AUTOMATIC TOLL COLLECTION AND GATE OPERATION USING IMAGE PROCESSING

Capstone Final Report

Submitted To
School of Electronics and Electrical Engineering

By

MARAM SREERAMA REDDY	-	11610529
CHALLA NAGA TEJA	-	11603089

Under the Guidance of

Mr. Mohit Kumar Goel

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING,
LOVELY PROFESSIONAL UNIVETSITY,
PHAGWARA, PUNJAB (INDIA) – 144411**

TOPIC APPROVAL PERFORMA

TOPIC APPROVAL PERFORMA

School of Electronics and Electrical Engineering (SEEE)

Program :   P135-H::B.Tech. (Electronics & Communication Engineering) (Hons.)

COURSE CODE :   ECE442B   **REGULAR/BACKLOG :**   Regular   **GROUP NUMBER :**   EEERGC0471

Supervisor Name :   Mohit Kumar Goel   **UID :**   16907   **Designation :**   Assistant Professor

Qualification : _____   **Research Experience :** _____

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Maram Sreerama Reddy	11610529	2016	E1601	9989831059
2	Challa Naga Teja	11603089	2016	E1602	8847373493

SPECIALIZATION AREA :   Signal and Image Processing   **Supervisor Signature:** _____

PROPOSED TOPIC :   Automatic toll collection and toll gate operation using image processing

Details are not entered

PAC Committee Members		
PAC Member (HOD/Chairperson) Name: Dr. Bhaveshkumar C. Dharmani	UID: 23409	Recommended (Y/N): NA
PAC Member (Allied) Name: Navjot Rathour	UID: 16885	Recommended (Y/N): NA
PAC Member 3 Name: Anil Kumar Rawat	UID: 13623	Recommended (Y/N): NA

Final Topic Approved by PAC: Design of an Automatic toll collection and toll gate operation using image processing

Overall Remarks: Approved (with minor changes)

PAC CHAIRPERSON Name: 11211::Prof. Bhupinder Verma   **Approval Date:**   16 Nov 2019

DECLARATION

We hereby declare that the project work entitled "AUTOMATIC TOLL COLLECTION AND GATE OPERATION USING IMAGE PROCESSING" is an authentic record of our own work carried out as requirements of Capstone Project for the award of degree of B. Tech in ECE from Lovely Professional University, Phagwara, under the guidance of Assistant Professor, Mr. Mohit Kumar Goel, during August 2019 - June 2020. The contents of this Capstone project, in full or in parts, have neither been taken from any   other source nor have been submitted to any other Institute or University for award of any degree or diploma.

MARAM SREERAMA REDDY   -   11610529

CHALLA NAGA TEJA   -   11603089

ACKNOWLEDGEMENT

We take this opportunity to present my votes of thanks to all those guideposts who really acted as lightening pillars to enlighten my way throughout this project that has led to successful and satisfactory completion of this study.

We are really grateful to Mr. Mohit Kumar Goel for providing us with an opportunity to undertake this project and providing us with all the facilities. We are highly thankful to him for this active support, valuable time and advice, whole-hearted guidance, sincere co-operation and painstaking involvement during the study and in completing the assignment of preparing the said project plan within the time stipulated.

Lastly, we thankful to all those, particularly the various friends who have been integument in creating proper, healthy and conductive environment and including new and fresh innovative ideas for us during   the project, without their help, it would have been extremely difficult for us to prepare the project plan in a time bound framework.

ABSTRACT

The project deals with examining the images and the respective information of the vehicles for automating the process of toll collection and making it more efficient and perfect. On any toll booth the vehicle has to stop for paying the toll. But in this system, a camera is used for capturing the image of the vehicle's number plate. The captured image would be converted into text using Automatic number plate recognition system, and the toll would be cut from the customer's account in the application we have created and the gate would be opened.

Moreover in our system if a vehicle is stolen and an entry is being made in the central database by the police, then if the vehicle passes through the toll booth, then a silent alarm would buzz which would indicate the operator at the toll booth, that the vehicle is a stolen one.

For the identification of the vehicle the information of the vehicles is already stored on the central database, so the captured number would be sent to the server received at the toll, and the server would take care of the billing.

By using system we would also monitor the speed of the vehicles on the highways, by installing our system at different parts of the highway.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The purpose of this project is collection the toll according to the vehicles and build the real time application which recognizes vehicle's license number plate at the entry gate. Automatic toll collection is considered as one of the intelligent transport system. It is aimed at making toll collection more efficient, reliable, safe and environmental friendly. Presently, the customers have to wait at the toll gate to pay the collectors, creating a traffic congestion, pollution and of course a lot of frustration. This Automatic toll collection successfully removes the unnecessary traffic delays, keeps an eye on any car that might not be lawfully registered. Automated toll collection is fast becoming a global accepted way of toll collection.

Image processing toll booths system is used as a system for the fast and efficient collection of tolls at the toll plazas. This is possible as the vehicles passing at the toll plaza do not need to stop to pay the toll, and the payment automatically takes place from the account of the users on the web application we create. This system used the technology of Automatic number plate recognition system.

Automatic vehicle identification system is used for the purpose of effective control. License plate recognition is a form of automatic vehicle identification. It is an image processing technology used to identify vehicles by using only their license plates. Real time license plate recognition plays a major role in automatic monitoring of traffic rules and monitoring the law enforcement on public roads. Since every vehicle carries a unique license plate, no external cards, tags or transmitters are required.

In this Toll collection system, the Automatic vehicle classification system verifies the classification of the vehicles automatically as the vehicle is already classified at the time of registration. The Traffic controller system is a computer system which manages the traffic in a single row or inline why using traffic signals and sensors.

Central server is used for processing payments, for more security and maintaining the records of each customer and their toll transactions. It stores the data coming from different toll plazas. A local computer of every toll plaza is connected to the central server through the internet. The consumer/owner of the vehicle has to register at the central server prior any transaction and may deposit money in their account. Automatic vehicle identification and classification totally depends on the vehicles number plate.

1.1 Objectives:

Automation the process of toll collection system. This will help in making the process of toll charging system faster smoother and error free.
Reduction of traffic congestion at the toll plaza, this will help in reducing the traffic and save a lot of time for the passengers.

Reduction of man power needed in the present system helps in reducing the costs borne by the government in process of collecting toll charges. Elimination of human errors by installing a computerized system is also another advantage of this.

Help police/vigilance department in monitoring the vehicle movements. Help in detecting unlawful driving actions such as not wearing helmets, seat belts, over speeding of vehicles etc.

Digitalizing and securing payment system and record keeping.

1.2 Novelty:

Automation of toll collections. This becomes the main objective and aim of the project which ensures that the vehicles need not stop at every toll plaza, and the automated process will help reduce the vehicle traffic flow to a significant level.

Minimum to no intervention of a man working on the system, eliminates possible human errors. Human errors are inevitable when a person works with a computer. But in this process since all the work is done by the computer itself without any interference of a person, the errors can be completely eliminated.

Constant monitoring of vehicles is a by-product of our system, which may find many use cases for various proposes, including helping solving a crime, detecting stolen vehicles, recognize vehicles with a fake or swapped licence plate etc.

CHAPTER 2

2. TECHNOLOGIES USED:

2.1 High resolution cameras

Digital imaging devices, such as digital camera contain in-built image processing system for the applications such as computer vision, multimedia and surveillance.

Information about the visual scene is acquired by the camera by first focussing and transmitting the light through optical system and then sampling the visual information using a image sensor and analog to digital converter. Typically zoom and focus motors control the focal position of the camera.

In our system we use this camera to capture the images of the vehicles, process them to retrieve the vehicle's registration number and hence the owners data from the central server.

fig 2.1.1

Pi Camera:

This project specifically, uses a pi camera designed by the Raspberry Pi Organisation themselves.

A 5MP camera which directly hooks up onto the raspberry pi, and is ready to use.

Specifications:

- 5 megapixel native resolution sensor-capable of 2592 x 1944 pixel static images
- Supports 1080p30, 720p60 and 640x480p60/90 video

fig 2.1.2

fig 2.1.3

2.2Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

Raspberry Pi is contained on a single circuit board and features ports for:

- o HDMI
- o USB 2.0

o Composite video

o Analog audio

o Power

o Internet

o SD Card

We use this to process the image captured, number plate retrieval and also acts as a mediator between the Image processing using and the central database.

Raspberry Pi has an ARMv6 700 MHz single-core processor, a Video Core IV GPU and 512MB of RAM. it uses an SD card for its operating system and data storage.

The Raspberry Pi officially supports Raspbian, a lightweight Linux OS based on Debian.

General specifications:
- Processor: Broadcom BCM2835
- CPU: ARM 1176JZFS (ARM 11 w/ v6 core, floating pt. @ 700MHz)
- GPU: Video core IV GPU ○ RAM: 512 MB
- USB: 2 USB 2.0
- Network: Ethernet
- Video out: HDMI
- Audio out: 3.5 mm jack
- SD Card Storage (Up to 32GB)
- Micro USB power
- Display Serial Interface Port (DSI)
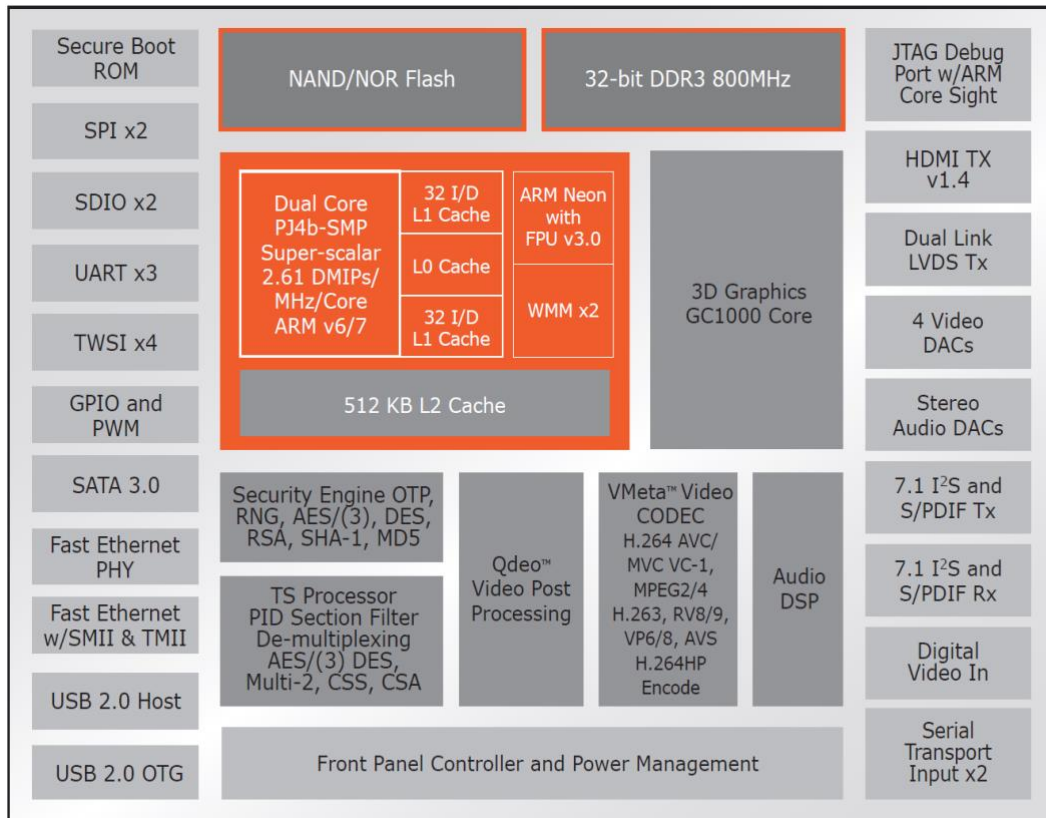- Camera Serial Interface Port (CSI)

fig 2.2.1

fig 2.2.2

2.3 Central Database

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through APIs.

Web applications can be designed for a wide variety of uses and can be used by anyone; from an organization to an individual for numerous reasons. Commonly used Web applications can include webmail, online calculators, or e-commerce shops.

In this Image processing system we use a Python/Django webserver to process the requests sent by the toll collection system, which may include bill generation, payment processing, invoice retrieval, sending notifications to the customers, contacting authorities for various reasons etc.

An SQL server is used for storing the data of the vehicles, digital record keeping of toll transactions and vehicle movements.

For the simplicity of usage of the project, here we use a lite version of SQL database, sqlite3, which resides in the amazon EC2 server.

2.4 Amazon EC2 Linux server

EC2 which stands for Elastic compute cloud is cloud computer platform provided by Amazon web services. This virtual computer helps us in taking requests from the embedded device on the field (raspberry pi here) and process the information.

It is responsible for processing vehicles movement, store details of the same and generating bills and transactions whenever necessary.

This also acts as the medium for the customers to retrieve their data and do bill payments etc.

Specifications for the project:

- Here we use a t2.micro server.

Instance	vCPU*	CPU Credits / hour	Mem (GiB)	Storage	Network Performance
t2.micro	1	6	1	EBS-Only	Low to Moderate

Operating system: Linux
Type: Ubuntu

The static IP assigned to the EC2 server used in this project is:

IP Address: IPv4 18.221.153.188

2.5   Django

Django is a backend web framework in python3. Its main advantage is the creation of complex and database driven websites. The framework helps with better reusability and pluggability of all the components within, less code, rapid development, and basic principle of not to repeat yourself. Python is used throughout the framework, even for settings and data models which will help us manage database design. It also provides an optional administrative create, read, update and delete interface that is generated through introspection and configured through admin models created in the Django project code of each application.

We use this for the case of processing the web requests the raspberry pi device sends to the ec2 server. A continuous python3 process will be running in the port 8000 of our ec2 server, which will receive each request and process it accordingly.

Some web requests include: Creating a user, adding a vehicle, adding toll gate entry of a vehicle, generating a bill on exiting toll plaza etc.

Our Django server will be running at

Address: http://18.221.153.188:8000/

Our Django server will always directly be in contact with a database server which will takes requests from Django to enter, modify, and delete data whenever necessary.

All the data in the database once stored is soft deleted, hence cannot be lost once it enters it in any way.

Some external libraries that here we use with Django wen framework are:

Django Rest API framework:

It is an API development framework which work with Django and helps us in developing web APIs. Since the raspberry pi we use in this project is an independent embedded device from the web server, the Rest Framework will help create a cross connection for exchanging information.

Which in this case will be the images and license plate numbers of the vehicles passing through the toll gate.

Django Cron:

To send the customers information regarding their vehicle movements, the toll bills and wallet balance etc., we need a service which will periodically run a routine and do the things needed.

Django Cron will help us run certain program codes in periodic intervals, either every day or every hour that will intimate the customers and the admin about the things that need immediate attention.

Google distance matrix:

Distance matrix is an API service provided by the company Google.

This Google maps API will help us in calculating distance between two toll plazas, thereby to calculated the toll chargeable.

This will do the same by taking two coordinates in the form of longitude and latitude and return us with the amount of distance between the two points and also the time required to cover it on average.

CHAPTER 3

Proposed Methodology:

3.1 Block Diagram:

Shown below is the block diagram of the project.
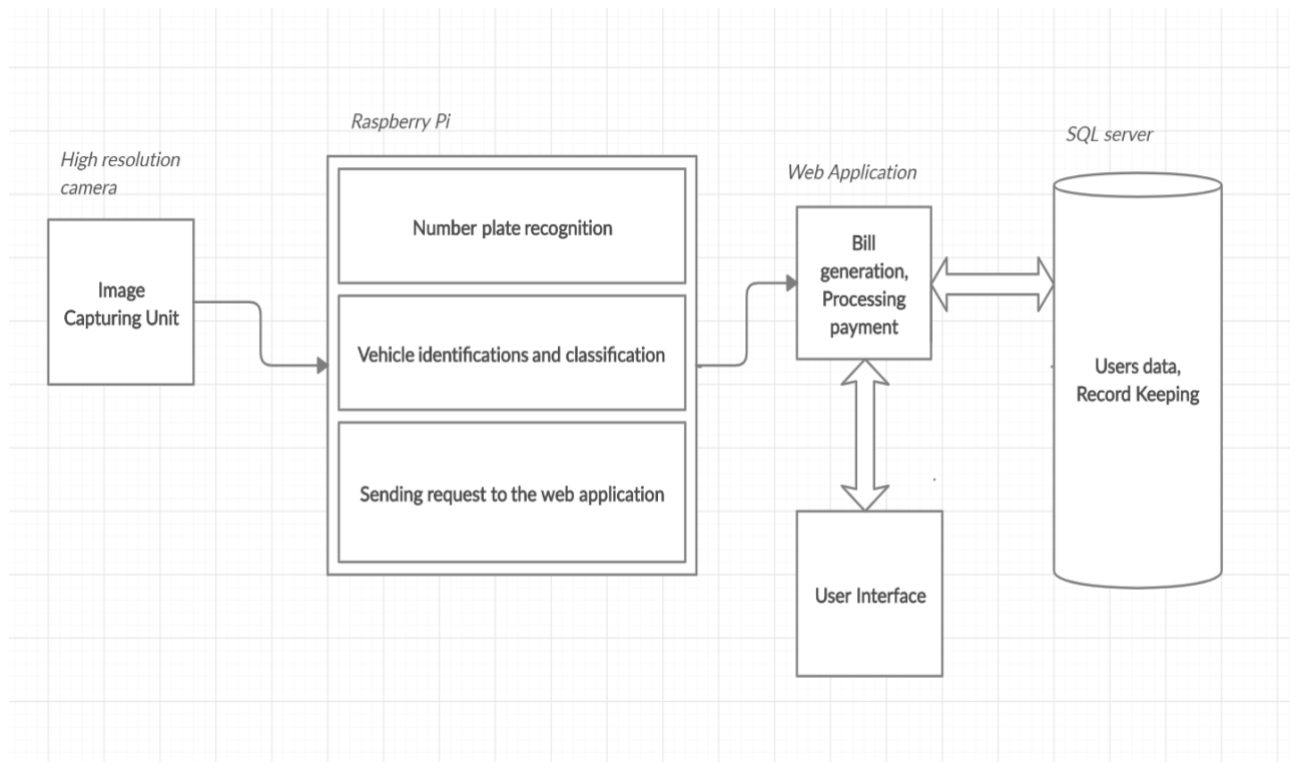

fig 3.1

The contents of the block diagram and their responsibilities:

Image Capturing Unit:

This contains a high resolution camera attached to a raspberry pi. This will be installed at the toll plazas are responsible for capturing images of the vehicles and sharing the same with the Raspberry pi unit, which is our main computing unit at the toll plaza.

Raspberry Pi:

The raspberry pi is responsible for receiving the images from the camera, it then later processes it for extracting the license plate number from it. Once it is done it needs to share the information regarding the toll plaza at the which the vehicle was detected, the entry or exit type, and time at it was captures and send it to the web server.

Web Application:

This block here runs a continues process which will take remote web requests from the raspberry pi device and process the in the required way.

It was responsible for data entry into the database, which it will be in contact with, generating bills, processing payments, creating user accounts, holding vehicle information etc. This will always be directly linked to a database server running parallelly.

Another function of the web server is to provide a User Interface for the customers to manage their account. Which includes managing their vehicles, adding and deleting etc., bill payments, contacting the company when needed etc.

SQL Server:

This block is consists of an SQL database server which runs parallelly with the web server and is always in direct contact with it. This is responsible for remembering data regarding the

vehicles, the toll gates, the user accounts, the toll entries and exits, the bills generated, the transactions that took place, the wallet information of the user etc.

Whenever a data request is done by the web server the SQL server needs to respond back with requested data or raise an error if any problem arises.

CHAPTER 4

Working and Results:

The project seeks to solve the problem of heavy delay and traffic congestion faced at the toll plaza by vehicles trying to hit the toll charged roads.

It consists of an image processing system which constantly monitors and captures the images of the vehicles entering the toll road.

For each vehicle, it then processes its images to retrieve the vehicle's registration number which helps in fetching the owners billing information from our web servers.

As soon as the vehicle enters/leaves the toll chargeable road, a bill would be auto-generated and shared with the owner's email and mobile number. The bill amount would be calculated by the logic we define within the server.

A grace period would be mentioned for the bill to be cleared off. In failure of doing so late fee would be added to the total amount for a level-2 grace period. Further failure in paying the bill would blacklist the vehicle from using the govt roads, and on the next capture of the vehicle the information will be shared to the local road traffic authorities.
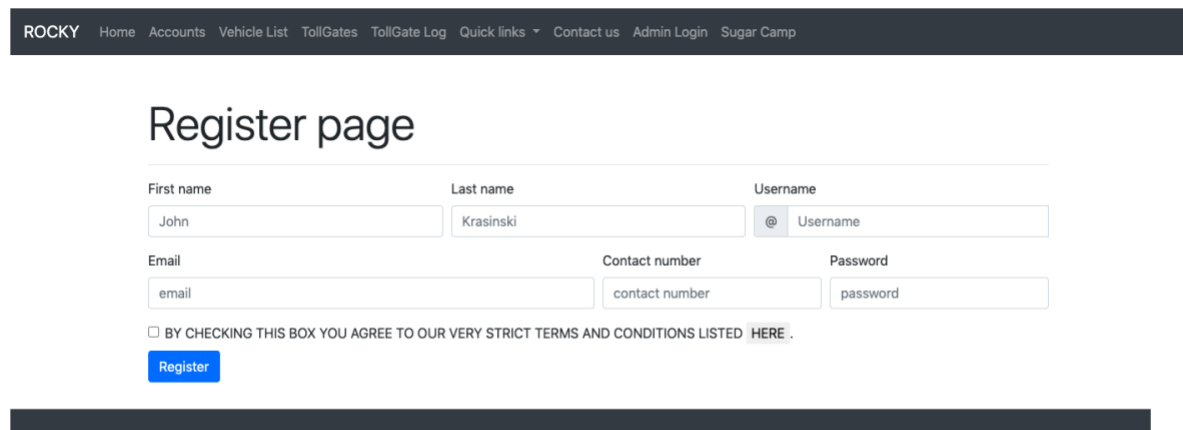
The system further helps in:

- Capturing stolen vehicles.
- Monitoring the movements of the vehicles as needed by the vigilance authorities.

4.1 Step by step procedure of project working:

Below is a step by step procedure of how the process works in the working of project, from a customer Point of View.

The customer first needs to create an account visiting the project website. The date she needs to submit are Name, Email and her Mobile number. A unique username is assigned to each user either by the admin herself or the customers can create by themselves.

Weblink: http://18.221.153.188:8000/register/

Fig. 4.1.1

Before proceeding with anything, the customer is expected to verify her mobile number. Without the mobile number verified the customer cannot go forward with adding a vehicle registered to their name.
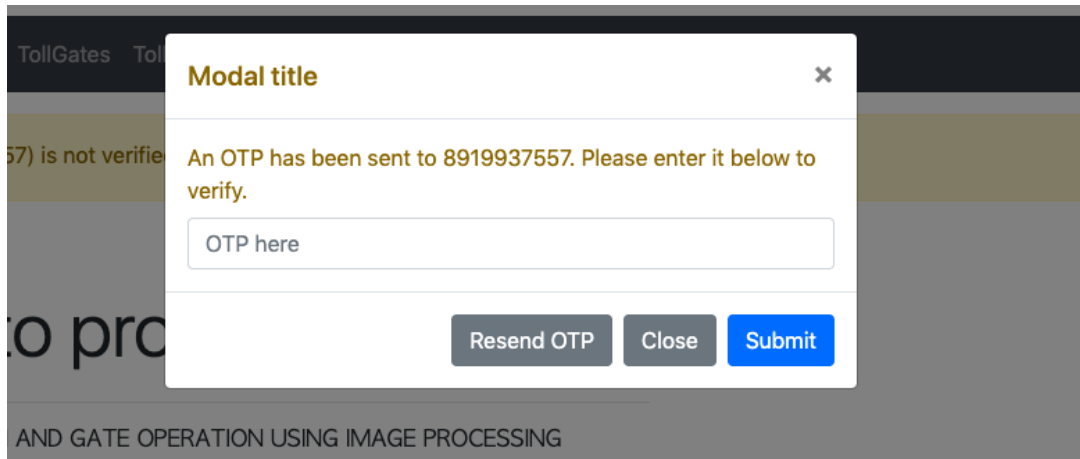
Fig. 4.1.2

The customer can click on the prompt shown on the web page when she is logged in and verify her mobile number.

This helps us prevent spam accounts and automated attacks.

After setting up the account, the customer can start adding her vehicles registered to her name. These vehicles are uniquely linked to one account only, and any bills generated to these vehicles will be charged to the same account.
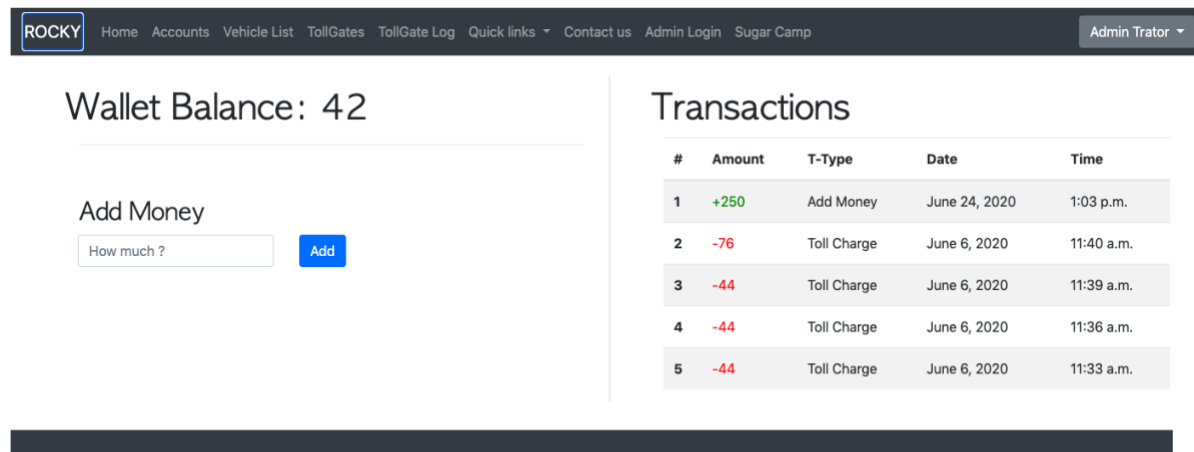
Link: http://18.221.153.188:8000/accounts/add-vehicle/

Fig. 4.1.3

Online Wallet:

- An in-built online wallet in the website will help the customer in managing her payments and tracking them transparently.

- This wallet balance can go into negative side as well, which will help the customers in ensuring smooth payments even when the wallet balance is insufficient for payments.

Link: http://18.221.153.188:8000/accounts/wallet/

Fig. 4.1.4

The website maintains 100% transparency regarding the transactions, and the customer will be able to track her payments no matter how long ago it took place. The customer can use the same link to add funds into her wallet with no upper limit.

The customer can anytime retrieve the information about her vehicles listed at the following link.

http://18.221.153.188:8000/accounts/my-vehicles/

After this, the customer is all set to use her vehicle(s) to enter and exit toll plazas across the country without any resistance. The cameras tracking the vehicles flow will take care of everything.

Whenever the customer leaves through toll plaza after entering one, automatically a bill will be generated.

Customer can obtain this bill from the website directly. The bill amount will be debited from the wallet linked to that account. In case of insufficient funds, the wallet balance will go negative and the same will be intimated to the customer.

The customer will also receive an automated SMS to her registered mobile number regarding the transaction.

The toll billing list can be obtained from:
http://18.221.153.188:8000/billing/

Fig. 4.1.5

The customer can Download the bill by clicking on corresponding link
An example to bill generated:

ROCKY

INVOICE TCKT159362

CUSTOMER	Admin Trator	ROCKY Corporation.
VEHICLE	Toyota X1	607, Street no.3
REG	MH20DV2366	Towli Chowki, Hyderabad
EMAIL	admin@gmail.com	TS 500070, IND
DATE	June 6, 2020	+91 (891)9937557
DUE DATE	July 16, 2020	queries.rocky@gmail.com

TOLLGATE PLAZA	TOLL TYPE	DATE	TIME
Thukkuguda ORR Toll Gate	ENTRY	June 6, 2020	11:40 a.m.
DAYARA Toll booth	EXIT	June 6, 2020	11:40 a.m.

ATTRIBUTE	VALUE
TICKET NUMBER	TCKT159362
TRANSACTION ID	RCKY430753
DISTANCE COVERED	23259.0 meters

AMOUNT	66.67
GST 14%	9.33
GRAND TOTAL	76.0

NOTICE:
A finance charge of 1.5% will be made on unpaid balances after 40 days.

Fig. 4.1.6

4.2 Working of the project Backend:

When any vehicle passes through the toll plaza installed with this project, a picture will be taken of the vehicle by the Pi Camera attached to the Raspberry Pi.
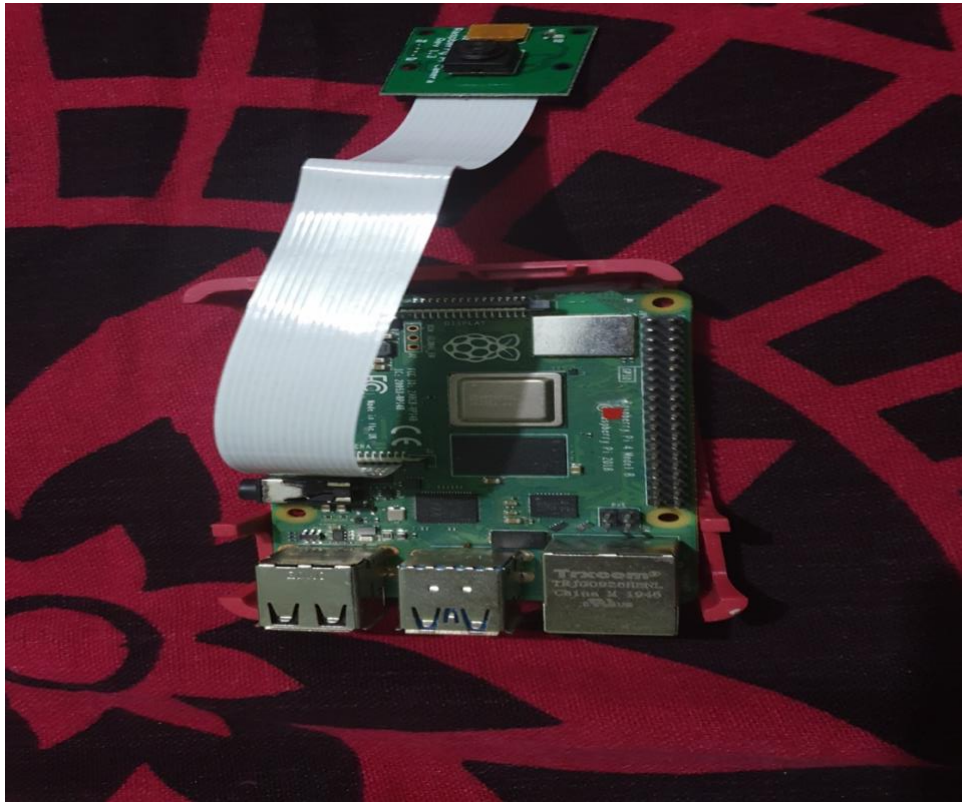
Fig. 4.2.1

The Pi is equipped with software written using Open CV library of Python which will help the Pi in extracting the License plate registration number in the form of a text string.
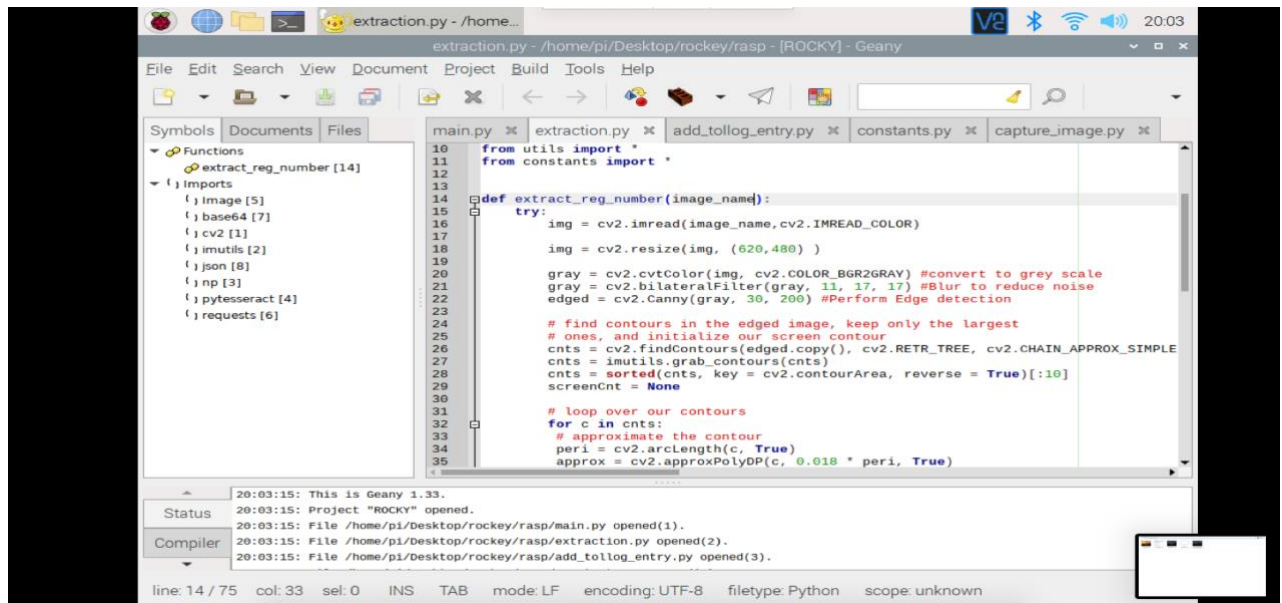
Fig. 4.2.2

The registration number along with a request of Entering/Exiting a toll gate will be sent to the web server which will add the record to it. The server will make some basic checks and do the process or respond throwing an error.

If the toll gate record type is an exit, then a the distance between the toll gates the entry and exit will be calculated and the amount will be calculated accordingly. A bill will be generated and the amount will the debited from the wallet associated to the vehicle.

The same will be sent to the customer through an SMS to the registered mobile number.

4.3 Pictures taken by the Pi camera:

The Raspberry Pi camera takes pictures of the moving vehicles as the go past the gate and share it with the Computer device.

A sample picture taken by the camera looks as follows.

Fig. 4.3.1

This will be sent to the raspberry pi immediately as it is captured.

4.4 Working code of the raspberry pi device:

The raspberry pi, connected to the high resolution pi camera, processes the images in python3 languages and uses the same to share the information with the web server.

The working code of the Pi in a top down approach manner can be seen below:

This helps in taking the picture of the vehicle and send the same for licence plate extraction., and then make a web request to the Web Server.

```python
# import picamera
import datetime
import constants
from extraction import extract_reg_number
from add_tollog_entry import add_tollog_entry

def capture_image():
        with picamera.PiCamera() as camera:
                camera.start_preview()
                try:
                        name = str(datetime.datetime.now())
                        camera.capture(name)
                except:
                        print(constants.CAPTURE_FAILED)
        exit()
        return name

def main():
        print("-- STARTING UP --")
        # image_name = capture_image()
        image_name = "new.jpg"
        # reg_number = extract_reg_number(image_name)
        reg_number = 'MH20DV2366'
        print(reg_number)
        add_tollog_entry(reg_number, constants.Ghatkesar_Toll_Plaza_ORR,
constants.TTYPE_EXIT)


main()
```

```python
import cv2
import imutils
import numpy as np
import pytesseract
from PIL import Image
import requests
import base64
import json

from utils import *
from constants import *
```

```python
def extract_reg_number(image_name):
        img = cv2.imread(image_name,cv2.IMREAD_COLOR)

        img = cv2.resize(img, (620,480) )

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #convert to grey scale
        gray = cv2.bilateralFilter(gray, 11, 17, 17) #Blur to reduce noise
        edged = cv2.Canny(gray, 30, 200) #Perform Edge detection

        # find contours in the edged image, keep only the largest
        # ones, and initialize our screen contour
        cnts = cv2.findContours(edged.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
        cnts = imutils.grab_contours(cnts)
        cnts = sorted(cnts, key = cv2.contourArea, reverse = True)[:10]
        screenCnt = None

        # loop over our contours
        for c in cnts:
         # approximate the contour
         peri = cv2.arcLength(c, True)
         approx = cv2.approxPolyDP(c, 0.018 * peri, True)

         # if our approximated contour has four points, then
         # we can assume that we have found our screen
         if len(approx) == 4:
          screenCnt = approx
          break

        if screenCnt is None:
         detected = 0
         print ("No contour detected")
        else:
         detected = 1

        if detected == 1:
         cv2.drawContours(img, [screenCnt], -1, (0, 255, 0), 3)

        # Masking the part other than the number plate
        mask = np.zeros(gray.shape,np.uint8)
        new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
        new_image = cv2.bitwise_and(img,img,mask=mask)

        # Now crop
        (x, y) = np.where(mask == 255)
        (topx, topy) = (np.min(x), np.min(y))
        (bottomx, bottomy) = (np.max(x), np.max(y))
        Cropped = gray[topx:bottomx+1, topy:bottomy+1]

        #Read the number plate
        text = pytesseract.image_to_string(Cropped, config='--psm 6')
        text = remove_chars_from_string(text, UNWANTED_CHARS)
        reg_number = text.replace(' ', '')
        reg_number = remove_chars_from_string(reg_number, UNWANTED_CHARS)
        print("Detected Number is:",reg_number)
```

```
        return str(reg_number)
```

The code above uses python3 library OpenCV, and is solely responsible for extraction of license plate number from the image.

add_toll_entry.py

```python
import requests
import json
import constants


def add_tollog_entry(vehicle_reg, tollgate_id, ttype):
        headers = {
                                "content-type":"application/json",
                        }
        data = json.dumps({
                                "tollgate_id": tollgate_id,
                                "vehicle_reg": vehicle_reg,
                                "ttype": ttype
                        })

        resp = requests.post(
                                                url     =
constants.ADD_LOG_URL,

                                                headers = headers,
                                                data    = data
                        )
        print(resp.json())
```

The program above shows how a web request is sent to the Django server regarding adding of a toll gate entry into the database table.

utils.py

```python
def remove_char_from_string(string, char):
        returner = ''
        for ch in string:
                if ch != char:
                        returner += ch
        return returner

def remove_chars_from_string(string, chars):
```

```
    returner = ''
    for ch in string:
            if ch not in chars:
                    returner += ch
    return returner
```

The above utility functions are used to clean the license plate number from any unwanted characters included.

4.5 Extracting the license plate number:

The picture will be grey scaled and processed by the raspberry pi and the license plate is extracted in following way
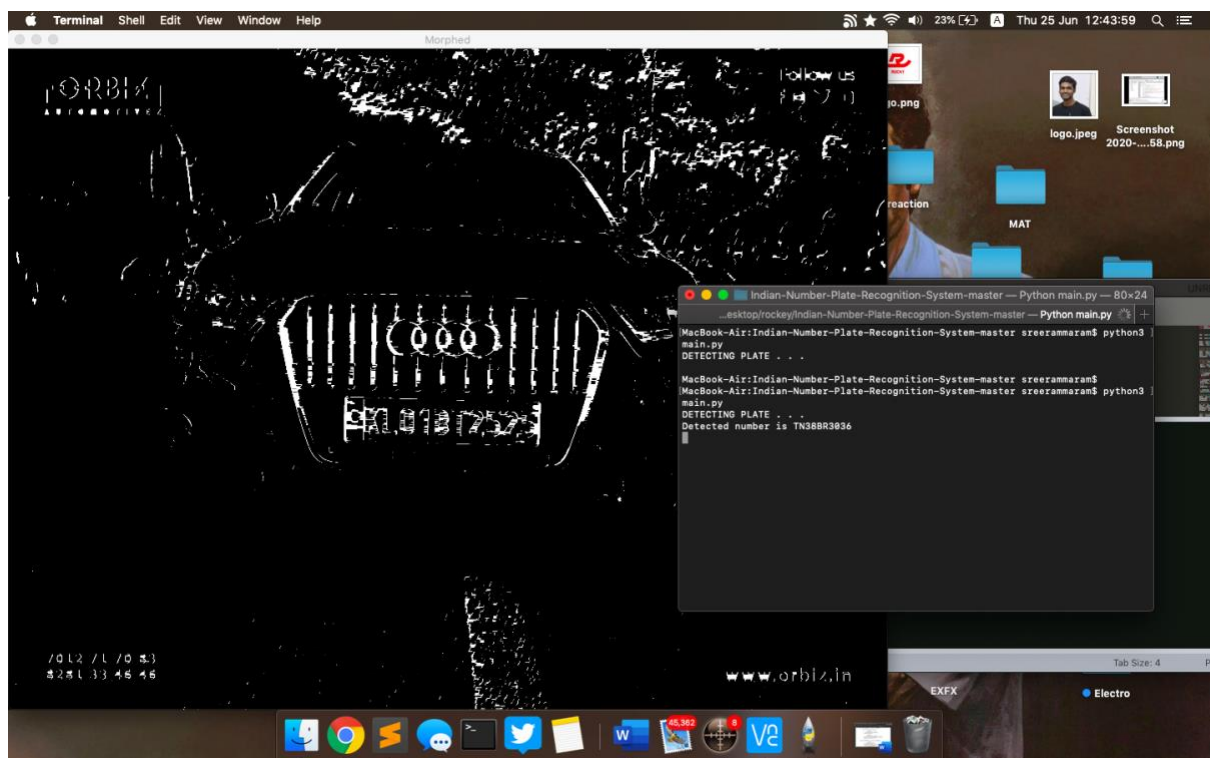
Fig. 4.5.1

In case the device fails to extract the license plate number, it will save the image and intimate the admin if the system about the same at a later time.

Every picture ever taken by the camera device will be stored in the raspberry pi itself in a folder games 'images_captured' in the project root directory.

4.6 SQL Database structure:

The database will store all the data being given by the web server and retrieves the same whenever necessary.

A Graphical user interface helps us go through the data concretely without even help of the Django server which usually acts as a mediator.

Fig. 4.6.1

Fig. 4.6.2

Following is the Entity-Relationship model Diagram of the database create for the current project. The database well normalized to make the retrieval efficient and lossless at the same time.
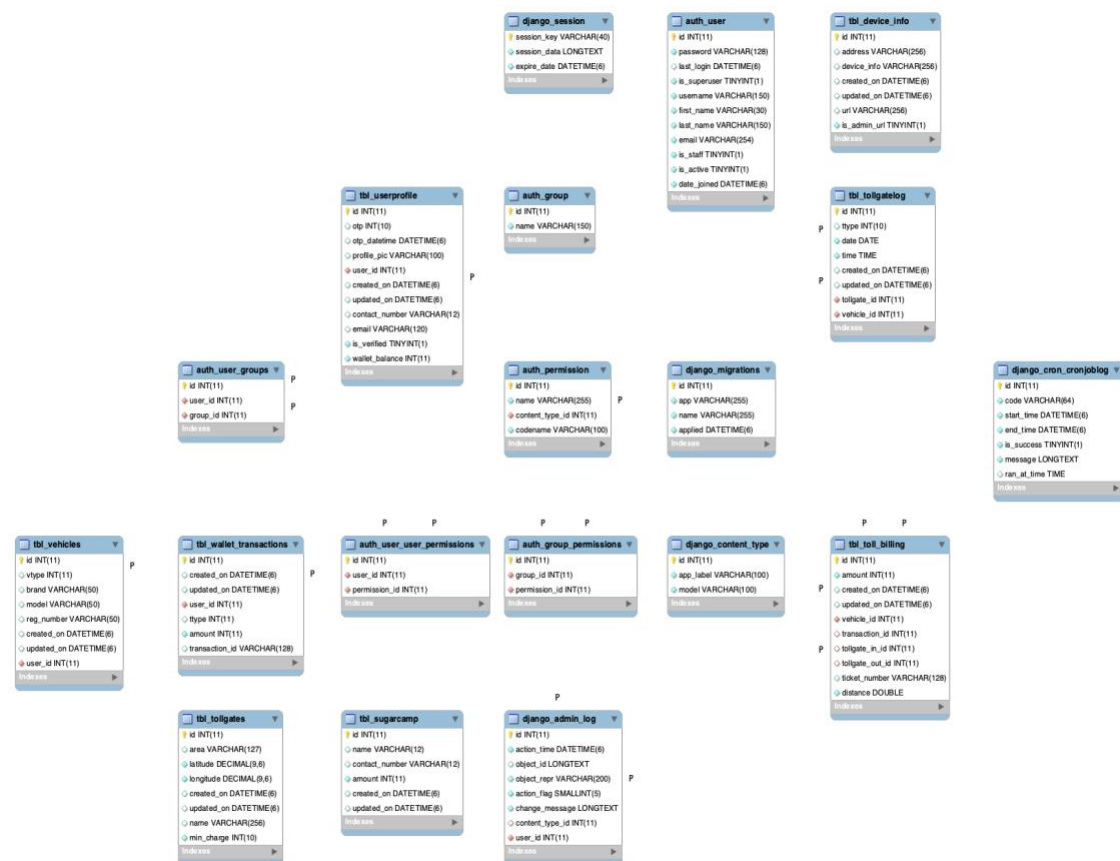
Fig. 4.6.3
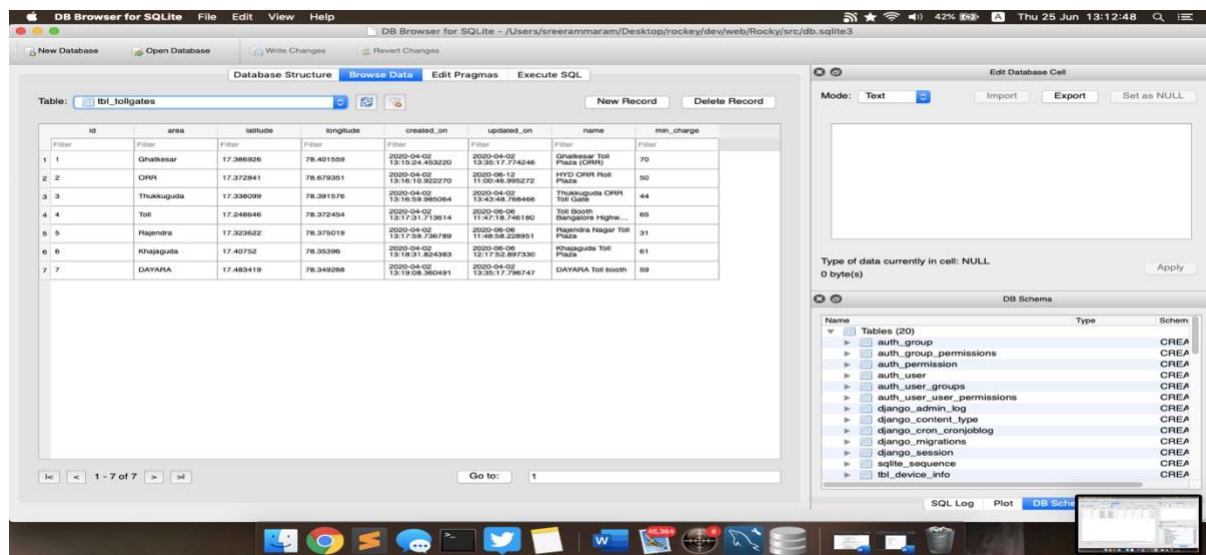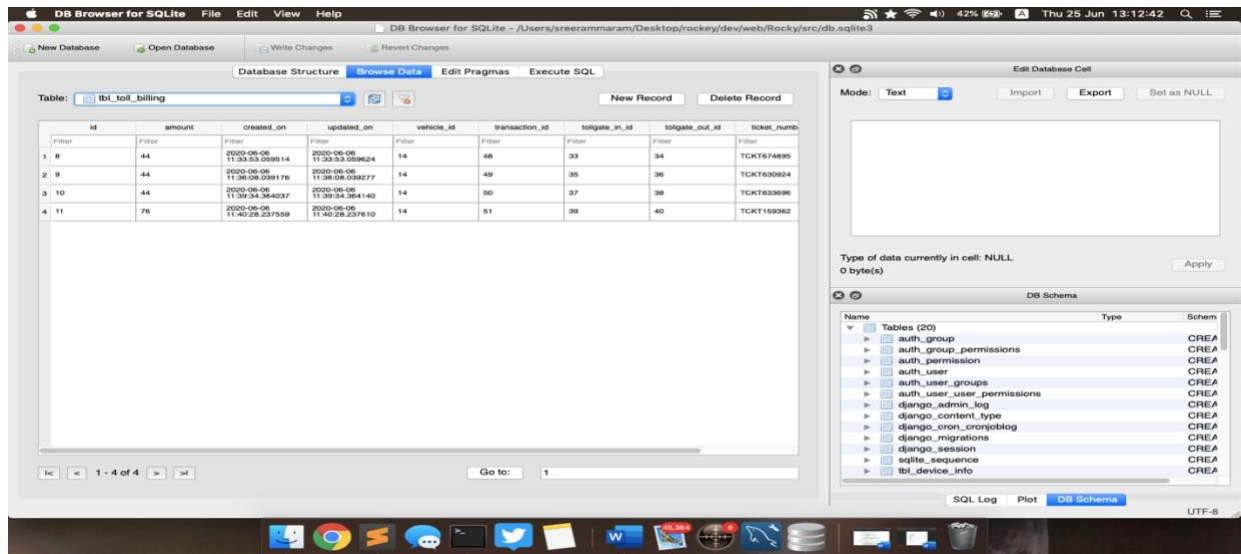
4.7 Working code of the Django Web server:

A continuous python process will be running on the Amazon EC2 server on port 8000. This acts as a mediator between raspberry pi device and the database system.

The project root directory of the web server looks as following:
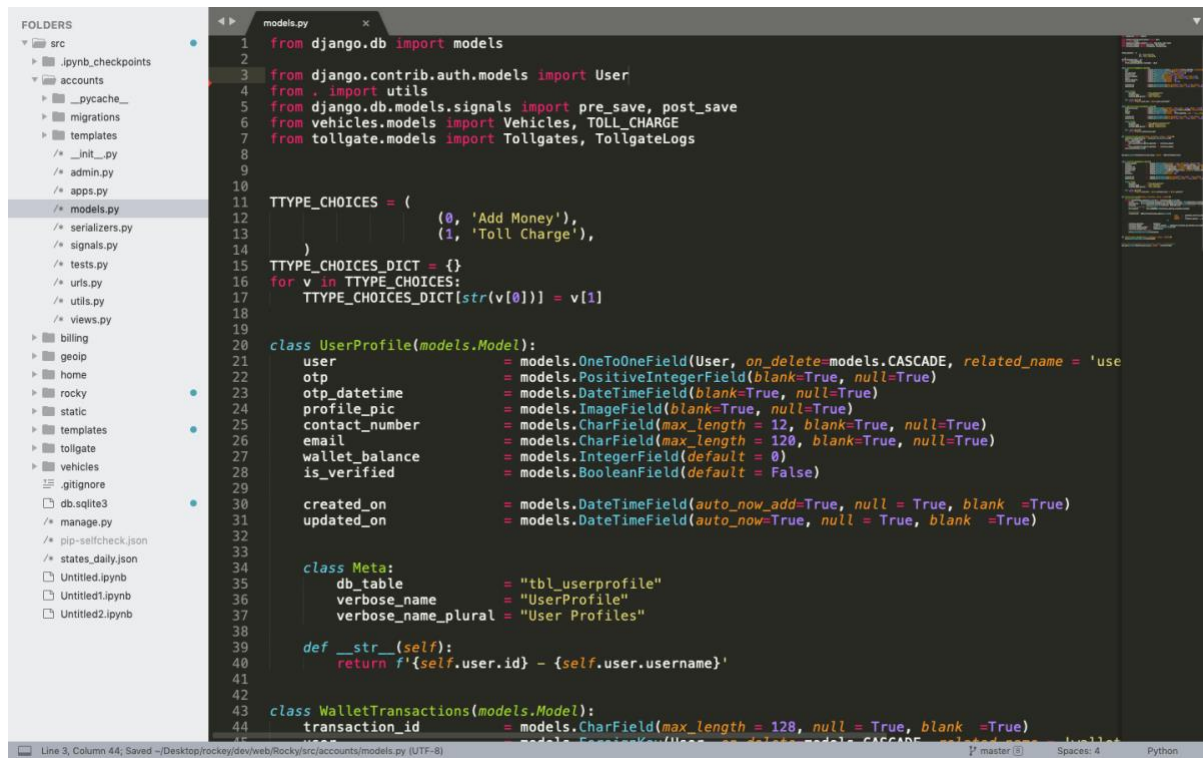
Fig 4.7.1

The application has the internal apps created such as Accounts, Billing, Tollgate, Vehicles etc. which each of it has certain responsibilities and functions assigned.

Example of handling an API request by the server when raspberry pi device sends it request regarding adding of a toll gate entry.

```python
class TollgateLogs(models.Model):
    tollgate                          =
    models.ForeignKey(Tollgates, on_delete=models.CASCADE, related_name
= 'log')
    vehicle                           =
    models.ForeignKey(Vehicles, on_delete=models.CASCADE, related_name
= 'log')
    ttype                             =
    models.PositiveIntegerField(choices = TOLL_TYPE, null = True, blank
=True)
```

```python
        date                               =
        models.DateField(default = curr_date)
        time                               =
        models.TimeField(default = curr_time)

        created_on              =   models.DateTimeField(auto_now_add =
True, null = True, blank  =True)
        updated_on                         =
        models.DateTimeField(auto_now = True, null = True, blank  =True)


        class Meta:
                db_table            = "tbl_tollgatelog"
                verbose_name        = "TollgateLog"
                verbose_name_plural = "TollgateLogs"


        def __str__(self):
                return f'{self.id}:
TollgateLog:{TOLL_TYPE_DICT[str(self.ttype)]}  {self.vehicle} - {self.date}
- {self.time}'



class AddTollLogByNumberSerializer(serializers.Serializer):
        tollgate_id               =         serializers.IntegerField()
        vehicle_reg               =
        serializers.CharField()
        ttype                     =
        serializers.IntegerField()

        def validate_tollgate_id(self, tollgate_id):
                if not models.Tollgates.objects.filter(id =
tollgate_id).exists():
                        raise serializers.ValidationError(f"Tollgate does
not exist for id:{tollgate_id}")
                return tollgate_id

        def validate_ttype(self, ttype):
                if not ttype in [t[0] for t in models.TOLL_TYPE]:
                        raise serializers.ValidationError(f"Not a valid toll
type :{ttype}")
                return ttype



class TollgateLogAddByNumberAPIView(APIView):

        def post(self, request):
                # print("HIT ADD REG")
                request_data   =      self.request.data
                serializer          =
        AddTollLogByNumberSerializer(data = request_data)
                if not serializer.is_valid():
                        raise exceptions.ValidationError(serializer.errors)

                tollgate       = Tollgates.objects.get(id =
serializer.data["tollgate_id"])
```

```python
                ttype              =
TOLL_TYPE_DICT[str(serializer.data["ttype"])]

                vehicle_reg = serializer.data["vehicle_reg"].upper()
                vehicles      = Vehicles.objects.filter(reg_number =
vehicle_reg)
                if not vehicles.exists():
                        raise APIException(f"No vehicle found for
reg_number: {vehicle_reg}")
                vehicle        = vehicles.first()

                tollgate_log = TollgateLogs.objects.create(

                                    tollgate = tollgate,

                                    vehicle  = vehicle,

                                    ttype    =
str(serializer.data["ttype"])

                                )

                return response.Response(f'''Tollgate:
{tollgate.name.upper()}, Vehicle: {vehicle.brand.upper()} |
{vehicle.reg_number.upper()},ttype: {ttype.upper()}''')
```

The above server code handles the web request made by raspberry pi device shown on page no. 31.

It is responsible for testing the authenticity of the data, for example if the license plate number is valid or not, if the specific vehicles lies in our database, if the toll gate ID is legitimate, the correct attributes of each database element is followed or not etc,

In case of any errors, the server need to respond back with appropriate response code and text, and show not let the system crash at any moment of time.

4.8 Project Repositories:

Django Project Git Repository:

The Rest of project code can be found at the following repository:

https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Rocky

Raspberry Pi Git Repository:

The software present in the raspberry pi device can be found/cloned at the following link:

https://github.com/sreeram315/RockyRaspberry

Note:
Both the Repositories are private and the credentials will be created and shared upon request.

4.9 Advantages:

The project will help in minimizing or even completely eliminating the traffic congestion faced by the vehicles at the toll plaza. This will ensure a smooth flow of vehicles there by saving time, reducing the human stress and effort into this.

Automation of toll charges billing will ensure the billing is done at super-fast speed since it is no more need human intervention. Auto-deduction of amount as needed can be done.

In modern times, security becomes the most important aspect while working with any kind of digital payments. This project will help in securing and digitalizing payments through secure gateways.

Human errors are common and always possible when people work with computers. The automated computerized system will help eliminate that entirely.
This also minimizes the need of man power.

Monitoring of the vehicle movements can be done which will help in solving complaints and can also be used for vehicle tracking by government agencies if required.

Centralizing information, Record keeping using databases becomes the biggest plus of using the present system. Digital records of vehicle movement and payments will ensure they are permanent and help in case whenever some information needs to be retrieved.

CHAPTER 5

5.1 Future Scope:

Secure record keeping of vehicles entering the toll plaza, as well as vehicle moments help in: Data analytics, which will help us realise some useful information which can be used to help regulate traffic in the future times.

Help police forces to monitor vehicle movements any time in the past which may help in solving some cases.

Development around the present system to computerize the traffic system more and more.

5.2 Conclusion:

The primary objective of the project is to automate the process of toll collection. The hardware system we are going to develop is going to help us with the same.

While meeting the primary objective of the project i.e. to reduce the time spent by the vehicles at the toll plaza, we also by means of it help in fulfilling other things like securing payments, vehicle monitoring etc.

References:

1. http://ijcsn.org/IJCSN-2016/5-2/Vehicle-Counting-and-Automated-Toll-Collection-System-using-Image-Processing.pdf

2. http://www.iaeme.com/MasterAdmin/UploadFolder/IJCET_09_03_015/IJCET_09_03_015.pdf

3. http://ijarcet.org/wp-content/uploads/IJARCET-VOL-4-ISSUE-5-1922-1927.pdf

4. http://meseec.ce.rit.edu/551-projects/spring2017/2-3.pdf