1. INTRODUCTION

Automatic Toll Collection Using Optical Code Recognition and Encryption provides an alternative to the current existing toll collections techniques and tries to overcome the drawbacks of these system. In this system, the number plate of a particular vehicle is encrypted using DES algorithm and an OCR code is generated out of it. High Definition (HD) cameras capture vehicle number plate and image processing is performed using Open CV functions on the code. It is then matched in the database. If matched, the tax is debited from the users account and the vehicle is allowed to pass within seconds. If the balance is insufficient then only once pass is allowed. After that the payment has to be made manually.

1.1 Problem Summary

Automation being the need of today has made tremendous advancement in all most all areas. One of which is Automatic toll collection which is rapidly becoming a globally accepted toll collection system. Initial toll collection being manual in nature and time consuming, thereby leads to traffic congestion, pollution and a lot of frustration. Quite obviously, this process makes it necessary for the manual services to be rendered continuously through the day and night. Ever increasing need for efficient, reliable, safe, and cheap toll taxation along with image processing resulted in the development of different kinds of solutions. Automatic Toll Collection System using Optical Code Recognition and Encryption aims at successfully removing unnecessary traffic delays, faster and reliable processing, keeping an eye on unauthorized vehicles, etc. In this paper we present a model based on image processing which can be employed for real time automatic vehicle toll collection system. A High Definition (HD) camera is used to capture an OCR code of an incoming vehicle's windshield. This is followed by the process of feature extraction wherein a particular feature, say the OCR code on the windshield is extracted using suitable techniques. Verification is done and deduction process is carried out thereby allowing the vehicles to pass.

1.2 Aims and Objective

- When the vehicle arrives on the toll plaza at a particular distance the sensor senses and the digital camera captures the OCR image on the vehicle.

- The OCR Code is processed using several techniques.

- Encrypted data on the image is matched with the encrypted data in the database. If matched, the data is decrypted using the same algorithm as used for encryption and the details of the vehicle are identified.

- If there is sufficient balance then the appropriate amount is deducted from the account. The details of amount deducted and balance are displayed on the screen.

- Once the balance is deducted, the vehicle is allowed to pass.

- In case, if the account is nil, then the vehicle is allowed to pass exactly once and the balance will go in minus. On the next journey he/she should pay the appropriate amount in cash immediately. Then only he/she is allowed to pass the toll plaza.

1.3 Problem Specification

User Authentication: When the vehicle arrives on the toll plaza at a particular distance the sensor senses and the digital camera captures the OCR image on the vehicle. The OCR Code is processed using several techniques.

Encryption: Encrypted data on the image is matched with the encrypted data in the database. If matched, the data is decrypted using the same algorithm as used for encryption and the details of the vehicle are identified.

Digital Payment: If there is sufficient balance then the appropriate amount is deducted from the account. The details of amount deducted and balance are displayed on the screen. Once the balance is deducted, the vehicle is allowed to pass. In case, if the account is nil, then the vehicle is allowed to pass exactly once and the balance will go in minus. On the next journey he/she should pay the appropriate amount in cash immediately. Then only he/she is allowed to pass the toll plaza.

1.4 TECHNOLOGY AND LITERATURE REVIEW

The Key Technologies used in developing this project are:

1.4.1 Raspberry 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.

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.

1.4.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception.

When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

1.4.3 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of “My", the name of co-founder Michael Widenius’s daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL and Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

1.5 Plan of Work

In order to meet your objectives, we are going to create an android application which will provide information on farming and provide the farmers with various different kinds of knowledge like latest farming techniques, pests control information, FAQs etc... We will also create a website for a farming expert who will be able to modify information on the app and keep farmers up to date with the latest information.

Activity / Time	Phase 1 (1 Month)	Phase 2 (2 Month)	Phase 3 (6 Month)	Phase 4 (2 Month)	Phase 5 (1 Month)
Problem statement					
Design application and collection of data for database					
Coding					
Testing and reviewing					
Report writing					

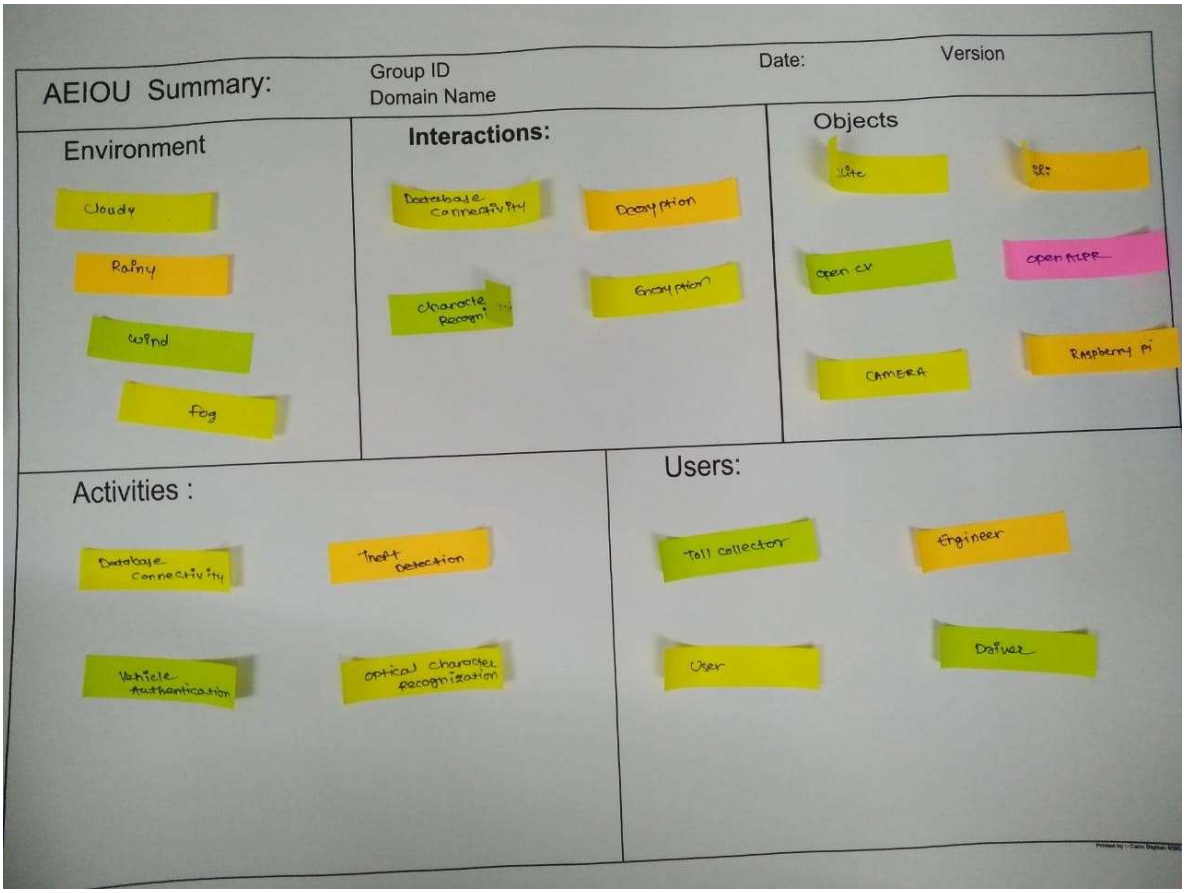
Phase 1:	Reviewing of problem statement.
Phase 2:	Designing of application and data gathering.
Phase 3:	Coding the logic behind the application.
Phase 4:	Testing of the application for bugs and errors and reviewing.
Phase 5:	Paper work like report writing etc.

1.6 Materials and Tools Required

In order to create an embedded system we are going to use raspberry pi, for the database we are going to use MySQL database, for coding in raspberry pi we are using python language. We also used camera & GSM module.

2 Designs: Analysis, Design methodology and Implementation Strategy

2.1 Observation Matrix (AEIOU Summary)



2.1.1 Activity

Activities are goal-directed sets of actions—paths towards things people want to accomplish. It is the modes people work in, and the specific activities and processes they go through

- Database connectivity
- Theft Detection
- Vehicle Authentication
- OCR

2.1.2 Environment

Environments include the entire arena where activities take place. It is the character and function of the space overall, of each individual's spaces, and of shared spaces

- Cloudy
- Rainy
- Fog
- Wind

2.1.3 Interaction

Interactions are between a person and someone or something else; they are the building blocks of activities. It is special interactions between people, objects in their environment.

- Encryption
- Decryption
- Database connectivity
- Character Recognition.

2.1.4 Object

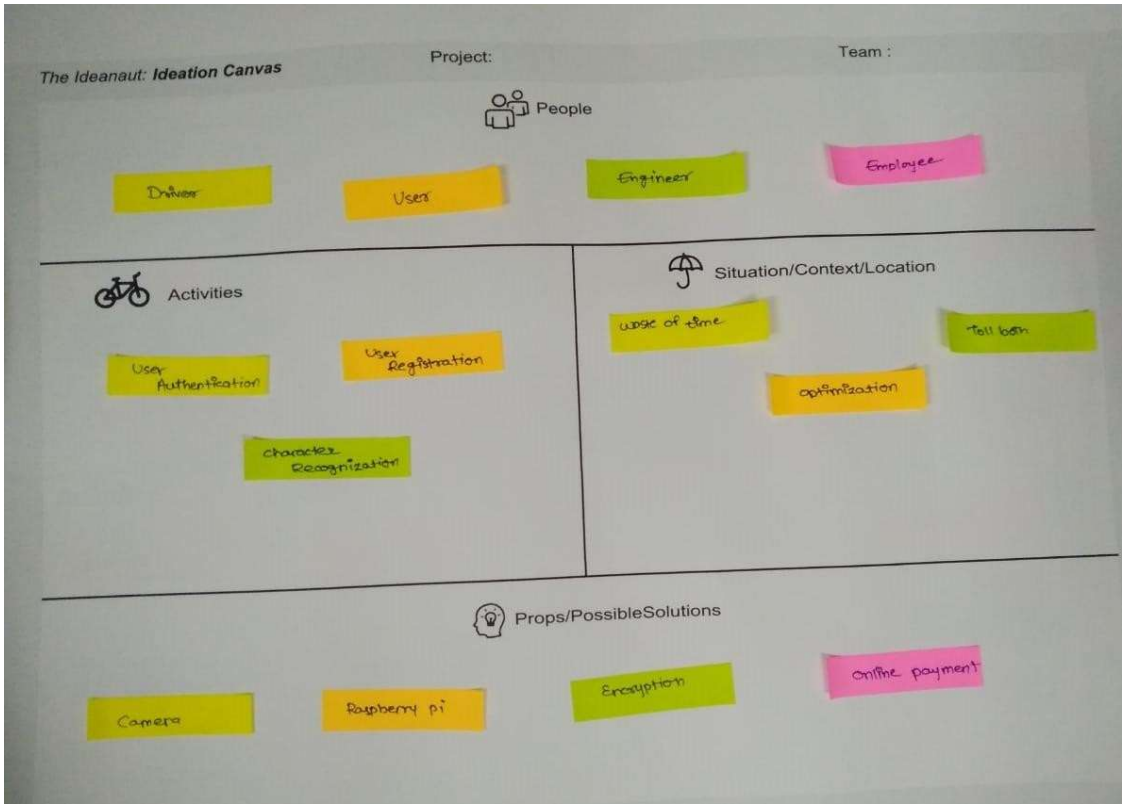
Objects are building blocks of the environment, key elements sometimes put to complex or unintended uses (thus changing their function, meaning and context). They are the objects and devices people have in their environments and the way they relate to their activities.

- Open CV
- Wi-Fi
- SQLite
- Open ALPR

2.1.5 Users

- Engineer
- User
- Driver
- Toll collector

2.2 Ideation Canvas



2.2.1 People

- Driver
- User
- Engineer
- Employee

2.2.2 Activities

- User authentication
- User registration

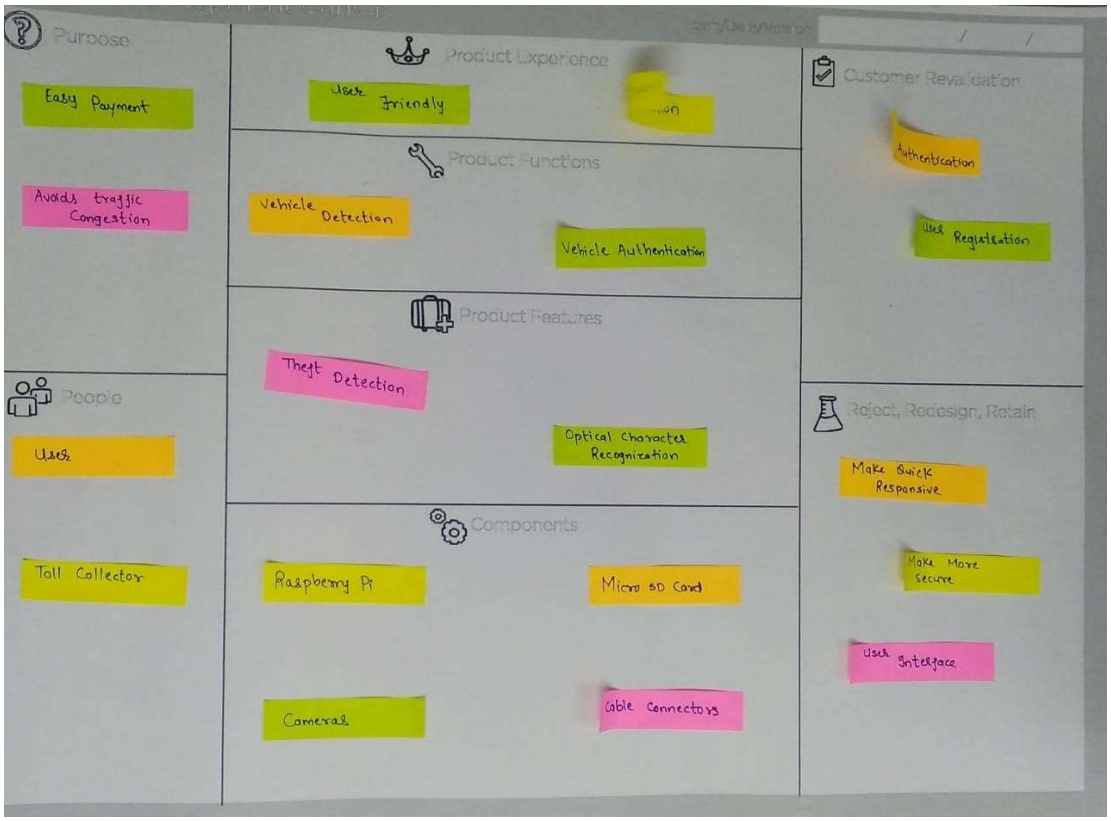
2.2.3 Character recognitionSituation/Context/Location

- Waste of time
- Optimization
- Toll booth

2.2.4Props/Possible Solutions:

- Camera
- Raspberry pi
- Encryption
- Online payment

2.4 Product Development Canvas



2.4.1 Purpose

- Easy payment
- Avoid traffic congestion

2.4.2 People

- User
- Toll collector

2.4.3 Product Experience

After the introduction of this product, the user will feel relaxed.

- Reliable
- User friendl

2.4.4 Product Function

- Vehicle detection
- Vehicle Authentication

2.4.5 Product Features

- Theft Detection
- OCR

2.4.6 Components

- Raspberry pi
- Micro SD card
- Cameras
- Cable connectors

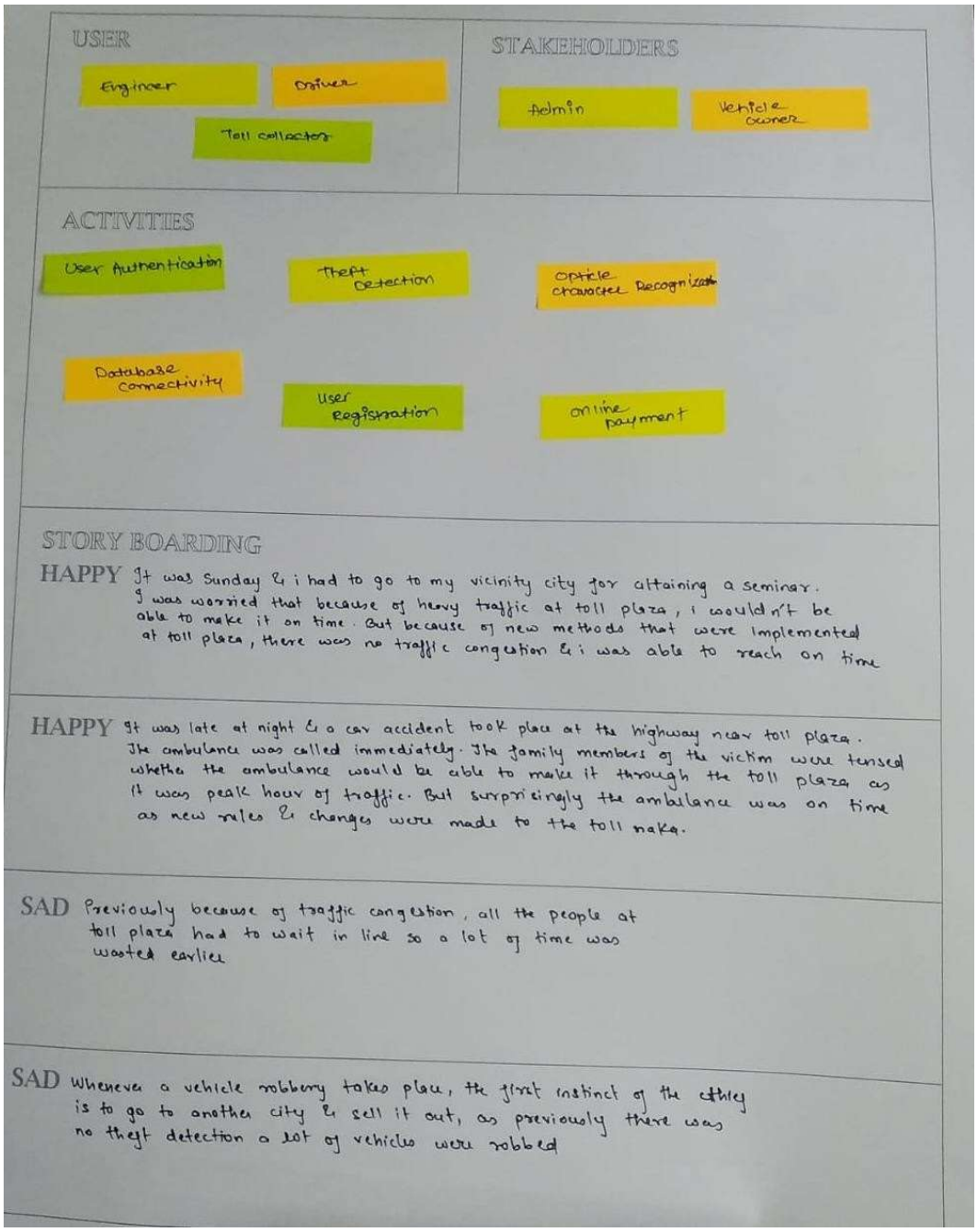
2.4.7 Customer Revalidation

- Authentication
- User registration

2.4.8 Reject, Redesign, Retain

- Make quick responsive
- Make more secure
- User interface

2.5 Empathy Mapping Canvas



2.5.1 Users

- Engineer
- Driver
- Toll collector

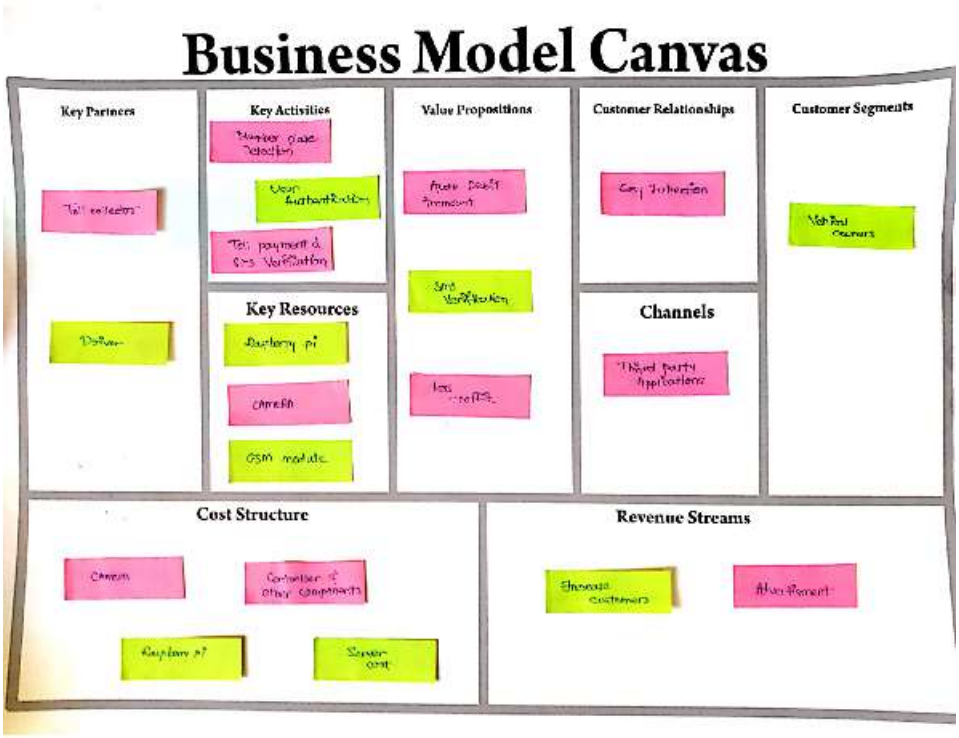
2.5.2 Stakeholders

- Admin
- Vehicle owner

2.5.3 Activities

- User Authentication
- Theft Detection
- OCR
- Database Connectivity
- User registration
- Online Payment

2.6 Business model canvas



2.6.1 Key Partners

- Toll collector
- Driver

2.6.2 Key Activities

- Number Plate Detection
- User Authentication
- Tall Payment and SMS Verification

2.6.3 Key Resources

- Raspberry pi
- Camera
- GSM Module

2.6.4 Value Proposition

- Auto Debit Amount
- SMS Verification
- Less Traffic

2.6.5 Customer Relationship

- Easy Interaction

2.6.6 Channels

- Third Party Application

2.6.7 Customer Segments

- Vehicle Owners

2.6.8 Cost Structure

- • Camera
- • Controller & Other Components
- • Raspberry pi
- • Server Cost

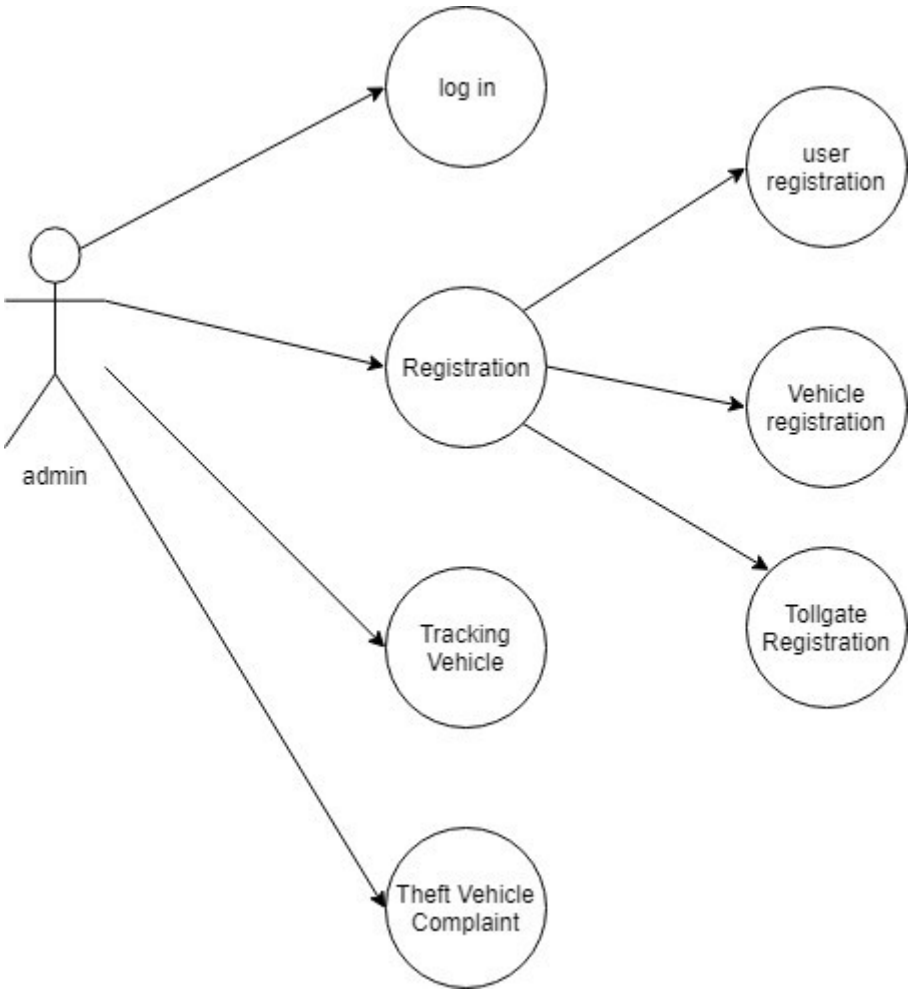
2.6.9 Revenue Streams

- Increase Customers
- Advertisement

2.6 Diagrams

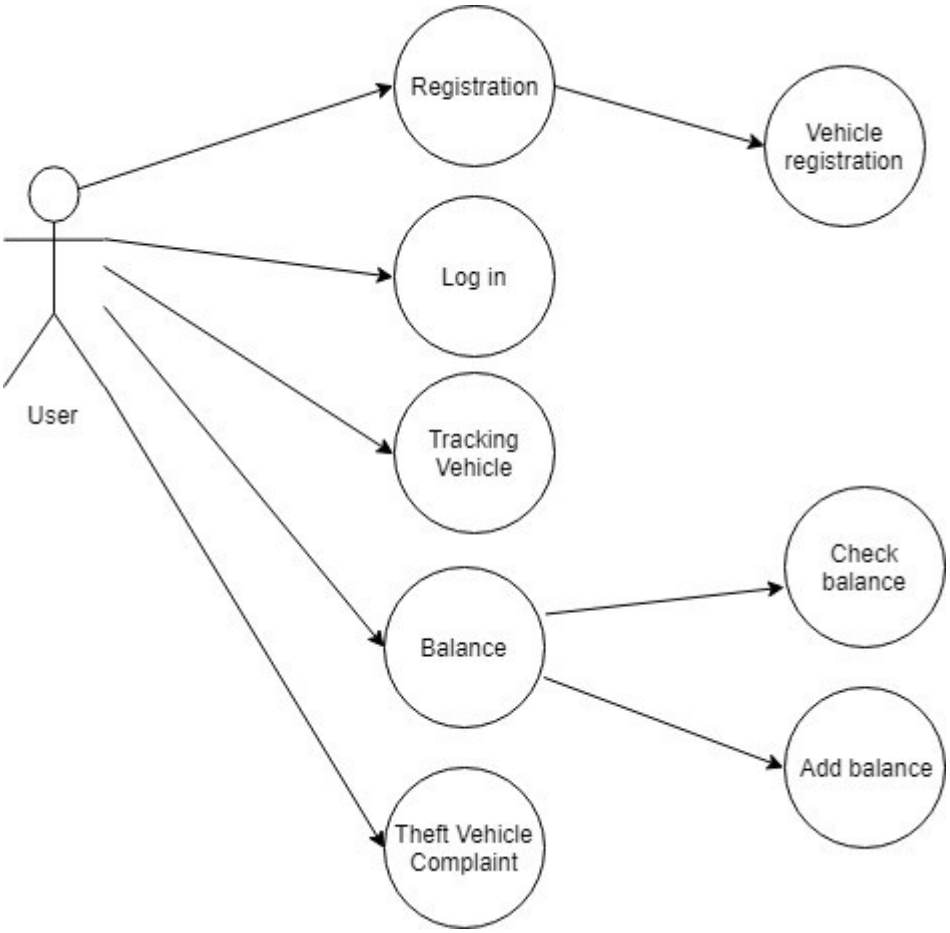
2.6.1 Use Case Diagrams

Admin:



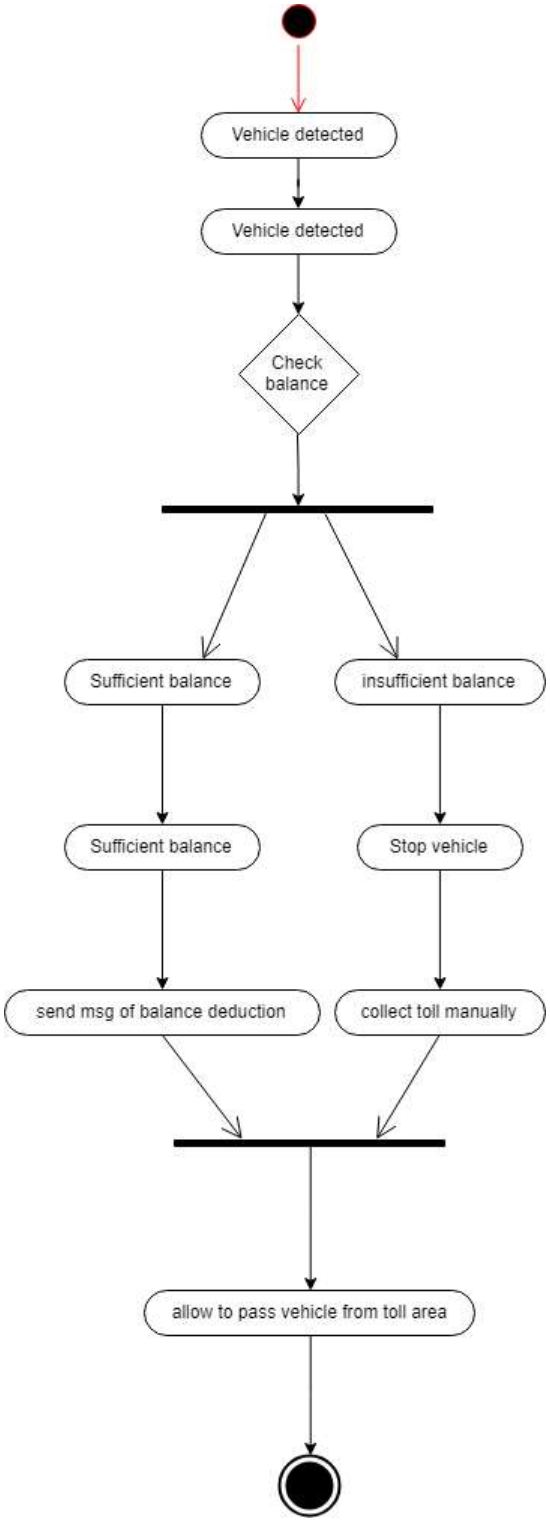
Admin Use Case Diagram

User:



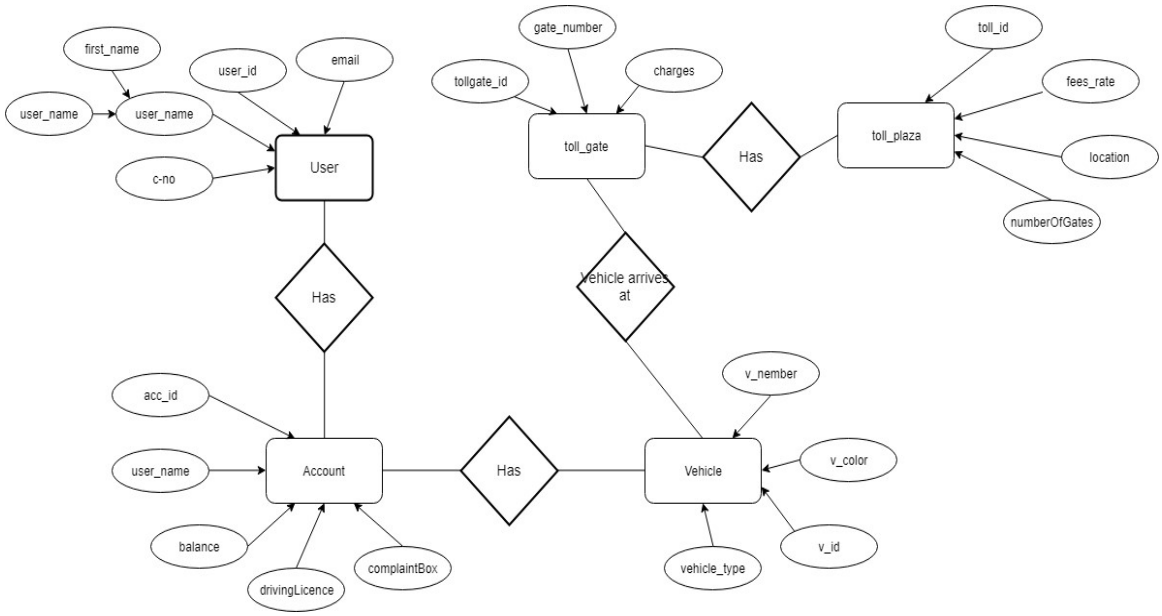
User Use Case Diagram

2.6.2 Activity Diagrams



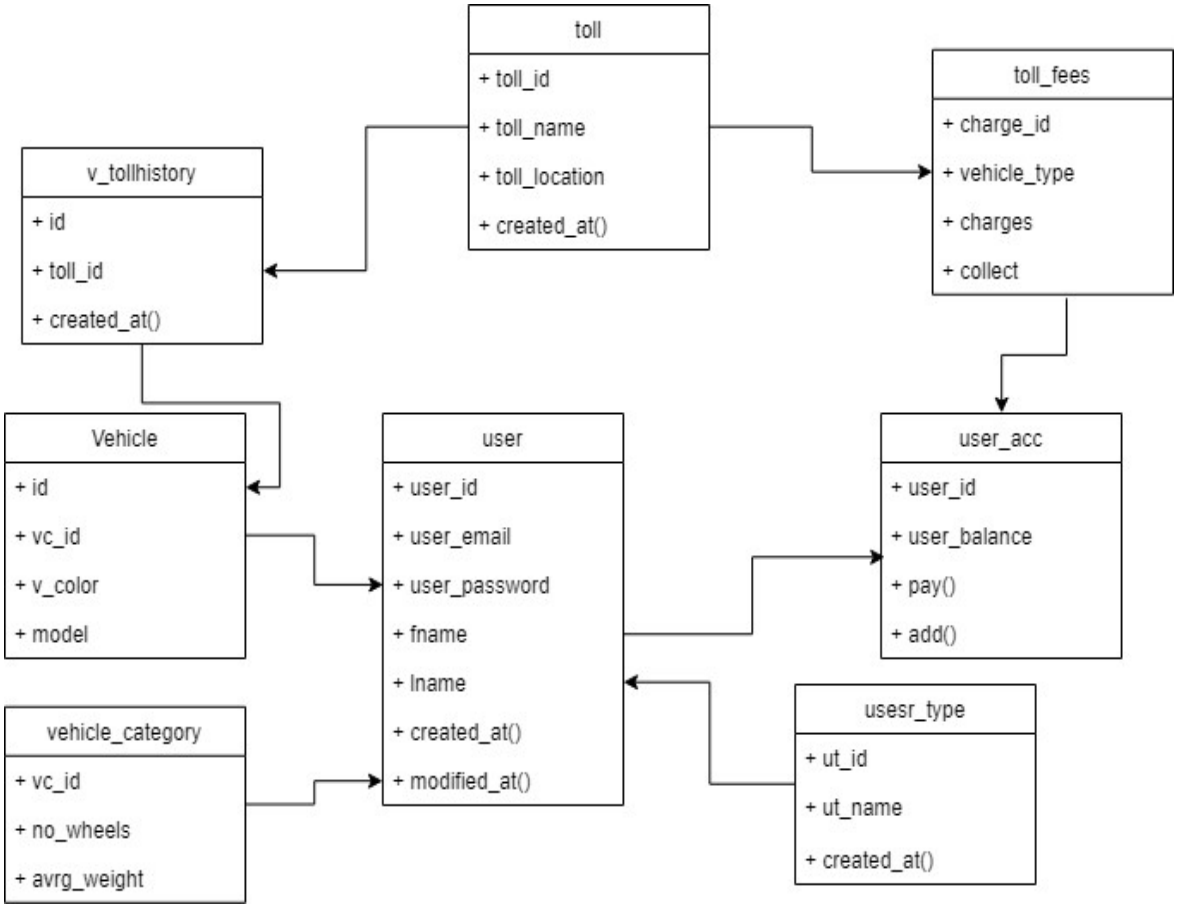
Activity Diagram

2.6.3 E-R Diagram



ER Diagram

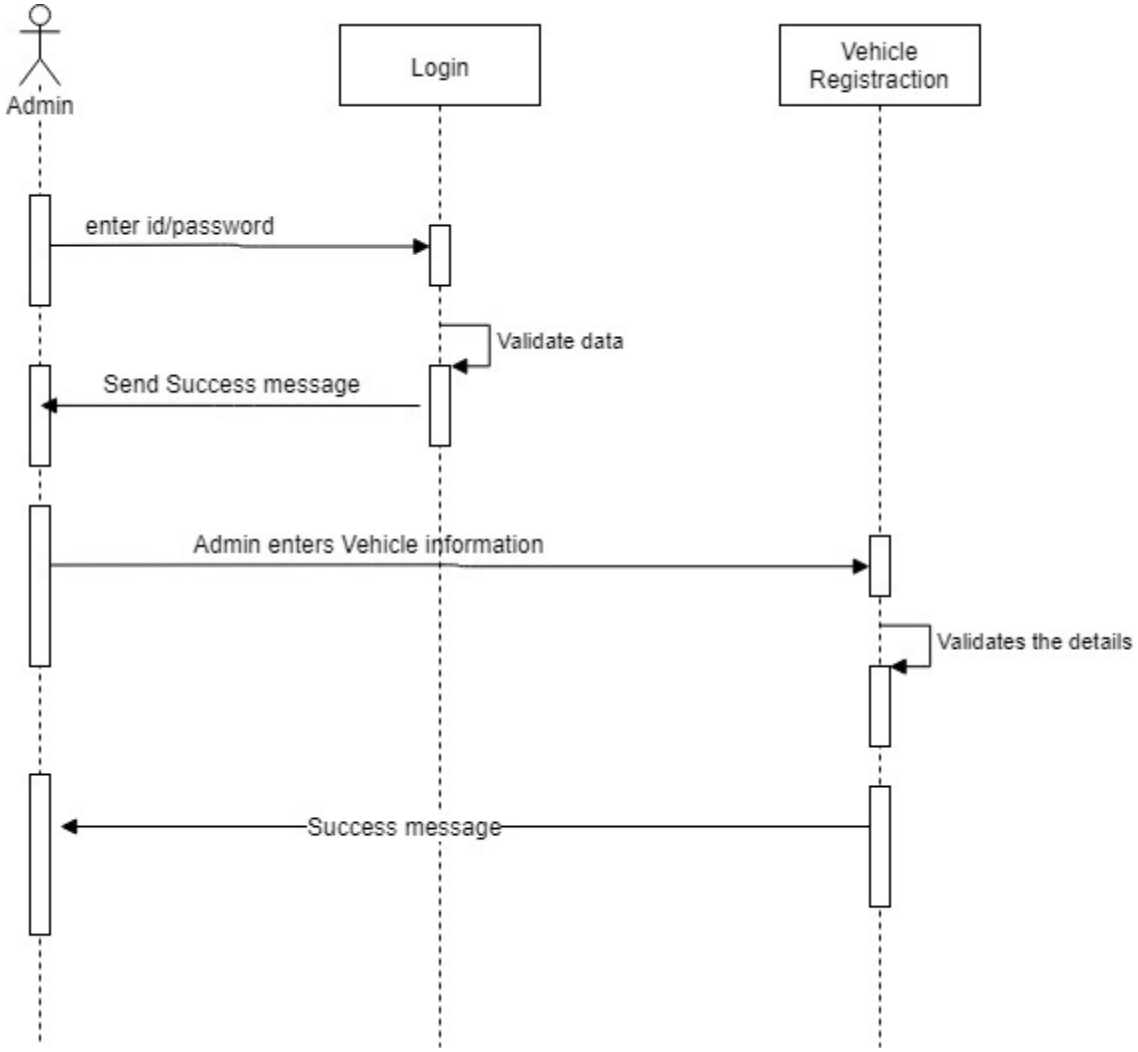
2.6.4 Class Diagram



Class Diagram

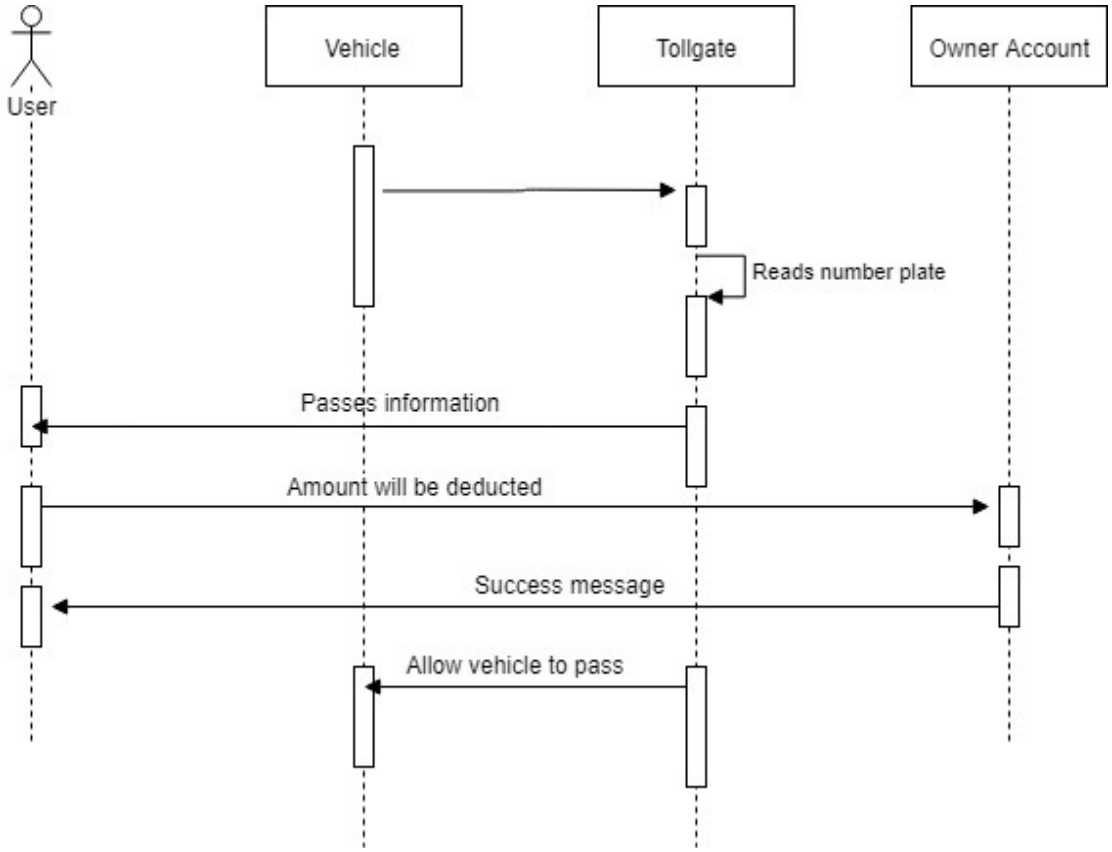
2.6.5 Sequence Diagram

Admin:



Admin Sequence Diagram

User:



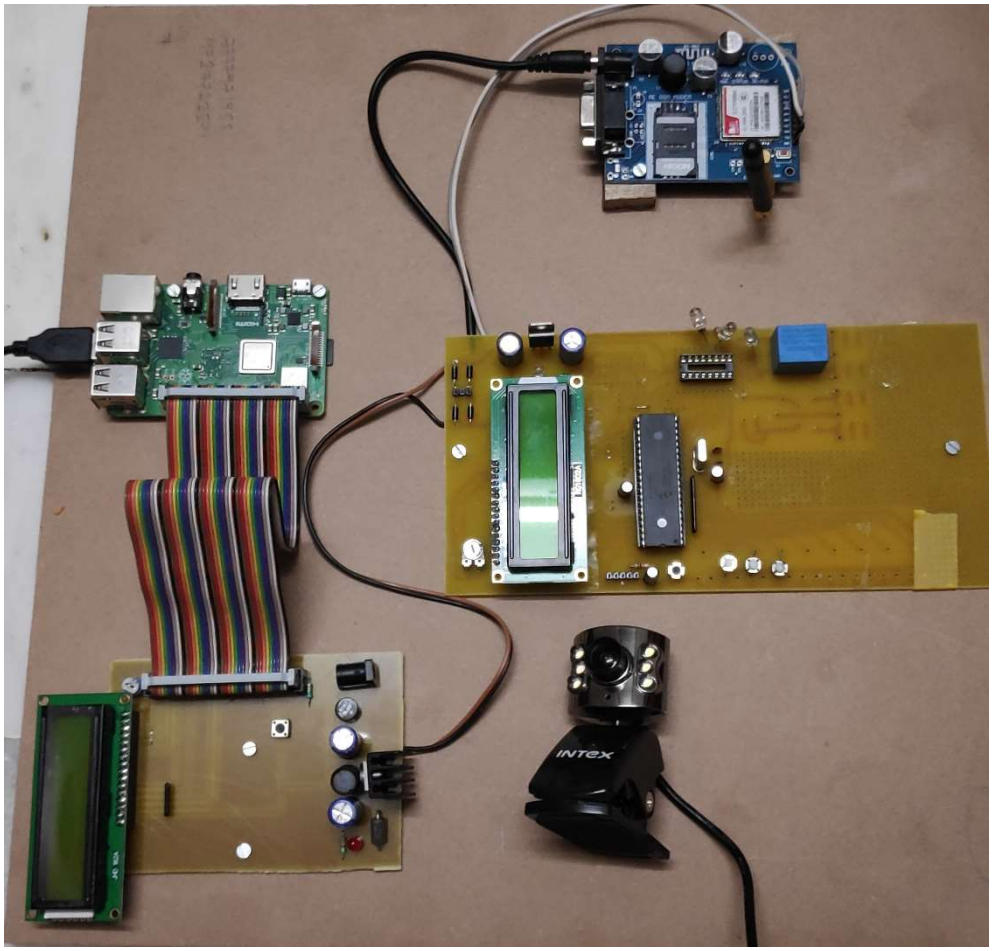
User Sequence Diagram

3. Implementation

3.1 Modules in the System

- Vehicle number detection by system
- Verify number with database
- Check for sufficient balance
- SMS verification to User

3.2 Screenshots



4. Summary

4.1 Advantages of the System

- Reduce traffic congestion
- Reliability
- Easy payment

4.2 Unique Features

- OCR

4.3 Conclusion and Scope of further Work

Automatic Toll Collection System using Optical Code Recognition and Encryption uses a simple sticker which is cheap as compared to RFID Tags and avoids the chances of forgery in case of ANPR systems. In addition, it provides security, faster processing, avoids traffic congestion, pollution and is reliable and an efficient toll collection system overcoming the drawbacks of the existing toll collection systems.

5 References

- Khadijah Kamarulazizi, Dr. Widad Ismail "Electronic Toll Collection System Using Passive RFID Technology" in Journal of Theoretical and Applied Information Technology.

- Pranoti Salunkhe, Poonam Malle, Kirti Datir, Jayshree Dukale, "Automated Toll Collection System Using RFID" in IOSR Volume 9, Issues 2(Jan-Feb 2013).

- Kumar Rajeev Ranjan, Abhinav Sinha "Automatic Vehicle Registration System for Tollbooths".

- Priyanka Chhoriya, Govinda Paliwal, Poonam Badhan "Image Processing Base Automation Toll Booth In Indian Condition", April 2013.

- Kresimir Romic, Irena Gailc, Alfonzo Baumgartner "Character Recognition Based On Region Pixel Concentration for License Plate Identification".