to guarantee system security. Accordingly, electronic toll

collection system deserves deeply research whether from technology, economy or environment protection.

III. CONCLUSION

In this paper, we proposed the approach of automatic toll collection using QR code that was seen as new age technology and allows us to greatly reduce the manpower and reduces traffic congestion. QR code allows for the toll collection system to be much better than the RFID tags used in today's vehicle. QR code gives authorities to set variable pricing for toll services and allows for a fair tax collection. Due to the technology, the QR code is generated after recharge and allows to get the QR code can be shown if the scanner is unable to scan the code. Due to the encryption algorithm used we try to integrate a payment gateway into the android application. This technology has high securable and low amount of speed cost. The messaging system build into the application allows for a regular update within the usage of toll. The implementation of this technology takes very little money and allows for updated driving dynamics as different toll gates will have different scanners that will tag each and every vehicle which very efficient as there are multiple scanners placed at regular points that tags the vehicle hence the scanning can never be avoided. The android application supports the QR code in building a strong network of commuters that help to keep the Indian roads less congested by the traffic.

REFERENCES

[1] KBalamurugan,Dr.S.Elangovan,Dr.R.Mahalakshmi,R.Pavithra**"**Automatic Check-Post and Fast Track Toll System Using RFID and GSM Module with Security System**"** *International conference on Advances in Electrical Technology for Green Energy (ICAETGT)*2017 978-1-5386-2138-7/17/$31.00 ©2017 IEEE

[2] Sudheer Kumar Nagothu **"**Automated Toll Collection System Using GPS and GPRS**"***International Conference on Communication and Signal Processing* 978-1-5090-0396-9/16/$31.00 ©2016 IEEE

[3] ZhonglinXu, Xinhui Di, Houming Wang, Jun Xu, Rolf Adomat, Chen Zhang **"** Automated Toll Gate Passing**"***Intelligent Vehicles Symposium(IV),Changshu* 978-1-5386-4452-2/18/$31.00 ©2018 IEEE

[4] Machindra Vetal, Pravin Gore, Nikhil Pathil, Kiran Sapate, Manisha Singh **"** Android Based Smart Toll System Using QR Code**"***International Journal of Innovative Research in Computer and Communication Engineering*, Vol.5,Issue 9,2017 ISSN:2320-9801

[5] K.Gowrisubadra, Jeevitha.S, Selvarasi.N **"**A Survey on RFID Based Automatic Toll Management**"***International Conference on Signal Processing, Communications and Networking(ICSCN)*2017,978-1-5090-4740-6/17/$31.00 ©2017 IEEE

[6] Rahul Raj C. P, SeshuBabu Tolety, Catheine Immaculate, **"**QR code based navigation system for closed building using smart phones,**"** 978-1-4673-5090-7/13/$31.00 ©2013 IEEE.

[7] Kasturi Shah, Prajakta Joshi, Dishaa Garg **"**Automatic Toll Collection Using QR Code**"***International Research Journal of Engineering and Technology(IRJET)*,Vol.3,Issue 10,2016 ISSN:2395-0056-reference

[8] R.Latha,C.Sivaraj,V.Sivaranjani,S.Vimaleeshwari **"**E-Payment for Toll-Gate Processing using QR Technology**"***Journal of Chemical and Pharmaceutical Sciences(JCHPS)* Issue:3,2017

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

Study of QR Code Enabled Toll Collection and Payment Gateway Integration

Denny P Francis[1], Joys Johney[2], Leo Lonappan[3], M. D. Damodaran[4], Midhun Benny[5]
[1]*Asst. Professor, Department of Computer Science, DIST,* [2, 3, 4, 5]*Dual Degree MCA, DePaul Institute of Science and Technology, Angamaly*

Abstract: ***Android based application is used for the usage of generating the unique QR code which is encrypted using the algorithm. The application allows for the user to register and login using personal information and generate QR code by providing the vehicle details hence a unique QR code is generated for that vehicle, any number of vehicle can be added which belongs to that person. The application integrates a payment gateway which is approved by the government hence enables an ease of transaction. We have included many additional features like emergency SOS call button, SMS system which regularly updates the customer On each toll passage along with the balance amount. 24/7 support and road side assistance can be initiated within the application. With the payment gateway integration the user can recharge the wallet within the application hence enhancing the digital India scheme.Today, all most all toll plazas are operated manually. We often encounter traffic jams at almost all toll plazas due to this toll collection and the in-efficient training provided to the toll plaza workers. The QR code is encrypted using algorithm. In the present day RFID (Radio Frequency Identification) has been used for this purpose. In order to overcome the issues of RFID tags QR code is proposed.***

Index Terms: Toll Plaza, QR code, Encryption, RFID.

I. INTRODUCTION

A toll road which fee is assessed for passages have been implemented to recoup the money spends for construction and maintenance of the road. Transportation has become a human life daily routine hence there is a probability that everyone would encounter a toll plaza at one point of their life. Manual toll collection system became out-dated due to the manpower required and number of drawbacks. Indian government has recently pushed for automatic toll collection using RFID technology this has several drawbacks. RFID uses radio waves to communicate with the receiver. An IR receiver is used to receive the pulse and a sends it to the controller, which transmits the vehicle through the RF transmitter located in the vehicle. Some of the existing system uses RFID tags while others use the GSM module and INFRARED technology. We have designed the system keeping in mind to reduce the cost and also to find a solution to the existing traffic congestions at the toll booths. We suggested the use of QR technology for the *payment of toll gate processing*. Recent trends in the market have shown an increase in the use of QR codes than can be scanned and read by camera in a smartphone. The QR code allows storing data, numbers and even using it for payment and transfer of money. Android based application is used for the usage of generating the unique QR code which is encrypted using the algorithm. The application allows for the user to register and login using personal information and generate QR code by providing the vehicle details hence a unique QR code is generated for that vehicle, any number of vehicle can be added which belongs to that person. The application integrates a payment gateway which is approved by the government hence enables an ease of transaction. We have included many additional features like emergency SOS call button, SMS system which regularly updates the customer On each toll passage along with the balance amount. 24/7 support and road side assistance can be initiated within the application. With the payment gateway integration the user can recharge the wallet within the application hence enhancing the digital India scheme.

II. LITERATURE REVIEW

Automated toll collection and check post system using Radio Frequency Identification (RFID) and Global System for Mobile Communication (GSM) module. The recognition is succeeded with the guidance of passive radio frequency. In this paper [1], vehicle particulars like unique ID is saved in RFID tag which is attached in the vehicle. Image process and GPS is combined with RFID and GSM module to create the system a lot reliable and secure. The hardware design of the paper consists of transmitter and receiver module. Transmitter module is fixed in the vehicle as an active tag. Receiver is the automated check post and e-toll control. ATmega328 Arduino controller is a 28 pin Arduino controller which has 32 bit natural working registers. The servo Motor SG90 type is used in this setup. The servo Motor is used for automatic gate operation whenever the motor receives the signal from the controller. SIM800 GSM is a communication device designed for global market. RFID uses radio wave to process the information from the devices. The bi-directional connection network with endpoints has been designed to use RFID to produce an electronic product code. An experimental study result has also been given at the end of the paper.

In paper [2], the toll collection system is designed primary for the use by GSM and GPS systems. The GPS is used to find the position of the vehicle. The GPRS kit should be installed in the vehicle in track the vehicle. Each GPRS system has a subscriber identity module. This system also incorporates geo fences of the toll plaza to get the information regarding the location of the toll plaza. The GPS and GPRS are integrated into an ARM microcontroller. In order for the GPRS system to work the system requires 1 GB of data and a stable network connection which should be switched on all the time. The system also requires a mobile tower to be located at the toll plaza to process the information of the vehicle in the rage of the tower. The system compares the position of the vehicle and when the vehicle is within the 5 meter radius of the toll plaza the database is updated and the amount is debited from the user's account which will be immediately followed by an SMS to the customer. The position of the vehicle or the GPRS SIM module is compared using the haversine formula, and if the vehicle is within the range of toll plaza, amount will be taken from the account.

3D environment modelling for the toll collection which directly affects the decision making part, in paper [3] the new toll gate approach is divided into perception, decision making and motion control. A deep natural network is used in perception. Virtual lanes are generated from the 3D environment result and an optional is selected. In motion control a collision free path is planned and transmitted to maneuver the vehicle. The automation driving vehicle receives the localization and mapping (SLAM) algorithm estimates the motion state of the vehicle. The traffic patterns are further understood and scanned to detect passable electronic toll collection gate. These virtual lines are generated only for the types of vehicle were this system can be implemented. Convolutional neural network (CNN) based algorithm is used for object detection to sense other vehicle, and ETC gates. YOLO algorithm a type of CNN with structure of Google Net, is divided into 24 convolutional layers with connected layers of 2. The determination of total queue length is calculated by first detect the neighbouring vehicle and correlate the distance of vehicle to the gate. The SLAM used to update or construct maps on an unknown environment. LSD-SLAM (Large Scale Direct Monocular) is applied to track and map by image intensities divided into three steps: tracking, depth map estimation and map optimization. The system takes a while to scan all the gates as the gates could be occluded by trucks or buses. The coordinates are interpreted and the best cubic coordinate origin and the ETC gate are selected. The motion control adopts an adaptive method. It receives a collision free trajectory data from path planning using curvature, yaw rate and velocity. In this approach LQR-PID algorithm is used. LQR can exactly pave a way to the optimum pole. LQR predicts an expectation as inputs to PID controller. PID is a classic control with strong adaptation and robustness. YOLO algorithm can detect most of the vehicles. ETC signs can also be detected using this algorithm. The proposed system is designed to be more universally usable, even without HD map and V2X. The V2X will make visual classification of ETC gate obsolete.

In paper [4], a proposed system is a web application and android application. The system is designed primarily for devices like smartphones, personal computers and all other devices which support web services. The main objective behind this paper is to design the application which provides an effective and easier way to payment of road toll. Keeping in mind the Indian condition the application contains QR code for recognition with centralized availability of data. Throughout the system it enables user to pay from the account created after reaching the toll booth. The system gives many advantages and assures and accurate collection of toll amount. This paper uses GUI for collection of toll, the real time monitoring and management is done. The architecture of the system uses user application to generate QR code which can be connected by GPS for the connection of toll plaza receiver. The data base of the system saves the details of the vehicle connected and other different toll collection. The server coordinates all the different activates of the application.

RFID is a dependable technology in paper [5], the RFID automatic toll gate system can automatically discover the vehicles of the identities, reading items in motion and tracing of the vehicles can be done by accurately by RFID. In this paper they have executed a framework which will punish for infringement of toll entryway and they believe it will prompt to a fastidious activity. The framework developed will help in reducing the number of mischance. The framework integrates the RFID, AVR microcontroller, the database creation and GUI outline. RFID tags are fixed and attached and through this the reader reads the data. The main aim of this paper is addressing the prevention of motorists and toll authorities manually perform ticket payments and also check driving without documents. This system proposes to identify theft vehicles. When the gate is automated it requires minimum human intervention hence efficiency can be improved.

The framework expands wellbeing. The paper also looks at the adequacy of toll stations and the road developments that limited to the toll road. RFID technologies implemented in the system adopt a kind of frequency chips which authenticates and authorizes protocol model used to guarantee system security. Accordingly, electronic toll collection system deserves deeply research whether from technology, economy or environment protection.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

III. OBJECTIVES

The objective of QRCode Tag is to establish a system where the tag used for automatic toll collection is between two parties and an ease of use through a user focused application unlike the FASTTag existing application available as of now. The application will have added features to further support the user in his/her commute through the highways of India. Feedback from the user about the application, Report and issue about the highway. Emergency services that is available throughout and over the course of journey. A main account which will erase the third parties present in today's system. Generation of tag, adding and deletion of multiple vehicles.

IV. SOFTWARE REQUIREMENTS

1) For this project we have used the following software for the development of the application that is available to the user.

a) Andriod Studio

b) Google Chrome

c) Microsoft Word

d) Notepad

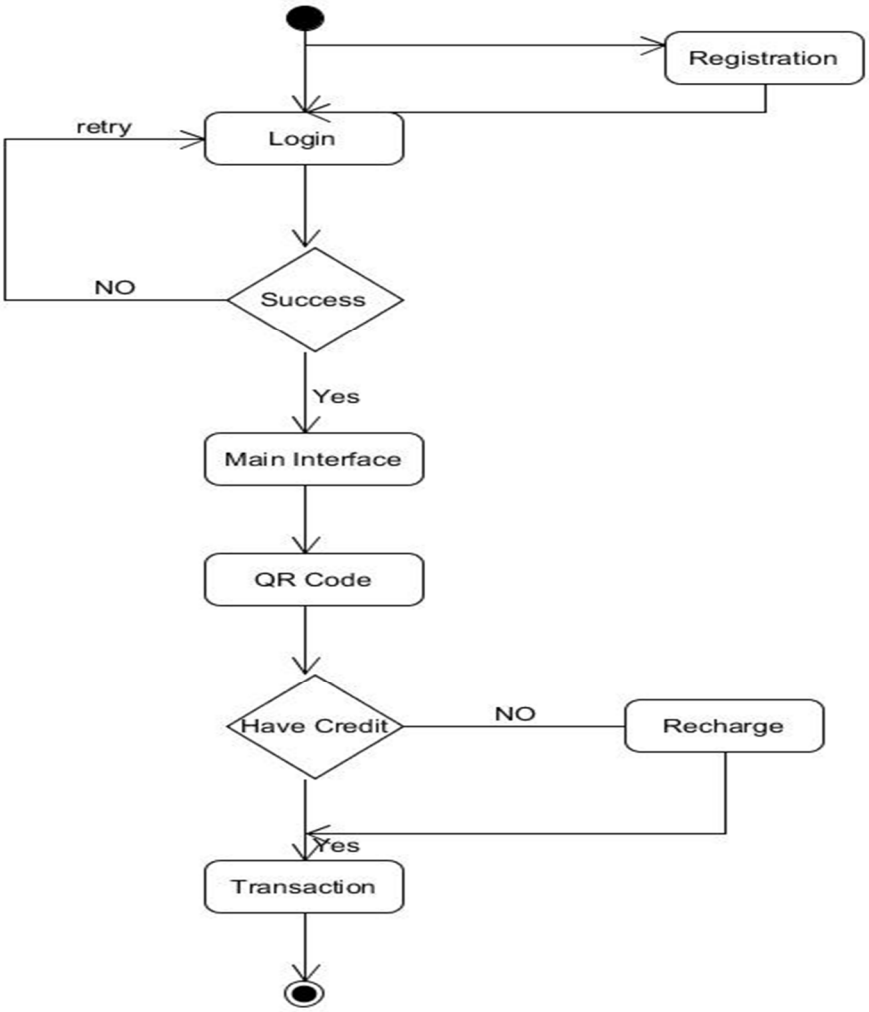
2) We have made use of the following languages for the development of this application.

a) PHP

b) Java

c) Andriod

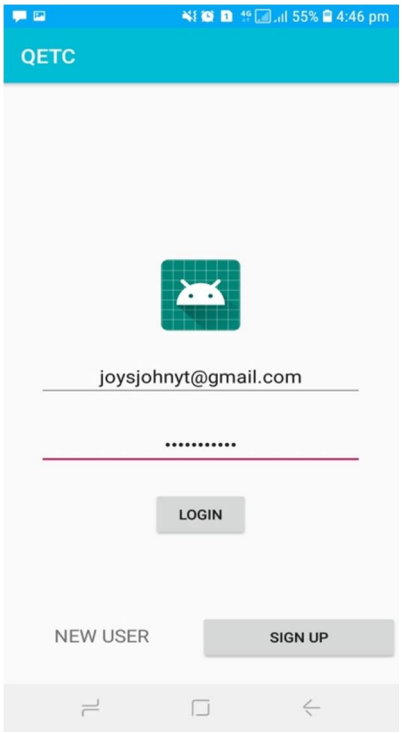
V. SYSTEM ARCHITECTURE



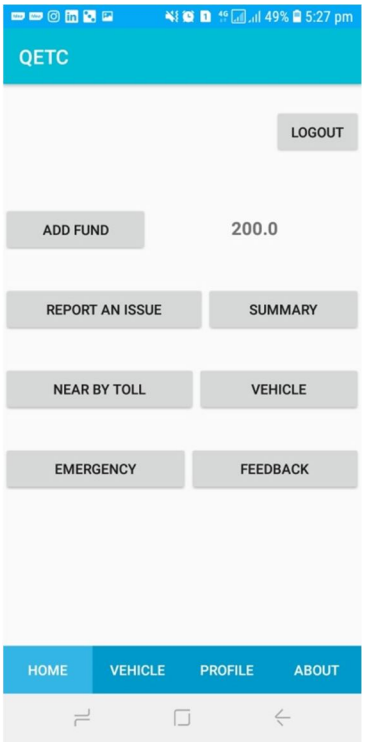
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

VI. RESULT ANALYSIS

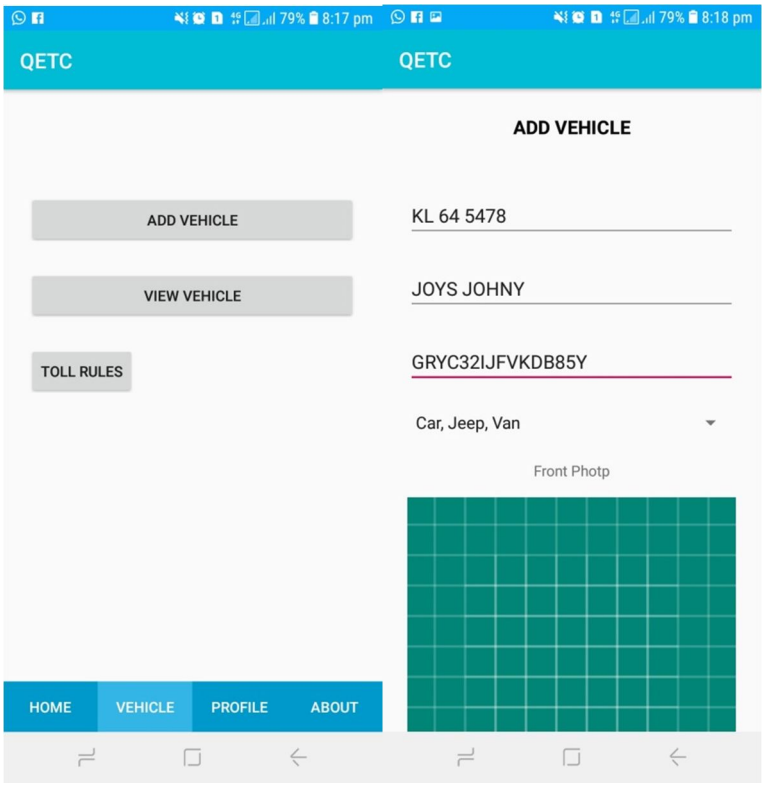
A. The login page to which the application opens into. The secure login page allows for the registered users to access the application**.**



B. The home page where the user will be directed to after login. The home page will take the user to different features with the application such as adding vehicle, generating QRCode, report an issue on the highway, emergency dials and many more.

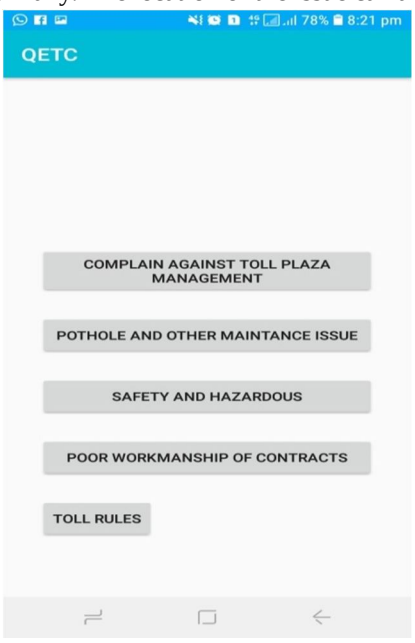


C. The vehicle tab will take the user to the important feature of the application where the user can add the vehicle, generate the QRCode Tag for that vehicle, view the number of registered vehicle and delete the vehicle where the QRCode Tag will also become invalid.

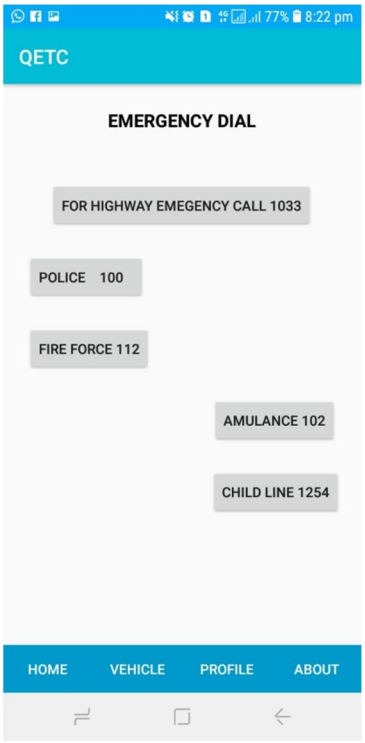


The add vehicle page will have the user to upload photos of the car as well as RC. The generation will happen based on the class of vehicle selected.

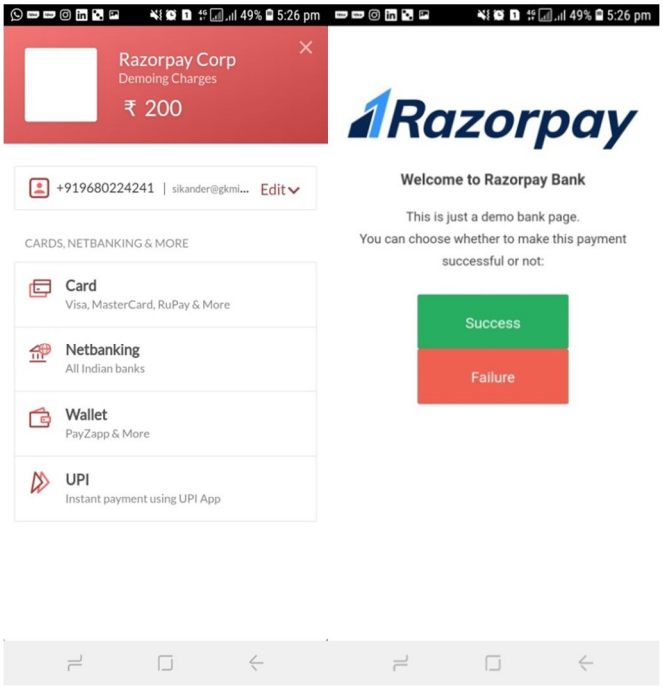
D. The report an issue page is accesible from the home page of the application. The report an issue has a consolidated four issues that are commonly found during the travel in the highways. The issue will take to a issue reporting page where the user can comment the issues and upload visual evidence if any. The location of the issue can also be inserted.



E. The emergency tab will allow the user to access the emergency dial which has a variety of emergency services all integrated into a single page window. This helps in user to dial and contact different services and also the selection of appropriate emergency services.



F. This application allows the user to recharge the tag from the application itself hence reducing the third party system and helps with the integration of bank accounts into the tag. This imrpves the trust in the tag system and the government as they are no longer required to memorize a 13 digit bank account number or obliged to any banks for that matter. The payment gateway is integrated into the system which will recharge the main account from which the toll amount will be deducted.



VII. CONCLUSION

In today's world it is important that the application for a particular system be user-friendly and should be designed in such a way the burden on the user should be greatly reduced. When a new technology like QRTag is been proposed instead of the existing technology, the proposed system should cover all the demerits of the existing system. The user application was one such system where the user had limited access due to the system design. We proposed a new system and a new user application for it. The user application added many features which help the user in his/her daily commute to the highways. The application generated the QR code for the application that can be activated from the RTO of the user. Many new features can be added to the application at a later stage or updated based on the feedback that the user submits.

REFERENCES

[1] Devayani1, Kowsika, Mr. G. Murugan, Mrs. V. Hemalatha, Dr. P. Gomathi "Student Information System", April-2020.
[2] KBalamurugan, Dr.S.Elangovan, Dr.R.Mahalakshmi,R. Pavithra"Automatic Check-Post and Fast Track Toll System Using RFID and GSM Module with Security System", International conference on Advances in Electrical Technology for Green Energy (ICAETGT)2017 978-1-5386-2138-7/17/$31.00 ©2017 IEEE.
[3] Rahul Raj C. P, SeshuBabu Tolety, Catheine Immaculate, "QR code based navigation system for closed building using smart phones," 978-1-4673-5090-7/13/$31.00 ©2013 IEEE.
[4] Kasturi Shah, Prajakta Joshi, Dishaa Garg "Automatic Toll Collection Using QR Code"International Research Journal of Engineering and Technology(IRJET),Vol.3,Issue 10,2016 ISSN:2395-0056.
[5] R.Latha,C.Sivaraj,V.Sivaranjani,S.Vimaleeshwari "E Payment for Toll-Gate Processing using QR Technology"Journal of Chemical and Pharmaceutical Sciences(JCHPS) Issue:3,2017.

CHAPTER Ⅷ
REFERENCES

[1]. Devayani1, Kowsika, Mr. G. Murugan, Mrs. V. Hemalatha, Dr. P. Gomathi "Student Information System", April-2020.

[2]. KBalamurugan, Dr.S.Elangovan, Dr.R.Mahalakshmi,R. Pavithra"Automatic Check-Post and Fast Track Toll System Using RFID and GSM Module with Security System", International conference on Advances in Electrical Technology for Green Energy (ICAETGT)2017 978-1-5386-2138-7/17/$31.00 ©2017 IEEE.

[3]. Rahul Raj C. P, SeshuBabu Tolety, Catheine Immaculate, "QR code based navigation system for closed building using smart phones," 978-1-4673-5090-7/13/$31.00 ©2013 IEEE.

[4]. Kasturi Shah, Prajakta Joshi, Dishaa Garg "Automatic Toll Collection Using QR Code"International Research Journal of Engineering and Technology(IRJET),Vol.3,Issue 10,2016 ISSN:2395-0056.

[5]. R.Latha,C.Sivaraj,V.Sivaranjani,S.Vimaleeshwari "E Payment for Toll-Gate Processing using QR Technology"Journal of Chemical and Pharmaceutical Sciences(JCHPS) Issue:3,2017.

[6]. ZhonglinXu, Xinhui Di, Houming Wang, Jun Xu, Rolf Adomat, Chen Zhang **"** Automated Toll GatePassing**"***Intelligent Vehicles Symposium(IV),Changshu* 978-1-5386-4452-2/18/$31.00 ©2018 IEEE

ISSN [ONLINE] : 2395-1052

THE BOARD OF

IJSART

INTERNATIONAL JOURNAL FOR SCIENCE AND ADVANCE RESEARCH IN TECHNOLOGY

is here by awarding this certificate to

JOYS JOHNEY

In recognition of publication of the paper entitled

QR CODE ENABLED TOLL COLLECTION WITH ENCRYPTION AND PAYMENT GATEWAY INTEGRATION

Published in E-Journal

Volume 6,Issue 1 in January 2020

EDITOR IN CHIEF

PAPER ID : IJSARTV6I135667

Email id : editor@ijsart.com | website : www.ijsart.com

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.129

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL IMPACT FACTOR : 7.177

Certificate

It is here by certified that the paper ID : IJRASET27770, entitled

Study of QR Code Enabled Toll Collection and Payment Gateway Integration

by

Joys Johney

after review is found suitable and has been published in Volume 8, Issue IV, April 2020

in

International Journal for Research in Applied Science & Engineering Technology

Good luck for your future endeavors

Editor in Chief, iJRASET