Automated Toll Management System

Submitted in partial fulfilment of the requirements of the degree of

Bachelor of Engineering

by

Divya Bhosle(08)

Sumit Jha(22)

Sai Narayan(45)

Nimritee Sirsalewala(63)

Guide:

Prof. Ranjana Singh

Department of Computer Engineering

Watumull Institute of Electronics Engineering and Computer Technology

2018– 2019

Declaration

I/we declare that this written submission represents my/our ideas in my/our own words and where others' ideas or words have been included, I/we have adequately cited and referenced the original sources. I/we also declare that I/we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my/our submission(s). I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Nimritee Sirsalewala (63):_____

Date: 09/04/2019

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to Prof Dr. Sunita Sharma, principal and Prof. Nilesh Mehta, H.O.D, Department of Computer Engineering of Watumull Institute of Electronics Engineering & Computer Technology for providing us an opportunity to do our project work on "Automated Toll Managemnet System".

We sincerely thank to our project guide Prof. Ranjana Singh, Department of Computer Engineering, Watumull Institute of Electronics Engineering & Computer Technology, Ulhasnagar for guidance and encouragement in carrying out this project work.

Date: 09/04/2019

Place: Ulhasnagar, Thane

Ms. Divya Bhosle (08)

Mr. Sumit Jha(22)

Mr. Sai Narayan(45)

Ms. Nimritee Sirsalewala(63)

ABSTRACT

Automated Toll Management System is designed to automatically allow vehicles to pass Toll Points without having to stop at all, eliminating the use of barricades that are used in all the Toll gates in India that result in a lot of jams, and wastage of fuel, not to mention time. Our proposed system keeps track of all the vehicles that pass through, recording the time, vehicle details etc. It also has a Report Generation Module that enables the Tolling Authorities to keep track of the number of transactions. This system also has some other features like notifying Enforcement Authorities about doubtful vehicle behaviour. Considering how important it is to ensure a smooth transition from a manned to fully automatic system, our system also provides required features for the same.

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1

Introduction

1.1 Problem Statement

This application targets to solve two biggest shortcomings of the existing Toll Management Systems:

i. Excess of fuel consumption caused due to vehicles having to stop for considerable amount of time waiting for change or due to slow manned operation at the tool booth.

ii. Since the process of generating tickets at the booth is manually done, it opens up windows that allow corporators to not account for all the vehicles that pass through the toll gate.

The problem with the existing TMSs is that in addition to creating jams, it also results in a lot of wastage of fuel. A joint study conducted by students at IIM Kolkata and The Transport Corporation of India reveals that a shocking 27000 crore rupees goes up in fumes dues to time wasted at a toll plaza. The situation is compounded with the same study group going on to state that an additional 60000 odd crore of fuel is wasted whilst slowing down and stopping at toll plazas. Our TMS seeks to do away with the massive waste of time and fuel by eliminating the time spent at the tollbooth.

Also, India's toll checkpoints are hotbeds of corruption. In Mumbai, Thane and Raigad alone, there is a corruption of Rs. 5000 crores at the toll nakas. This number can easily jump to 10000 crores, and we are not even talking about the whole country. This situation

has arisen because the TMSs in use right now have so many loop holes that allow for the booth operators to let vehicles pass without having to generate any receipt.

Our proposed system eliminates any chance of the same. Both the above mentioned drawbacks can be overcome by simply automating the process of toll ticket generation. Our proposed system removes the use of a man operating at the toll booth. With a combination of RFID Sensors and high speed cameras forming a highly efficient LPR System. The cameras use an OCR module to read the number plates on the vehicles that pass by a toll gate, and the RFID receivers access the data transmitted by the RFID tag inside the vehicle. The number plate in text form that is given as an output by the cameras is cross referenced with the number plate that is obtained from the RFID receivers. If the number plates obtained from both the modules match, it is considered a normal transaction. Else, it is safe to assume that the car is probable stolen, hence the police authorities are immediately notified, solving another issue of stolen vehicles that pass through toll gates.

1.2 Scope of the Project

Realizing the importance of resources like fuel, and time, and considering how other countries like the US manage their tolling systems, it is inevitable for TMSs in India to adapt to changing times. With the technology available to us, it is easy, necessary and definitely possible to upgrade our functioning TMSs.

The first goal is to aid the users with the registration process. This system we are talking about is a giant leap, since the deduction of money is automatic, every vehicle that passes through must be registered. We need to be prepared for all the vehicles that pass through the gates that are not registered. The system will flag all such vehicles, as the camera will capture the vehicle's number plate, but the RFID receivers will get nothing. Eventually, people will have to get themselves registered. When they do, and when they fill their wallets with money, all the money that previously couldn't be paid will be deducted. If a user fails to get himself registered for a long time, they will be sent a warning letter by details obtained from their license plate. This system too, is automatic. The users can get themselves registered through a centralized website, that notifies the authorities, who

either send the generated RFID tags to the user, or the user collects them from the toll plaza.

Talking about the hardware on site, a combination of RFID Receivers and high speed, infrared cameras form the ANPR. Both these setups mean we can set up a cross referencing system that ensures the RFID tag being used really belongs to the user.

As mentioned earlier, since this system is a huge leap from the existing System, the transition period is very crucial. Necessary measure must be taken to ensure speedy acquisition of RFID tags and dealing with defaulters. Once the initial period passes, its going to be a smooth ride ahead. Since the database will be centralized for all the Toll Gates, multiple registrations are not required.

Chapter 2

Review of literature

2.1 Electronic Toll collection

An ETC system is capable of electronically charging a toll to an established customer account. The system can determine whether a passing car is registered, automatically charging those vehicles, and alert the local highway patrol about users that are not registered. The ETC method allows vehicles to pass through a toll facility without the driver having to pay anyone the booth. Fig 2.1 represents Electronic Toll collection [1].

The Electronic Toll Collection (ETC) system is currently used throughout the world. Some countries that have the ETC system are Canada, Poland, Japan, Italy and Singapore, among many others. Some Electronic Toll Collection Systems in India, owned by NHAI (National Highways Authority of India) operated by different toll management systems are NH-6 toll road at Kharagpur operated by Toll Tax Toll Collection System, Delhi Gurgaon Expressway at Delhi operated by Metro Electronic Toll Collection Systems, Lucknow, Sitapur Expressways Ltd at Uttar Pradesh operated by Rajdeep - Toll Management System, Madhurai, Rameshwaram Express ways Ltd and Cochin in Tamil Nadu operated by Technovaa– Toll Management System, GMR Ambala-Chandigarh Expressway in Haryana & Punjab operated by Rajdeep - Toll Management System etc[2].

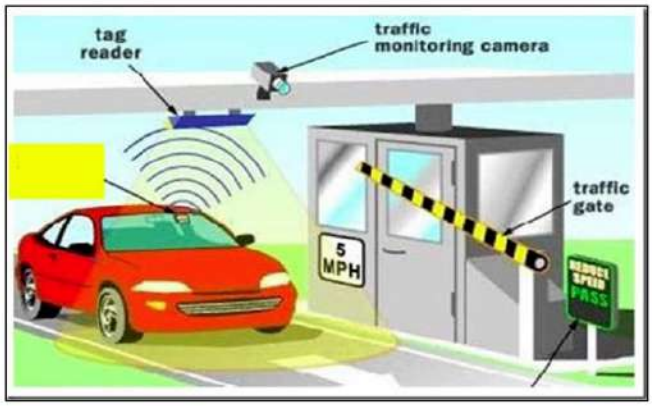

Fig 2.1: Electronic Toll Collection System

2.2 FASTag

FASTag is a simple to use, reloadable tag which enables automatic deduction of toll charges and lets you pass through the toll plaza without stopping for the cash transaction. FASTag is linked to a prepaid account from which the applicable toll amount is deducted. The tag employs Radio-frequency Identification (RFID) technology and is affixed on the vehicle's windscreen after the tag account is active.

FASTag is a perfect solution for a hassle free trip on national highways. FASTag is presently operational at 180 toll plazas across national and state highways [3]. More toll plazas will be brought under the FASTag program in the future. Fig 2.2 represents FASTag [4]. Benefits of FASTag include:

i. Saves Fuel and Time: FASTag is read by the tag reader at the plaza and the toll amount is deducted automatically, when the vehicle approaches the toll plaza. The vehicle with FASTag doesn't need to stop at the toll plaza for the cash transaction.

ii. SMS Alerts for Transactions: Customer will receive SMS alerts on his registered mobile numbers for all the transactions done in his tag account.

iii. Online Recharge: Customer may recharge his tag account online through, Credit Card/ Debit Card/ NEFT/ RTGS or Net Banking.

iv. Web Portals for Customers: Customers can access their statements by logging on the FASTag customer portal.

Fig 2.2: FASTag

2.3 Pay By Phone

It is for parking system and toll payment system. For toll payment by this app the user need to download this app. After this register all the details of the approaching user to get started on the service. One account works everywhere for tolling system. Enabling tolling by logging on to the website and editing 'Vehicle Details' in your profile. They can make auto payments, meaning worry-free driving. And they will send the SMS & email payment notifications, so they can keep of their usage. Vehicle has to wait near toll plaza for the detection of the bar code. That bar code reader which is on the toll scan the barcode and the individual has to pay the amount.

On, going through various systems and its implementation, we find that, ETC system is installed at selected toll plaza in India which is owned by NHAI while Pay by phone system is currently under development. Moreover, on every toll plaza at least one lane of FASTag is being installed [5].

Fig 2.3: Pay by Phone

2.4 RFID

RFID chip is capable of carrying 2,000 bytes of data or less. The RFID device serves the same purpose as a bar code or a magnetic strip on the back of a credit card or ATM card; it provides a unique identifier for that object, just as a bar code or magnetic strip must be scanned to get the information, the RFID device must be scanned to retrieve the identifying information. The advances in the technologies related to wireless communication has led to the emergence of several engineering designs to aid the human requirements. Today, on one side the importance for secured access is growing in several fields and on the other side with technology advancements the RFID cards and readers are becoming low cost. Both these aspects are the primary reasons for rapidly growing RFID based authentication system. Today, several wireless technologies are used for building wireless networks. Among them the 2.4GHz wireless network is most widely deployed and used. As the data collected by the RFID readers is accurate and does not take more than few seconds to read all the products, it reduces the amount of time taken to count the stock and then store them at the designated location [5].

Fig 2.4: Radio Frequency Identification

2.5 Salik (TMS used in Dubai)

Being stuck in a traffic jam is no fun, especially on a hot summer day. Tempers fray and blood pressure rises with the frustration of arriving late for work or missing important meetings or family occasions.

But if the strain of going nowhere fast is irritating for the individual driver, the consequences for a city with a gridlocked economy can be far more serious, as a report published this week by international transportation consultancy INRIX makes clear.

The Global Traffic Scorecard, the largest study of its kind, analyses congestion in 1,360 cities in 38 countries – and its publication in the very week that Abu Dhabi has announced plans to follow Dubai's example and introduce road tolls can be described as timely.

INRIX analyses real time data from the navigation systems of 300 million cars around the world, which reveals exactly where, when and how fast – or slow – they are travelling. And, in an age of global economic competition, in which productivity can be a make-or-break yardstick for a nation's hopes of attracting inward investment, it will be a matter of grave concern to American economic planners, for example, that 10of the world's 25 most gridlocked cities are in the US.

The worst city in the world for motorists is Los Angeles, where in 2017 drivers spent on average in excess of 100 hours sitting in traffic jams – more than four days of their lives.

By comparison, life for motorists in the UAE, where traffic in four cities was monitored, seems positively rosy. The most congested location, predictably, is Dubai. Here in 2017 drivers spent an average of 29 hours going nowhere at peak times, putting the city at a reasonably smooth-flowing 208th place in the global congestion table. A separate measurement that only looked at 200 major metropolises placed Dubai in 78th.

Driving in Abu Dhabi is even easier on the nerves. Motorists in the capital endured an average of only 13 hours in congestion in 2017, ranking the UAE capital in 727th place in the global congestion chart.

And, says Graham Cookson, chief economist and head of research at INRIX, comparing the percentage of time Abu Dhabi drivers spend in congestion during peak and off-peak times reveals an even less stressful picture. In chronically busy cities, such as Los Angeles and London, these figures vary wildly, but in Abu Dhabi there is almost no variation [6].

Chapter 3

Description

3.1 Components and their function

3.1.1 Customer Service Portal:

This is where the Vehicles will be registered, and customers can view their wallet balance, fill their wallet with money. The CSP Systems are set up in till plazas where RFID tags will be generated as soon as the registration process is completed.

3.1.2 Ticket Generation:

This component does not have high UI Requirements, it might as well run without any interface at all. The information received from the other modules is fed into this module after a vehicle passes through the toll gate. Data is then transferred to the database which aid in the report generation method, and then also used in the payment gateway.

3.1.3 Report Generation:

Since every toll both has to maintain records of the vehicles that pass through them for accounting, and security purposes, Report Generation is an important component. This module provides well compiled reports based on number of vehicles that pass through the toll gate every day, Classes of Vehicles. This is where the toll prices are set, and other stuff like Vehicle Class Management is done.

3.2 Overall system view

This section sets out the overall system architecture, data and process flows, logical data model and terminology which are then used in the requirements statements. Figure 3.1.a illustrates the overall system and Figure 3.2.b the main business processes within the tolling operation that the System must support.

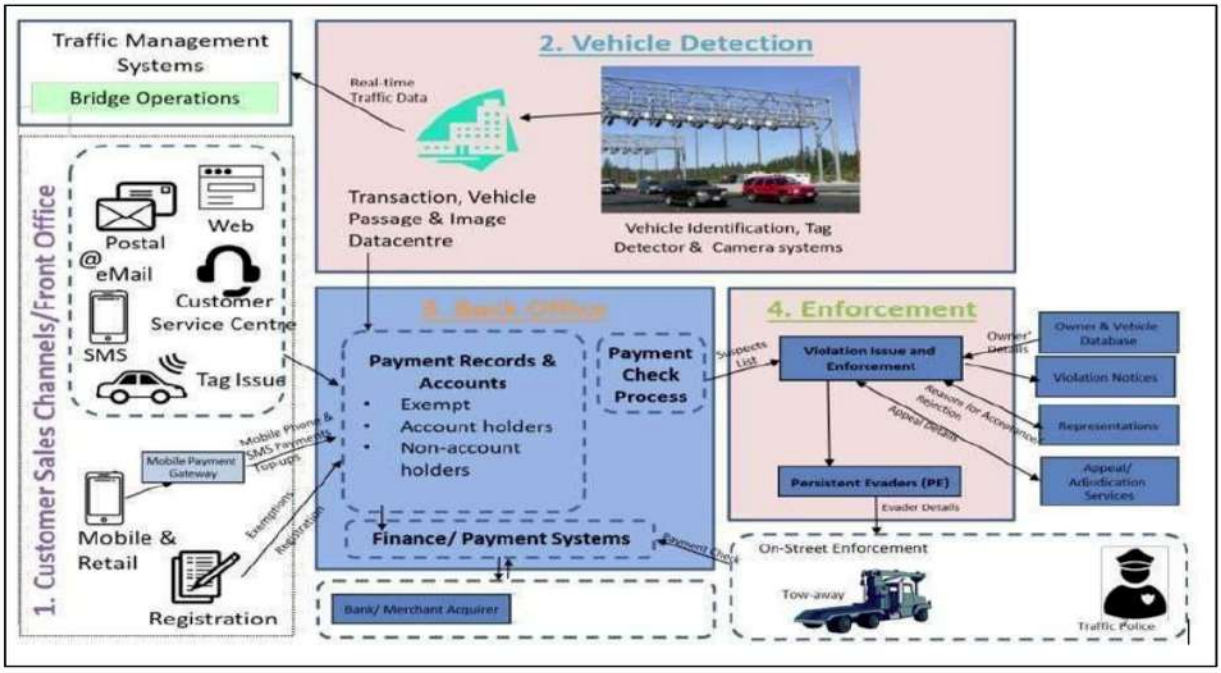

Fig 3.2.a: Overall Architecture

1. Customer Channels
- Contact channels (e.g. web, mobile, customer service centre)
- Payment channels (bank card, mobile, cash, billing)
- Account registration
- Marketing and communications

2. Vehicle Detection
- Create Vehicle Passage Records
- Tag transactions matched to images
- Vehicle classification (registration database)
- Manual check of license plates
- Manage detection data quality
- Store evidence and transactions

3. Back Office
- Reconcile vehicle transactions and payments/accounts
- Manage exemptions/ discounts
- Manage customer accounts
- Customer communications
- Manage customer data quality
- Verify vehicle detection data
- Refer non-payers to enforcement
- Account top-ups/ Billing of post-pay accounts
- Finance, accounting and reporting

4. Enforcement
- Validate violation data/evidence
- Vehicle owner details from database
- Issue violation notices
- Receive payments of violations/fine
- Progress violation escalating charges
- Manage enforcement data quality
- Manage representations
- Manage input to appeals and adjudications
- Identify persistent non-payers
- Initiate vehicle impounding action
- On-street enforcement support

Fig 3.2.b: Business Process

3.3 Interfaces

There are a number of key interfaces between the tolling system and the other third parties. These interfaces are illustrated in Figure 3.3 and described below.

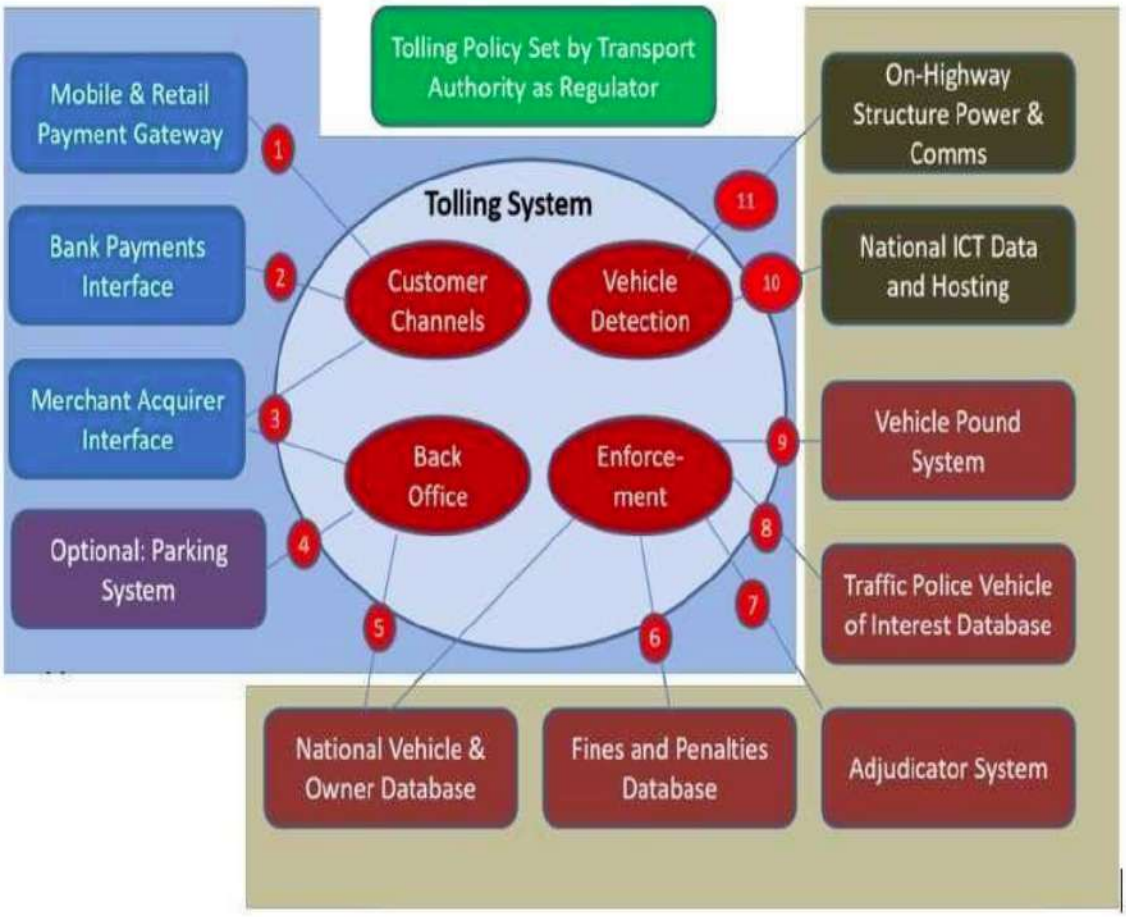

Fig 3.3 Key Interfaces

3.4 Customer Channels

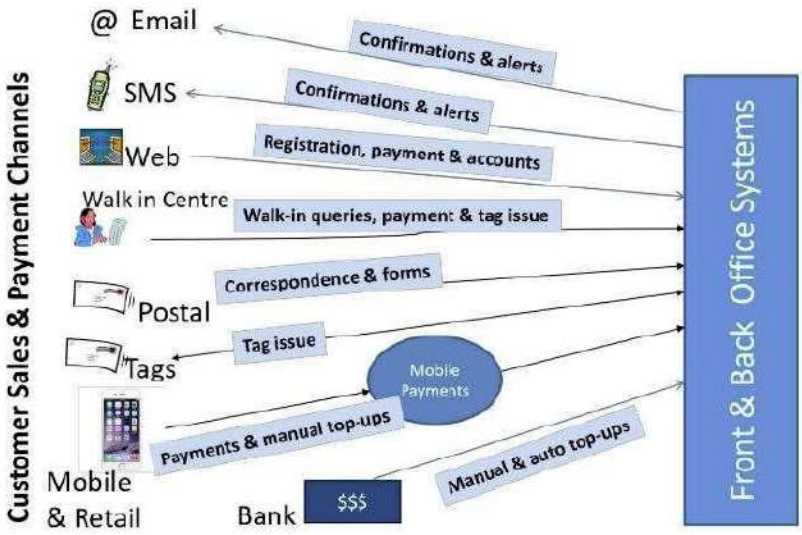

Fig 3.4: Customer Payment Channels

There are two types of payment means:

i. Through an account for which a User registers and can then make manual or automated account top-ups

ii. Simple payment of the toll amount due.

All types of payment channel can be used to either pay a toll charge or top up an account.

The supplier is to provide the Systems that enable the following channels and channel components:

i. Email and SMS for outgoing confirmations and alerts to the user.

ii. Website for registration, payments, account management and information.

iii. API for mobile payments from each of two mobile operators.

iv. A mobile app for customer account management.

v. Banking payments interface with Bank of the Maldives.

vi. Incoming post and correspondence management system including scanning, document storage, imaging and distribution.

vii. Client customer service agent workstation and enquiry functions.

viii. Customer service walk in centre functions so that customers who call in can have their queries resolved by client agent staff and be issued with tags

ix. Outgoing post functions to customers with communications, tags, statements etc.

3.5 Vehicle Detection Data Quality

The Toll System Architecture assumes:

i. Licence Plate Recognition (LPR) for all vehicles generating a 'Vehicle Passage Record'

ii. Learning capability in the LPR cameras supported by access to the Transport Authority's Vehicle and Owner database of licence plates, and therefore syntax and formats, to aid validation

iii. Use of RFID tags on most, if not all vehicles, to aid identification, and which are to be matched with licence plate 'reads'

iv. Use of double gantries so that vehicles can be photographed front and back. It is acknowledged that solutions exist using a single gantry, though they may not perform well given motorcycle and scooter 'lane discipline' in the region

v. Manual back office checking of images to validate that VPN 'reads' are correct.

3.6 Back Office

The Back Office will include performing the processes of:

i. Validating VPRs

ii. Matching payments with VPRs

iii. Settlement of account

iv. Potential Violator Identification

v. Potential Violator - Owner Details

vi. Financial Reconciliation

vii. Financial Reporting

viii. Fraud detection

ix. Management Information

x. Creating the Data Warehouse

xi. Reporting.

Figure 3.6 provides five use cases for the settlement of tolls, to illustrate the intended process.

Fig 3.6: Use cases for Toll Settlement

3.7 Enforcement

An enforcement System must be in place that supports:

i. Matching the violation evidence with vehicle details from the vehicle database

ii. Manual inspection of a violation notice to visually confirm the violation details

iii. Issue of the violation notices to the registered vehicle owner

iv. Managing the status of outstanding violation amounts and debts by vehicle owner

v. Receiving violation payments through the web portal

vi. Maintaining the data quality of enforcement data

The following figures mention the use cases for Enforcement:

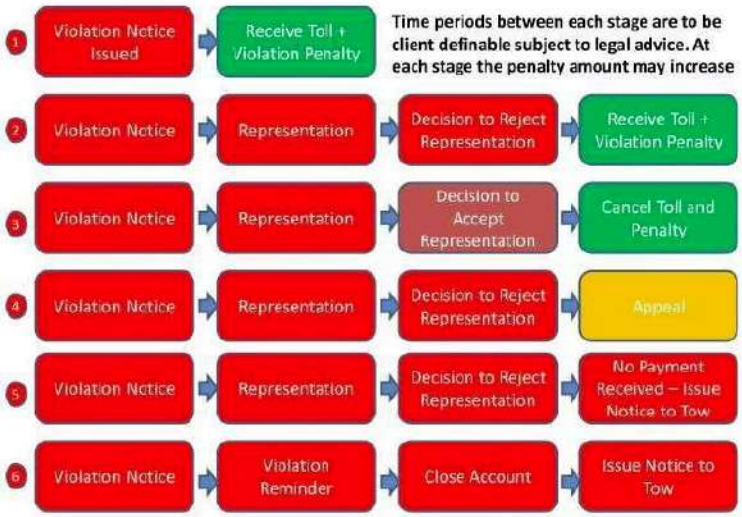

Fig 3.7: Use cases for Enforcement

3.8 Optical Character Recognition & RFID

The Optical Character Recognition technology in this system will be used to implement License Plate Recognition. The OCR extracts text from the recognised number plates and it matches the License Plate Number hence extracted is matched with the License Plate Number obtained from scanning the RFID tag of the vehicle. When these Numbers are a match, it is labelled as a normal transaction, otherwise the Enforcement Authorities are notified. The cameras placed at the gantries capture the license plate of the vehicle, and this data can be used to take further actions.

Fig 3.8: Optical Character Recognition

Chapter 4

Implementation

4.1 Algorithms Used

4.1.1 Automated Number Plate Recognition

ANPR can be used to store the images captured by the cameras as well as the text from the license plate, with some configurable to store a photograph of the driver. Systems commonly use infrared lighting to allow the camera to take the picture at any time of the day. ANPR technology tends to be region specific, owing to plate variation from place to place. ANPR uses optical character recognition (OCR) on images taken by cameras. Some license plate arrangements use variations in font sizes and positioning ANPR systems must be able to cope with such differences in order to be truly effective. More complicated systems can cope with international variants, though many programs are individually tailored to each country. The cameras used can include existing roadrule enforcement or closed-circuit television cameras, as well as mobile units, which are usually attached to vehicles. Some systems use infrared cameras to take a clearer image of the plates [7].

There are seven primary algorithms that the software requires for identifying a license plate:

i. Plate Localization[8]: responsible for finding and isolating the plate on the picture.

ii. Plate orientation and sizing - adjusts the dimensions to the required size.

iii. Normalization – adjusts the brightness and contrast of the image

iv. Character segmentation – finds the individual characters on the plates.

v. Optical character recognition

vi. Syntactical/Geometrical analysis – check characters and positions against country specific rules.

vii. The averaging of the recognised value over multiple fields/images to produce a more reliable or confident result. Especially since any single image may contain a reflected light flare, be partially obscured or other temporary effect

The complexity of each of these subsections of the program determines the accuracy of the system. During the third phase (normalization), some systems use edge detection techniques to increase the picture difference between the letters and the plate backing. A median filter may also be used to reduce the visual noise on the image.

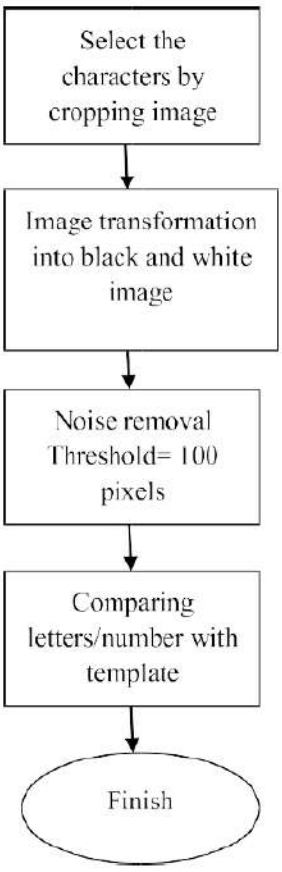

Fig 4.1.1: Automatic Number Plate Recognition

4.1.2 Number Plate Segmentation

Segmentation is the process by which we separate each number or letter in the number plate so as to process them further one by one. There are different methods that can be used for the achieving the segmentation process. Create bounding boxes such that the areas containing connected objects are separated, since a letter or a number would represent a connected object, a bounding box is created around each object. To prevent small or too large connected objects to be part of a bounding box a condition regarding the minimum size and the maximum size of a bounding box are defined. These conditions were found by trial and error to select the optimum values so that only the numbers and letters are selected and nothing else[9].

4.2 Hardware Requirements

4.2.1 Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform.

Technical Specifications:

i. Microcontroller: ATmega328P
ii. Operating Voltage: 5V
iii. Input Voltage (Recommended): 7 – 12V
iv. Input Voltage (Limit): 6 – 20V

v. Digital I/O Pins: 14

vi. Flash Memory: 32 KB of which 0.5 KB is used by bootloader

vii. SRAM: 2 KB

viii. EEPROM: 1 KB

ix. Clock Speed: 16 MHz

x. LED_BUILTI: 13

xi. Length: 68.6 mm

xii. Width: 53.4 mm

xiii. Weight: 25g

Fig 4.2.1: Arduino Uno

4.2.2 Male to Female Connecting Wires

Fig 4.2.2: Male to Female Connecting Wire

4.2.3 High Speed IR Night Vision License Plate Recognition Camera

(For Practical Purposes on Site)

i. High Speed IR Night Vision License plate Recognition Camera (For Practical Purposes on Site)
ii. Full high definition 2MP (1920 x 1080) resolution
iii. Back Light compensation (BLC) Supported
iv. Inbuilt Motion Detection Supported
v. High Light compensation (HLC) Supported
vi. Automatic White Balance
vii. I P 66 Weather proof outdoor enclosure

Technical Specifications:

i. Image Sensor : 1/3" Progressive Scan Sensor
ii. Signal System : PAL / NTSC
iii. Effective Pixels : 1920(H) x 1080(V)
iv. Frequency : 60 Hz / 50 Hz
v. Gamma Correction : 0.45
vi. Gain Control : Auto / Manual
vii. White Balance : Auto / Manual
viii. IR Cut : High Performance Mechanical IR Cut Filter
ix. IR Source : Inbuilt 2 X 4th Generation White Light Source Chip Array
x. IP Range : Up to 30 meters
xi. IR Life : 40000 Operation Hours

Since it wasn't possible to arrange for a High speed IR night vision camera, we supplemented it by making an android application that performs the same tasks that a camera with an integrated license plate detection software would perform.

4.3 Software Requirements

4.3.1 Java 8:

Java is a general-purpose programming language that is class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to "bytecode" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2018, Java was according to GitHub one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Class path (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest version is Java SE 12, released in March 2019. Since Java 9 is no longer supported, Oracle advises its users to "immediately transition" to Java 12. Oracle released the last public update for the legacy Java 8 LTS, which is free for commercial use, in January 2019. Java 8 will be supported with public updates for personal use up to at least December 2020. Oracle and others "highly recommend that you uninstall older versions of Java" because of serious risks due to unresolved security issues. Oracle extended support for Java 6 ended in December 2018.

4.3.2 NetBeans:

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Centre module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.
The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

i. User interface management (e.g. menus and toolbars)
ii. User settings management
iii. Storage management (carries out efficient storage)
iv. Window management
v. Wizard framework (supports step-by-step dialogs)
vi. NetBeans Visual Library
vii. Integrated development tools

4.3.3 Android Studio 3.1.4:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

The following features are provided in the current stable version:

i. Gradle-based build support
ii. Android-specific refactoring and quick fixes
iii. Lint tools to catch performance, usability, version compatibility and other problems
iv. ProGuard integration and app-signing capabilities
v. Template-based wizards to create common Android designs and components
vi. A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
vii. Support for building Android Wear apps
viii. Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
ix. Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9

features. While IntelliJ that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

4.3.4 Google Vision API:

Google Cloud's Vision API offers powerful pre-trained machine learning models through REST and RPC APIs. Assign labels to images and quickly classify them into millions of predefined categories. Detect objects and faces, read printed and handwritten text, and build valuable metadata into your image catalog.

Vision API can detect and extract multiple objects, and provide information about each object including its position within the image.

With Vision API's vision product search, retailers can create an engaging mobile experience that enables your customers to upload a photo of an item and immediately see a list of similar items for purchase from you.

Vision API uses OCR to detect text within images in more than 50 languages and various file types. It's also part of Document Understanding AI, which lets you process millions of documents quickly and automate business workflows.

4.3.5 Arduino IDE:

The Arduino integrated development environment (IDE) is a cross platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development

4.4 Methodologies

The proposed system can be further explained in terms of implementation with the help of flowcharts. The flowcharts provide in-detail explanation of how the User will interact with the system.

4.4.1 Vehicle and User Registration

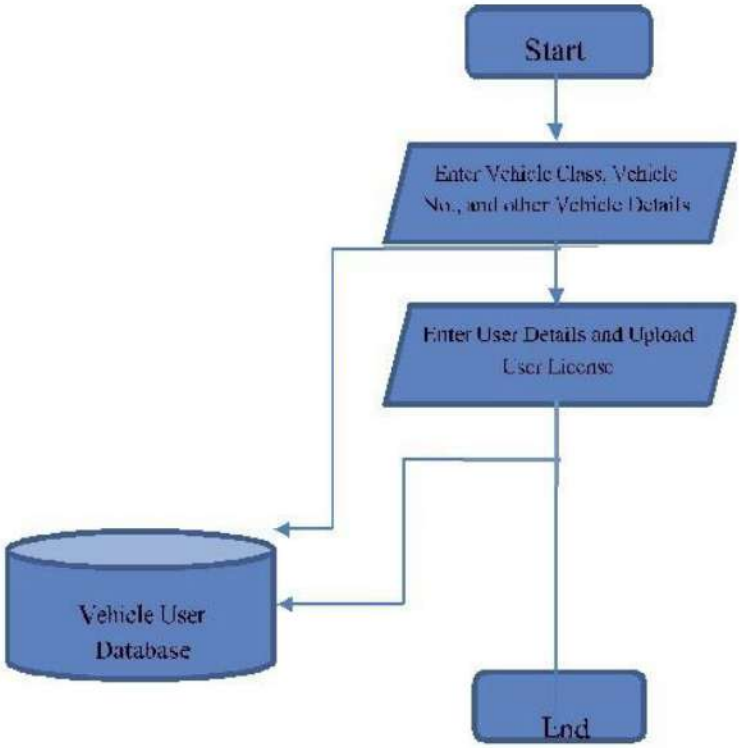

Fig 4.4.1: Vehicle User Registration Process

4.3.1 Lane Management

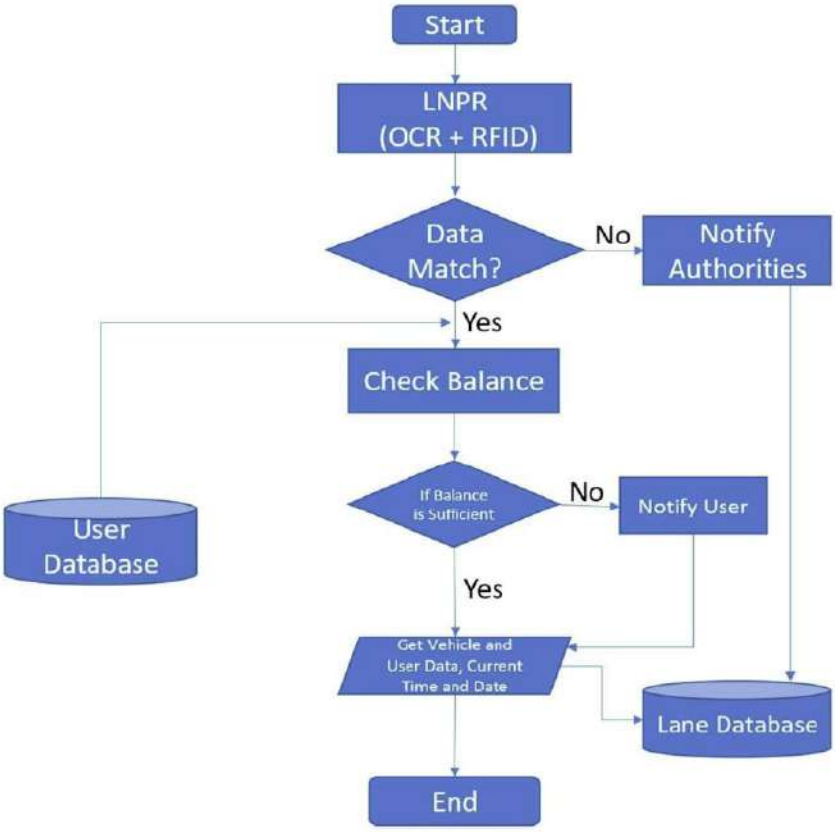

Fig 4.4.2: Lane Management Flowchart

Chapter 5

Testing

5.1 Testing Methods

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

i. Unit testing
ii. Integration testing
iii. System testing
iv. Acceptance testing

Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. These include:

i. Performance testing
ii. Security testing
iii. Usability testing
iv. Compatibility testing

The key to releasing high quality software that can be easily adopted by your end users is to build a robust testing framework that implements both functional and non-functional software testing methodologies.

5.2 Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs. This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users for beta testing along with those of your QA team. Beta testing is key to getting real feedback from potential customers and can address any final usability concerns.

The following images shows the Acceptance Testing carried out for the project.

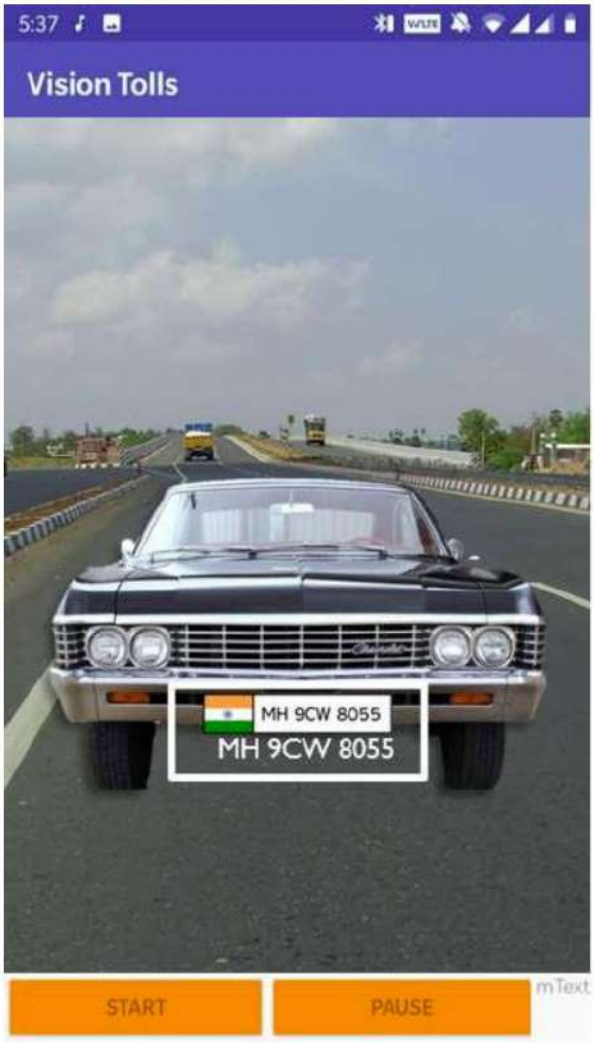

Fig 5.2.a: Vision Testing Image1

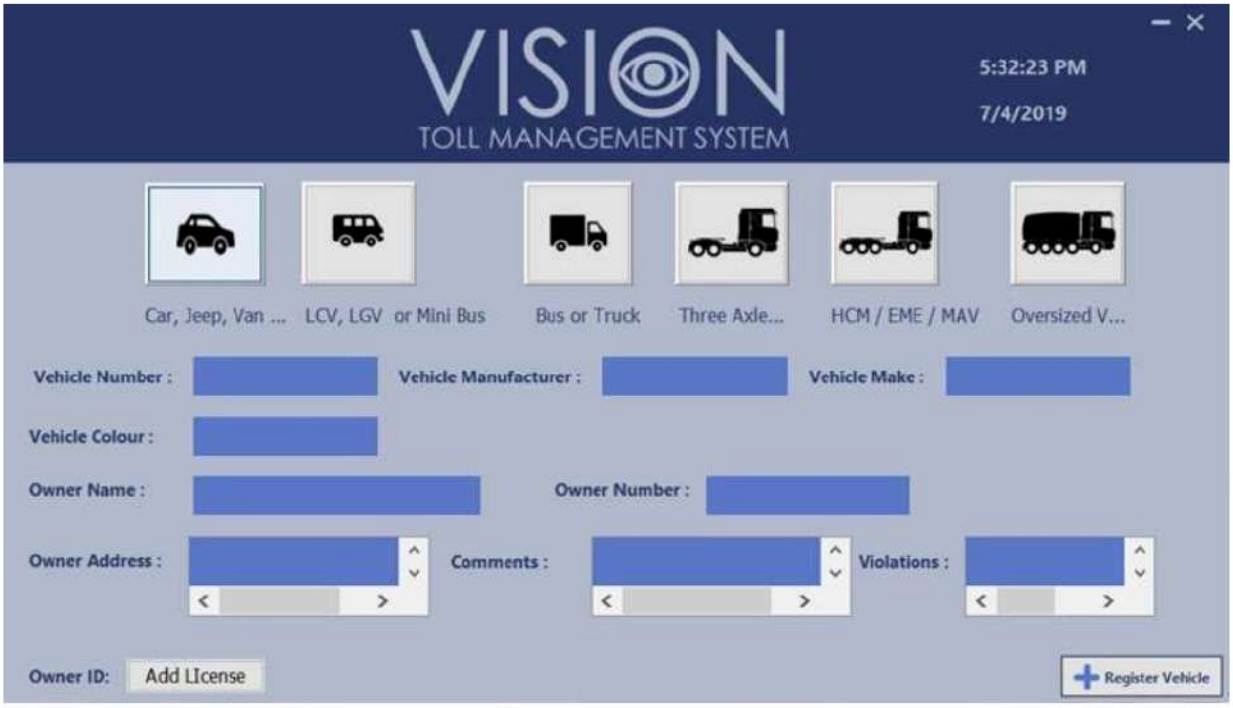

Fig 5.2.b: Vision Testing Image2

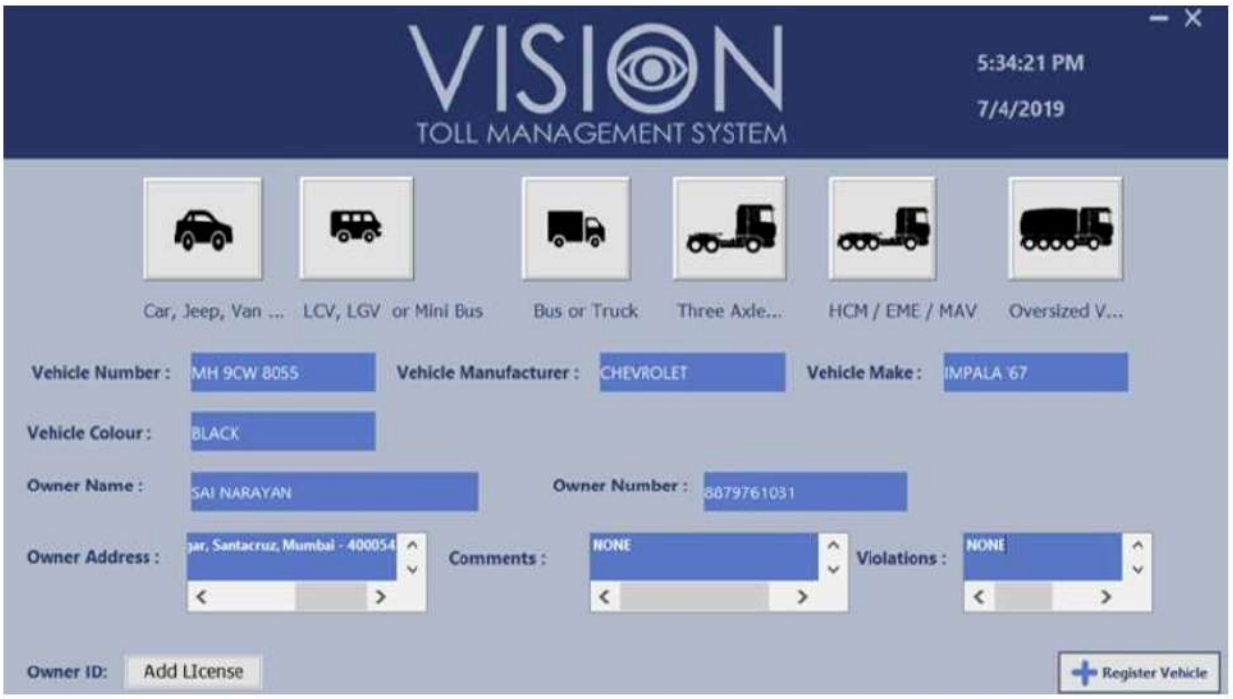

Fig 5.2.c: Vision Testing Image3

Fig 5.2.d: Vision Testing Image4

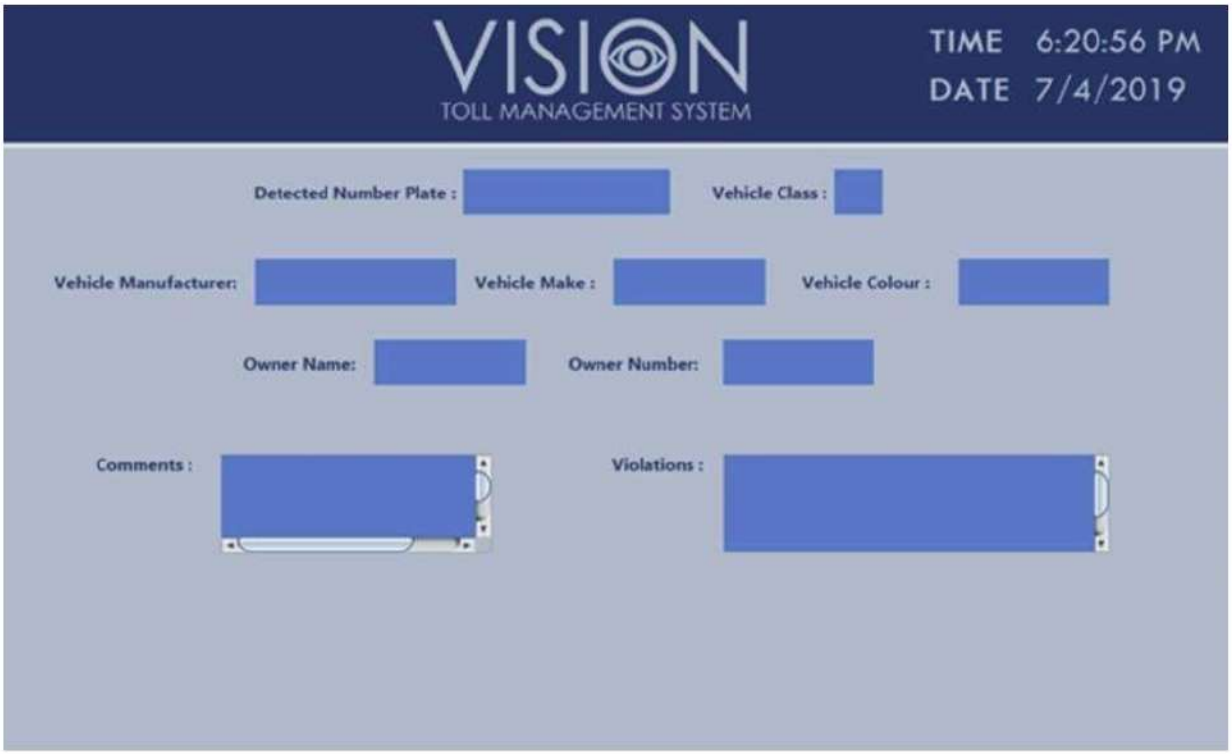

Fig 5.2.e: Vision Testing Image5

Chapter 6

Discussion

6.1 Conventional System

At present the conventional toll plazas are working manually. This method of toll collection is time consuming. The conventional way of collecting the toll from the vehicle owners or the drivers is to stop the car at the Toll Plaza and then pay the amount to the toll collector by the side of the toll booth, after which the gate is opened either mechanically or electronically for the driver to get through the toll station.

Another issue is that one needs to handle cash and even wait for getting the change. After paying the toll tax, a receipt is provided which one needs to preserve when the ticket purchased is a two-way ticket. Possibility that one may escape the toll plaza without paying is also there.

Suppose the manual toll collection system is very efficient, and time taken by one vehicle to stop and pay taxes is 50 seconds. Now if 200 vehicles cross the toll plaza, then time taken by 1 vehicle with 50 seconds average stop time in a month is: 50x30= 1500 seconds Yearly total time taken = 1500x12 = 18000seconds = 5.0 hours On an average each vehicle that passes through the toll plaza has to wait 5.0 hours, keeping their engines turned on. This figure is staggering, as if on an average we take 200 vehicles pass through the toll plaza each day, then yearly 72000 vehicles pass through the toll plaza, so each year 72000 vehicles waits for 5.0 hours keeping their engines on and thereby aiding pollution and wasting fuel and money.

6.2 RFID

RFID chip is capable of carrying 2,000 bytes of data or less. The RFID device serves the same purpose as a bar code or a magnetic strip on the back of a credit card or ATM card; it provides a unique identifier for that object, just as a bar code or magnetic strip must be scanned to get the information, the RFID device must be scanned to retrieve the identifying information. The advances in the technologies related to wireless communication has led to the emergence of several engineering designs to aid the human requirements. Today, on one side the importance for secured access is growing in several fields and on the other side with technology advancements the RFID cards and readers are becoming low cost. Both these aspects are the primary reasons for rapidly growing RFID based authentication system. Today, several wireless technologies are used for building wireless networks. Among them the 2.4GHz wireless network is most widely deployed and used. As the data collected by the RFID readers is accurate and does not take more than few seconds to read all the products, it reduces the amount of time taken to count the stock and then store them at the designated location[9].

Advantages:

i. Saves Fuel and Time: RFID Tag is read by the tag reader at the plaza and the toll amount is deducted automatically, when the vehicle approaches the toll plaza. The vehicle with Tag doesn't need to stop at the toll plaza for the cash transaction.

ii. SMS alerts for transactions: Customer will receive SMS alerts on his registered mobile numbers for all the transactions done in his tag account.

iii. Online recharge: Customer may recharge his tag account online through, Credit Card/

iv. Debit Card/ NEFT/RTGS or Net Banking.

v. No need to carry cash: Customer doesn't need to worry about carrying cash for the toll payments.

vi. Web portal for customers: Customers can access their statements by logging on the Tag customer portal.

Chapter 7

Conclusion

Realizing the importance of resources like fuel, time and money and considering how other countries manage their tolling systems, it is inevitable for TMSs in India to adapt to changing times. With the technology available to us, it is easy, necessary and definitely possible to upgrade our functioning TMSs.

This system is a giant leap since the deduction of money is automatic ,every vehicle that passes through must be registered and if not the system will flag all such vehicles and they will have to pay later on. If a user fails to get himself registered for a long time, they will be sent a warning letter by details registered through a centralized website, that notifies the authorities who either send the generated RFID tags to the user or the user collects them from the toll plaza. As mentioned earlier, since this system is a huge leap from the existing System, the transition period is very crucial. Necessary measure must be taken to ensure speedy acquisition of RFID tags and dealing with defaulters. Once the initial period passes, its going to be a smooth ride ahead. Since the database will be centralized for all the Toll Gates, multiple registrations are not required.

Automated Toll Management System will help saving user's time, money and fuel. Using this system corruption can be eradicated. We would have an automated system using OCR technology, RFID scanners and high-speed cameras. The major drawback of the conventional system will be removed by eliminating the use of barricades. Man power required would be next to negligible. Using this system, the government can notify the police about robbed car. Also, number of errors would be reduced.

Chapter 8

References

[1] https://images.app.goo.gl/nLSbk94Weh7Sr2AA8

[2] https://mobility.tamu.edu/mip/strategies-pdfs/traffic-management/technical-summary/Electronic-Toll-Systems-4-Pg.pdf

[3] http://www.fastag.org/

[4] https://images.app.goo.gl/scCGYyCPFamVBKFQA

[5] http://www.ijresm.com/vol1%2Ciss4%2CApril18/IJRESM14_23.pdf

[6] https://www.thenational.ae/uae/transport/salik-road-tolls-and-how-uae-motorists-fare-in-global-traffic-jam-rankings-1.704355

[7] https://www.researchgate.net/publication/263595027_Automatic_number_plate segmentation_and_recognition

[8] https://ieeexplore.ieee.org/document/1688109: A License Plate-Recognition Algorithm for Intelligent Transportation System Applications

[9] KhadijahKamarulazizi and DR. Wadad Ismail, "Electronic toll collection system using passive RFID technology", School of Electrical and Electronic, University Science Malaaysia, Journal of Theoretical and Applied Information Technology, 2005-2010.