A PROJECT REPORT ON

Automatic Number Plate Recognition System

Submitted By

Mst. Prathamesh Madhusudan Mhapsekar (FS13IF049)

Mst. Mandar Madhusudan Mhapsekar (FS13IF050)

Mst. Omkar Pradeep Dake (FW13IF001)

Under the Guidance of

Prof. Pooja L. Chelani

In Partial fulfilment of the requirement for the Award of Diploma

Information Technology Department

Government Polytechnic, Mumbai

Mumbai 400051

2015-2016

Certificate

This is to certify that the project entitled **'Automatic Number Plate Recognition System'** being submitted by **'Prathmesh Mhapsekar (FS13IF049), Mandar Mhapsekar (FS13IF050) and Omkar Dake (FW13IF001)'** to the department of Information Technology, Government Polytechnic, Mumbai for the Award of the Diploma in the academic year 2015-16.

Head of Department

(Dr.R.A.Patil)

Principal

(Dr.H.P.Taskar)

Project Guide

(Prof.P.L.Chelani)

External Examiner

(   )

Acknowledgement

We owe our gratitude to many people who have supported us throughout this journey

We would,firstly like to express our heartfelt gratitude towards our respected Principal **Dr. H. P. Taskar** and our Head of Department **Dr. R. A. Patil** for providing us immense facilities, guidance and never ending support.

The completion of any inter-disciplinary project depends upon cooperation and combined efforts of several sources of knowledge. We take this opportunity to express our profound gratitude and deep regards to our guide **Mrs. P. L. Chelani** for her exemplary guidance , monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her, time to time shall carry us long way in the journey of life on which we are about to embark.

We are highly obliged to **Mrs. S. A. H Shaikh, Mrs. U. C. Khake** and all the faculties at Government Polytechnic, Mumbai for the valuable information provided by them in their respective fields. we are grateful for their cooperation during the tenure of our project,which also paved the way for the completion of our task on time

Our Special thanks to **Mr. K. G. Karulkar, Mr. S. G. Jadhav, Mr. A. K. Hemade, Mr. P. B. Yadav, Mr. J. N. Sargar** and all the supporting staff who never left a single stone unturned for providing us with all possible resources, when we were in dire need of them.

Lastly, we thank almighty, our parents, family, friends and well wishers who always looked for the chance to help us in whatever means came forth and for their constant encouragement without which the project would not be a distant reality.

Signature of Student

(Mst. Prathamesh Mhapsekar)

(Mst. Mandar Mhapsekar)

(Mst. Omkar Dake)

Preface

We take great opportunity to represent this project report on Automatic Number Plate Recognition System put before readers some useful information regarding our project. We have made sincere attempts and have tried to present this matter in precise and compact form, the language being as simple as possible. We are sure that the information contained this volume would certainly prove useful for better insight in the scope and the dimension of this project in this true prospective

The perseverance and deep involvement shown by our group members made this vague task look very interesting and simple. We have designed this software project completely from scratch and we have not incorporated any ready made material from the Internet or any other sources to make our project seemingly more attractive and meaningful.

Abstract

Automatic number plate recognition has became very important in our daily life because of the unlimited increase of cars and transportation systems which make it impossible to be fully managed and monitored by humans, examples are so many like traffic monitoring, tracking stolen cars, managing parking toll, red-light violation enforcement, border and customs checkpoints. Yet its a very challenging problem, due to the diversity of plate formats, different scales, rotations and non-uniform illumination conditions during image acquisition. Automatic number plate recognition system is used for the effective control of these vehicles.

”Automatic number plate recognition system is an image processing technology that identifies vehicles by processing their number plate without direct human intervention.” ANPR is used by police forces around the world for law enforcement purposes. It is also used for electronic toll collection on pay-per-use roads basis and it is also used in police patrolling. ANPR also helps law enforcement in tracking stolen vehicles which is one of the biggest traffic crime in the metropolitan cities Thus reducing the work of police enforcement to a great extent. In our project we have implemented ANPR technology in toll Area were we would be managing the toll system as well stolen car tracking.

Contents

List of Figures

Introduction

In current days vehicles play important role in transportation and the use of vehicles is also increasing due to population growth and human needs. Automatic number plate recognition system is used for the effective control of these vehicles. Automatic number plate recognition system is an image processing technology that identifies vehicles by processing their number plate without direct human intervention. Thus reducing the work of police enforcement to a great extent. It consists of a camera that has the capability to capture an image, finds the location of the number plate in the image and then extracts the characters using character recognition tool that translate the pixels into alphanumerically readable character or string. Then the obtained information is used in different applications. ANPR can be used in many areas from speed enforcement and toll collection to management of parking lots etc. At present, in ANPR there are several techniques used for the recognition of number plate such as pattern matching, neural network character recognition and other image processing technology.

For Eg : As vehicle enters the input gate, number plate of the vehicle and the type of the vehicle is automatically recognized using image processing. This information is stored in the database and various applications are performed. So assuming if a car is registered as stolen by its owner and the same car appears at the toll or a signal than by using this technology (ANPR) the car would be recognized as stolen. Thus alarming the police in-charge regarding the stolen car

Applications of ANPR:

1. Automatic toll system.
2. Border Crossings.
3. Tracking of stolen vehicles.
4. Automatic parking lot payment.
5. Over-speeding Check
6. Smart traffic detection
7. Police Patrolling
8. Marathons

Flow of Working

- Camera is used to capture vehicle size and also the number plate of that vehicle.
- Camera will capture the number plate of the vehicle, it will send it to the PC.
- MATLAB software is used for the image processing. MATLAB code is used to recognize the number plate and also check whether the vehicle is stolen or not.
- We will create a database of the stolen vehicle. If current captured number plate matches with stolen number plate in the database so it will conclude that the vehicle is stolen and the gate mechanism will never be opened in that case and buzzer will turn on.
- To control the gate mechanism we are using the DC motor which is connected to the Uc through motor driver IC.
- If vehicle is not stolen type then it will checked whether the car is a registered car or not.
- If the car is registered than it will go from the automatic gate were the toll amount would be deducted automatically online and if the car is not registered than it will go through the manual gate were the payment would be done through cash.
- There will also be a another camera which will detect the size of the car through some MATLAB Algorithm.
- Toll charges depends on the size of the vehicle that we will be detected by the MATLAB code, according to that PC will send signal to the c using USB to TTL.
- LCD is then used to display the toll charges which is depended on the size of the vehicle and also determines the amount that would be deducted from the database entry of that vehicle

Existing System

Toll Collection in India is a matter of concern as the toll checkpoints are hotbeds of corruption. The current scenario is that the vehicle needs to slow down at the toll collection area. Then the person at the Toll will note down the car details like the car number plate, car size and many other details and then the car owner will pay the amount. This increases the traffic near the toll area. Sometimes the toll collector doesnt collect the toll fee from the owner or takes bribe. Currently in Toll booths, parking lot, Law and order maintenance we need a specific person to monitor all this activities This activity requires constant attention by the person and sometimes may be error prone and also time-consuming. Because of this the existing system becomes very inefficient as it uses more human resources, more time, more capital and is less accurate.

Drawbacks of Existing System

1. Vehicle needs to slow down at the toll collection area.
2. Traffic problems.
3. Collector doesnt collect the toll fee from the owner easily.
4. We need a specific person to monitor the Existing System.
5. Existing System is Time-consuming.
6. In Existing System, the behavior of the Collector can be fraudulent when collecting the toll.

Proposed System

This Project is based on concept of Image Processing. Number plate extraction is hotspot research area in the field of image processing. In this project we examine the image and the respective information will be processing based toll collection system and how to make more efficient and perfect. On any toll booth the vehicle has to stop for paying the toll. We are trying to develop a system that would pay the toll automatically and reduce the queue at the toll booth.

In this system camera is used for capturing the image of the vehicle number plate. According to the vehicle the toll would be cut from the customers account and then open the gate. 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 alarm would buzz which would indicate the operator at the toll booth that the vehicle is a stolen vehicle. For the identification of the vehicles, the information of the vehicles is already stored on the central database. So captured number will be sent to the toll.

Advantages of Proposed System

1. Vehicle does not need to wait at the toll booth
2. Traffic Reduces
3. Corruption Free Toll System
4. We do not need a person to monitor all the activity
5. It makes the process time efficient

Requirement Analysis

4.1 Software Requirements

MATLAB Software: MATLAB Software is used for Image Processing.Image Processing Toolbox provides a comprehensive set of reference-standard algorithms, functions, and apps for image processing, analysis, visualization, and algorithm development. It is also used for creating GUI of the System's main Application.
Operating System: Compatible OS with minimum of 2GB RAM
Xampp: Xampp Software is a Web Server which is used to host a website locally. In our project it is used for our ANPR System's Website which will be used by the users to check their vehicles and account details.
Keil: This is Microcontroller Burner Software.

4.2 Hardware Requirements

Mobile Camera: Mobile Camera is used to capture the image of the car for number plate extraction. For Number Plate Extraction we cannot use a web camera as its quality is very poor for OCR. Mobile camera should be at least 5mp for this process
Web Camera: In our Project we have used web camera to detect the size of the vehicle as this process does not require a high quality camera.
Microcontroller: Microcontroller is used to interface the Software with the Hardware
LCD: LCD is used to display information of the vehicle and amount deducted.
Motor Driver: Motor Driver is used to control the Motor used for gates in the toll system

Language used

5.1 MATLAB

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics.

5.2 Features

- High level language for scientific and engineering computing
- Desktop environment tuned for iterative exploration, design, and problem-solving
- Graphics for visualizing data and tools for creating custom plots
- Apps for curve fitting, data classification, signal analysis, and many other domain-specific tasks
- Add-on toolboxes for a wide range of engineering and scientific applications
- Tools for building applications with custom user interfaces
- Interfaces to C/C++, Java, .NET, Python, SQL, Hadoop, and Microsoft Excel
- Royalty-free deployment options for sharing MATLAB programs with end users

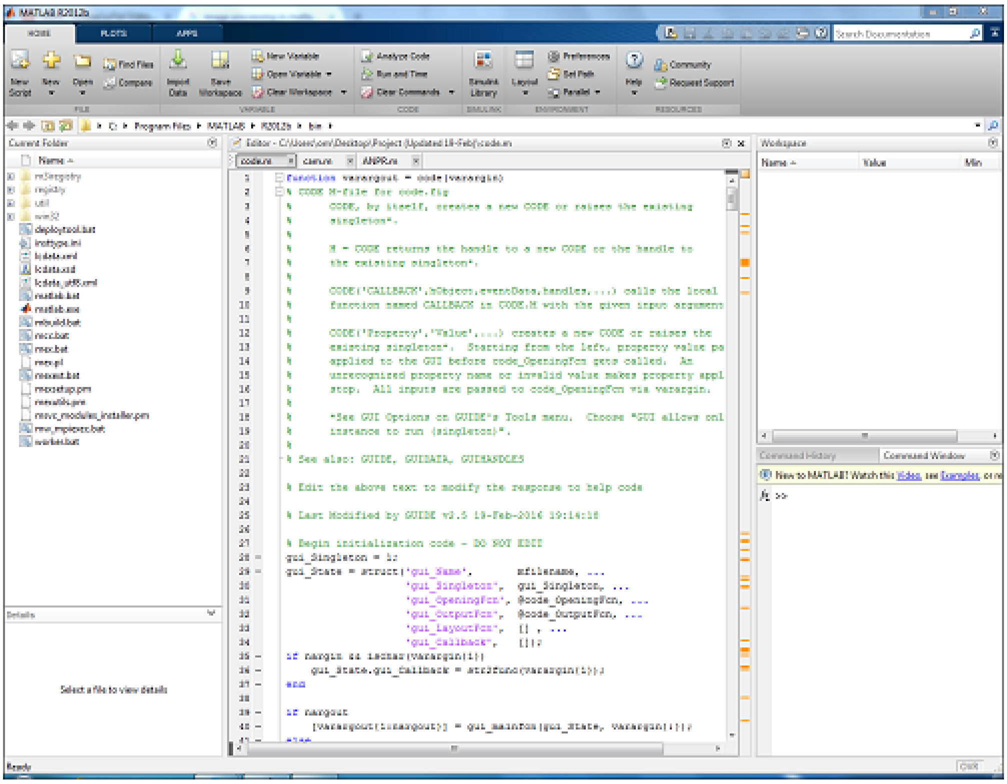
System Components

6.1 MATLAB

MATLAB is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

The Image Processing Toolbox

IPT is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of image processing operations, including Spatial image transformations, Morphological operations, Neighborhood and block operations, Linear filtering and filter design, Transforms, Image Analysis and enhancement, Image registration, Deblurring regio of interest operation, etc.



```
function varargout = code(varargin)
% CODE M-file for code.fig
%      CODE, by itself, creates a new CODE or raises the existing
%      singleton*.
%
%      H = CODE returns the handle to a new CODE or the handle to
%      the existing singleton*.
%
%      CODE('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in CODE.M with the given input argument
%
%      CODE('Property','Value',...) creates a new CODE or raises the
%      existing singleton*.  Starting from the left, property value pa
%      applied to the GUI before code_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property appl
%      stop.  All inputs are passed to code_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows onl
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help code

% Last Modified by GUIDE v2.5 18-Feb-2016 19:14:18

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @code_OpeningFcn, ...
                   'gui_OutputFcn',  @code_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
```

Figure 6.1: MATLAB GUI

6.2 Micro-Controller

For controlling the whole system. The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmels high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM con-tents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

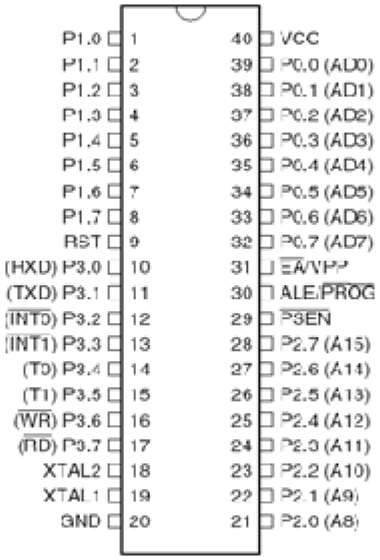


Figure 6.2: Microcontroller Pin Description

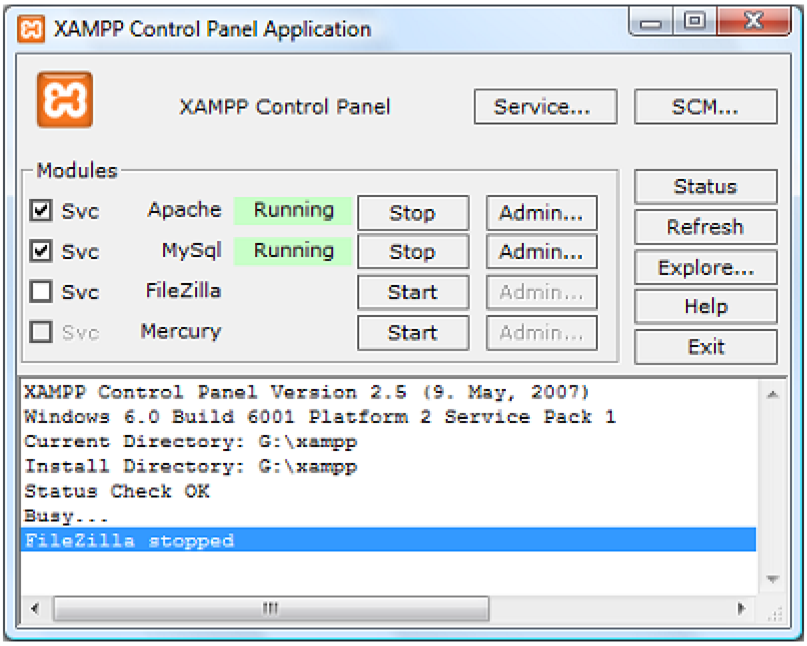
Pins 1-8: Port 1 Each of these pins can be configured as an input or an output.
Pin 9: RS A logic one on this pin disables the microcontroller and clears the contents of most registers. In other words, the positive voltage on this pin resets the microcontroller. By applying logic zero to this pin, the program starts execution from the beginning.
Pins10-17: Port 3 Similar to port 1, each of these pins can serve as general input or output. Besides, all of them have alternative functions:
Pin 10: RXD Serial asynchronous communication input or Serial synchronous communication output.
Pin 11: TXD Serial asynchronous communication output or Serial synchronous communication clock output.
Pin 12: INT0 Interrupt 0 input.
Pin 13: INT1 Interrupt 1 input.
Pin 14: T0 Counter 0 clock input.
Pin 15: T1 Counter 1 clock input.
Pin 16: WR Write to external (additional) RAM.
Pin 17: RD Read from external RAM.
Pin 18, 19: X2, X1 Internal oscillator input and output. A quartz crystal which specifies operating frequency is usually connected to these pins.
Pin 20: GND Ground.
Pin 21-28: Port 2 If there is no intention to use external memory then these port pins are configured as general inputs/outputs. In case external memory is used, the higher address byte, i.e. addresses A8-A15 will appear on this port. Even though memory with capacity of 64Kb is not used, which means that not all eight port bits are used for its addressing, the rest of them are not available as inputs/outputs.
Pin 29: PSEN If external ROM is used for storing program then a logic zero (0) appears on it every time the microcontroller reads a byte from memory.
Pin 30: ALE Prior to reading from external memory, the microcontroller puts the lower address byte (A0-A7) on P0 and activates the ALE output. After receiving signal from the ALE pin, the external register (usually 74HCT373 or 74HCT375 add-on chip) memorizes the state of P0 and uses it as a memory chip address. Immediately after that, the ALU pin is returned its previous logic state and P0 is now used as a Data Bus. As seen, port data multiplexing is performed by means of only one additional (and cheap) integrated circuit. In other words, this port is used for both data and address transmission.
Pin 31: EA By applying logic zero to this pin, P2 and P3 are used for data and address transmission with no regard to whether there is internal memory or not. It means that even there is a program written to the microcontroller, it will not be executed. Instead, the program written to external ROM will be executed. By applying logic one to the EA pin, the microcontroller will use both memories, first internal then external (if exists).
Pin 32-39: Port 0 Similar to P2, if external memory is not used, these pins can be used as general inputs/outputs. Otherwise, P0 is configured as address output (A0-A7) when the ALE pin is driven high (1) or as data output (Data Bus) when the ALE pin is driven low (0).
Pin 40: VCC +5V power supply.

6.3 Xampp

XAMPP stands for Cross-Platform, Apache, MySQL, PHP and Perl. It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server server application (Apache), database (MySQL), and scripting language (PHP) is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server is extremely easy as well.

XAMPP has 3 primary components. These are:

- **1. Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54
- **2. MySQL:** Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the worlds most popular database management system.
- **3. PHP:** PHP stands for Hypertext Preprocessor. It is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn, and works perfectly with MySQL, making it a popular choice for web developers



☑ Svc Apache Running Stop Admin...
☑ Svc MySql Running Stop Admin...
□ Svc FileZilla Start Admin...
□ Svc Mercury Start Admin...

```
XAMPP Control Panel Version 2.5 (9. May, 2007)
Windows 6.0 Build 6001 Platform 2 Service Pack 1
Current Directory: G:\xampp
Install Directory: G:\xampp
Status Check OK
Busy...
FileZilla stopped
```

Figure 6.3: Xampp Control Panel

6.4 LCD

For Displaying the toll amount of vehicle. LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special and even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

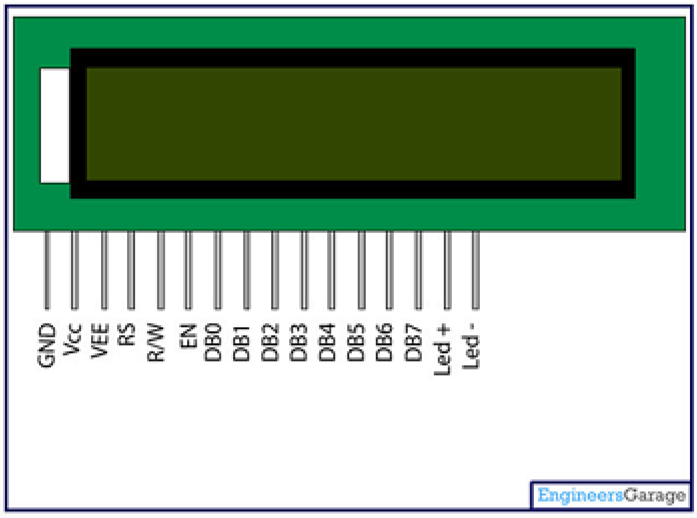


Figure 6.4: lcd Pin Description

6.5 Motor Driver

Since motors require more current then the microcontroller pin can typically generate, you need some type of a switch. which can accept a small current, amplify it and generate a larger current, which further drives a motor. This entire process is done by what is known as a motor driver. L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC, Dual H-bridge Motor Driver integrated circuit (IC). The l293d can drive small and quiet big motors as well.

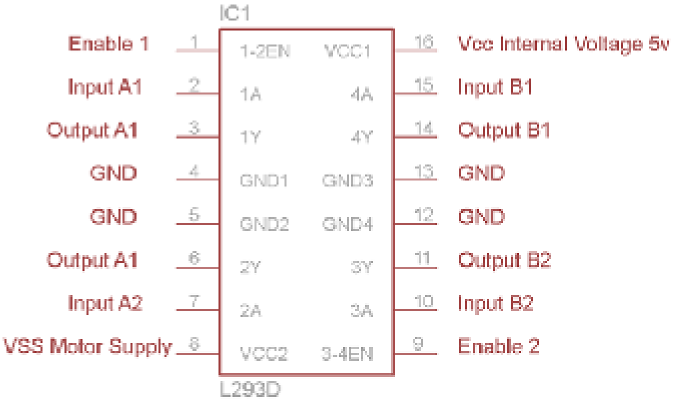


Figure 6.5: Motor Driver Pin Description

6.6 USB to TTL

It is use to interface the PC with the Micro-controller. The cable is easiest way ever to connect to your microcontroller/Raspberry Pi/WiFi router serial console port. Inside the big USB plug is a USB¡-¿Serial conversion chip and at the end of the 36” cable are four wire - red power, black ground, white RX into USB port, and green TX out of the USB port.

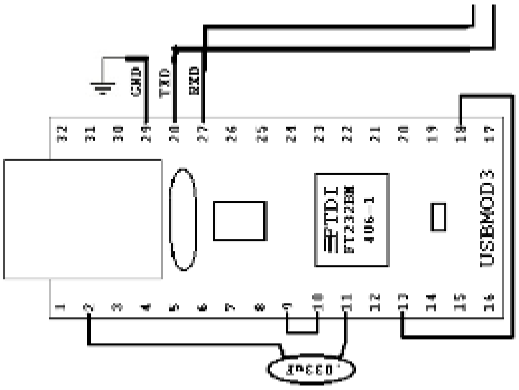


Figure 6.6: USB to TTL Pin Description

Project Planning

Project development life cycle is composed of a number of clearly defined and distinct work phases which are used by for planning, designing, building, testing, and delivering information systems. Like anything that is manufactured on an assembly line, project planning aims to produce high quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time-frames and cost estimates.

The project development life cycle framework provides a sequence of activities for system designers and developers to follow.

- Preliminary analysis and System Requirements
- System Design and Analysis
- Development
- Testing
- Implementation and Evaluation

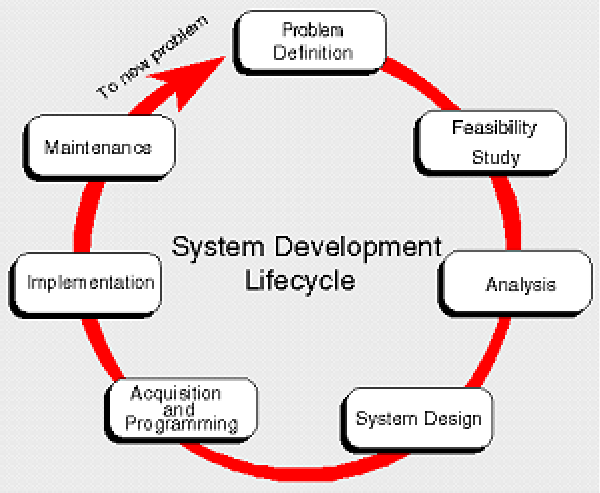


Figure 7.1: Project Planning

7.1 Preliminary Analysis

It consists of conducting a preliminary analysis, proposing alternative solutions and describing costs and benefits of project.

Conduct Preliminary Analysis: Before beginning the project we need to find out the scope and the nature of the project; the existing drawbacks of the current system and how the project will overcome those drawbacks.

Propose Alternative Solutions: After determining the drawback of existing system,we need to find out ways that will overcome these drawbacks. Indubitably during the analysis of the project, you might have come across some solutions that will help in overcoming those drawbacks. In accordance with proposing these alternative solutions, you have to implement them in the real world so as to improve the existing systems.

7.1.1 System Analysis

The systems investigate the IT proposal. During this step, we must consider all current priorities that would be affected and how they should be handled. Before any system planning is done, a feasibility study should be conducted to determine if creating a new or improved system is a viable solution. This will help to determine the costs, benefits, resource requirements, and specific user needs required for completion. The development process can only continue once management approves of the recommendations from the feasibility study.

7.1.2 Feasibility Study

Project feasibility is basically the likelihood the system will be useful to the organization. The main objective of feasibility study is to test the technical, operational and economical feasibility for adding new modules and debugging the old running system.

Economical Feasibility

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

The main aim of our project is to make the traffic system as automatic as possible. Thus the Aim of our Project reduces the manpower need. As less manpower is needed it decreases the capital expenditure on their salary. The Capital saved in their salary is way too high compared to the investment put on the system.

As our System also identifies vehicles breaking traffic rules as well as detecting stolen vehicle so it increases the revenue of the traffic department. The System also reduces corruption to a great extent thus saving lot of capital. Thus taking the revenue and the capital saved in manpower in consideration, implementing ANPR System is economically feasible

Technical Feasibility

Technical feasibility is one of the first studies that must be conducted after the project has been identified. In large engineering projects consulting agencies that have large staffs of engineers and technicians conduct technical studies dealing with the projects. It concludes that whether the technologies and methods used in the project can be implemented or not.
In our Project, the technologies we have used are image processing algorithms which are being used by the industry since long time. So implementing them is not a big deal. Technical feasibility also includes the accuracy your system is giving in the services it provides.
So after performing many test we have found that our project has about 80-90% accuracy which is good. Thus making ANPR System Technically Feasible.

Social Feasibility

Social Feasibility is defined as the way in which the system is helping people socially in their life. The ANPR System is helping people in lot of way. It helps people saving their money by paying online. It saves people's time by making the process more automatic. It prevents corruption thus making the society happy and stable.
People can easily recharge their account as well as find their stolen vehicle easily. Finding Stolen vehicle was very tedious process earlier but now due to ANPR system the precess has become very easy. Thus making ANPR System socially feasible.

7.1.3 System Requirements

It involves identifying the requirements of the end users. It defines project goals into defined functions and operation of the intended application.

7.1.4 Design

Most creative and challenging phase of system lie cycle is the system design. Design refers to the technical specification that will be applied on implementing the actual project. It produces the detail that states how a system will meet the requirements identified during the system analysis

It determines how the output is to be produced and in what format. Input data and the master files are designed to meet the requirements of proposed output. System design also describes data to be input, calculated or stored.

7.1.5 Development

This Part of the Project includes understanding different modules of the Project and Algorithms used in the Project This Algorithms are converted into Coded. The real code of programming is written here by the developer.

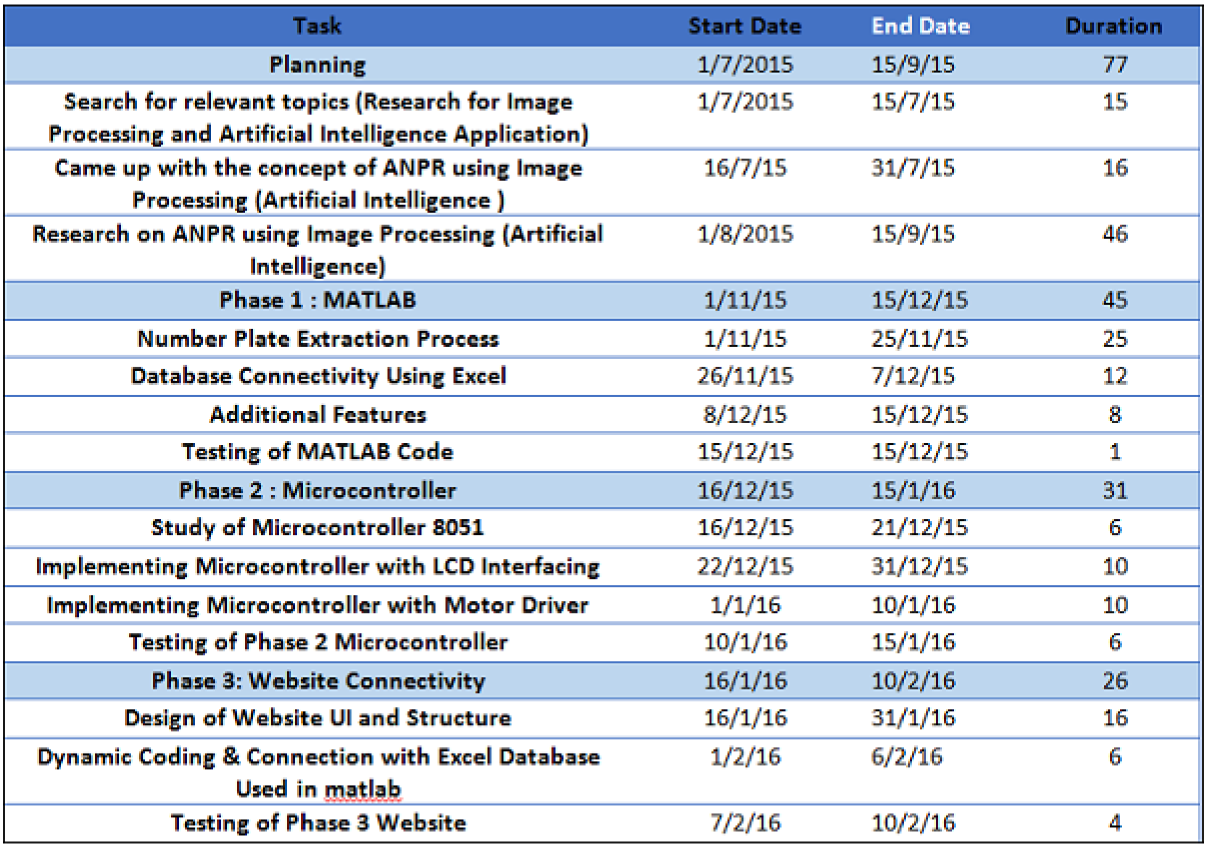
7.1.6 Testing

Once the programs are written, procedures are prepared, organizational changes are implemented, data are collected and hardware is installed the new system is ready to test. During testing desired system performance as stated in system requirements specification is compared with the actual performance. Unsatisfactory test results leads to repetition of one or more phase of development in order to correct the problems responsible for poor system performance.

7.1.7 Maintenance

Describes tasks to operate and maintain information systems in production environment. Includes post implementation and in-process reviews.

7.2 Plan of Work



Task	Start Date	End Date	Duration
Planning	1/7/2015	15/9/15	77
Search for relevant topics (Research for Image Processing and Artificial Intelligence Application)	1/7/2015	15/7/15	15
Came up with the concept of ANPR using Image Processing (Artificial Intelligence)	16/7/15	31/7/15	16
Research on ANPR using Image Processing (Artificial Intelligence)	1/8/2015	15/9/15	46
Phase 1 : MATLAB	1/11/15	15/12/15	45
Number Plate Extraction Process	1/11/15	25/11/15	25
Database Connectivity Using Excel	26/11/15	7/12/15	12
Additional Features	8/12/15	15/12/15	8
Testing of MATLAB Code	15/12/15	15/12/15	1
Phase 2 : Microcontroller	16/12/15	15/1/16	31
Study of Microcontroller 8051	16/12/15	21/12/15	6
Implementing Microcontroller with LCD Interfacing	22/12/15	31/12/15	10
Implementing Microcontroller with Motor Driver	1/1/16	10/1/16	10
Testing of Phase 2 Microcontroller	10/1/16	15/1/16	6
Phase 3: Website Connectivity	16/1/16	10/2/16	26
Design of Website UI and Structure	16/1/16	31/1/16	16
Dynamic Coding & Connection with Excel Database Used in matlab	1/2/16	6/2/16	6
Testing of Phase 3 Website	7/2/16	10/2/16	4

Figure 7.2: Plan of Work

7.3 Analysis Model

Incrmental Model

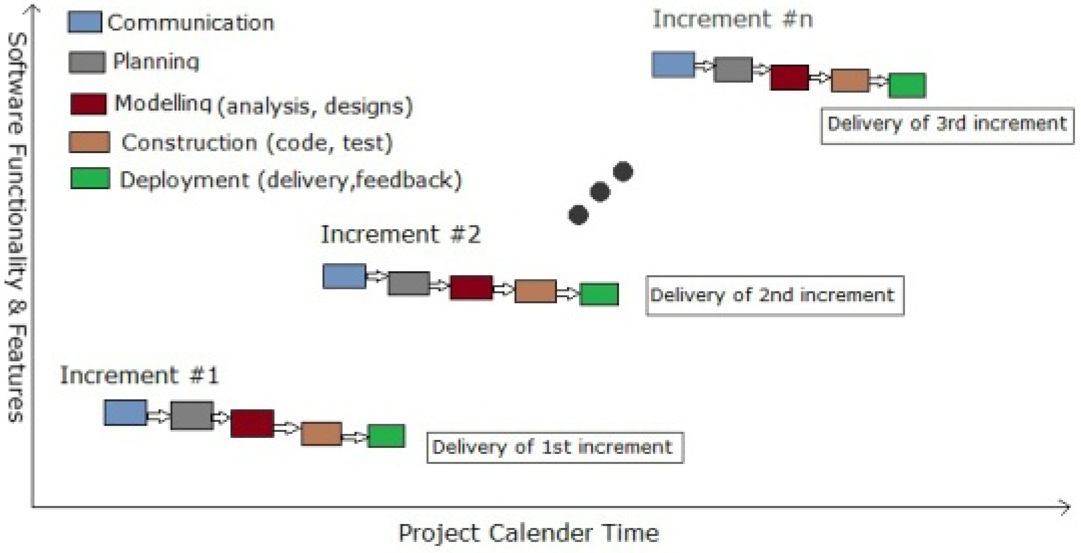

Communication
Planning
Modelling (analysis, designs)
Construction (code, test)
Deployment (delivery,feedback)
Increment #n
Delivery of 3rd increment
Increment #2
Delivery of 2nd increment
Increment #1
Delivery of 1st increment
Software Functionality & Features
Project Calender Time

Figure 7.3: Analysis Model

7.3.1 Incremental Model

To develop any software we need to adapt any one of available software models for successful and timely development and implementation of that software. So to build our ANPR System we also used the Incremental Model as we were going to add features to our projects in increments.

In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a multi-waterfall cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

7.3.2 How we used this Model

- The first build of our Project was the very basic and core functionalist which consisted of the very basic application which just extracts the number plate of the car and does not perform any manipulation

- In the second build we connected the database thus adding features like Accounting System, Stolen Vehicle Registration etc

- The third build of our Project was the additional features like Traffic Detection. We also added a website to allow users to interact with the system by checking account balance, registering themselves with the system, registering stolen vehicle, Reciept System etc.

- The last and the fourth build was adding the hardware part to our system like the gates and lcd.

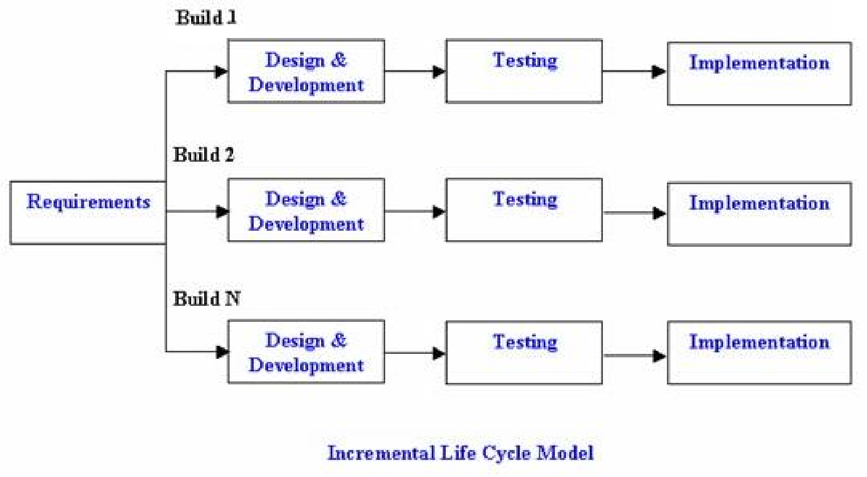


Figure 7.4: Incremental Model

System Analysis and Design

The term design describes a final system and the process by which it is developed. It refers to the technical specification that will be applied in implementing the system. Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements

ER Diagram

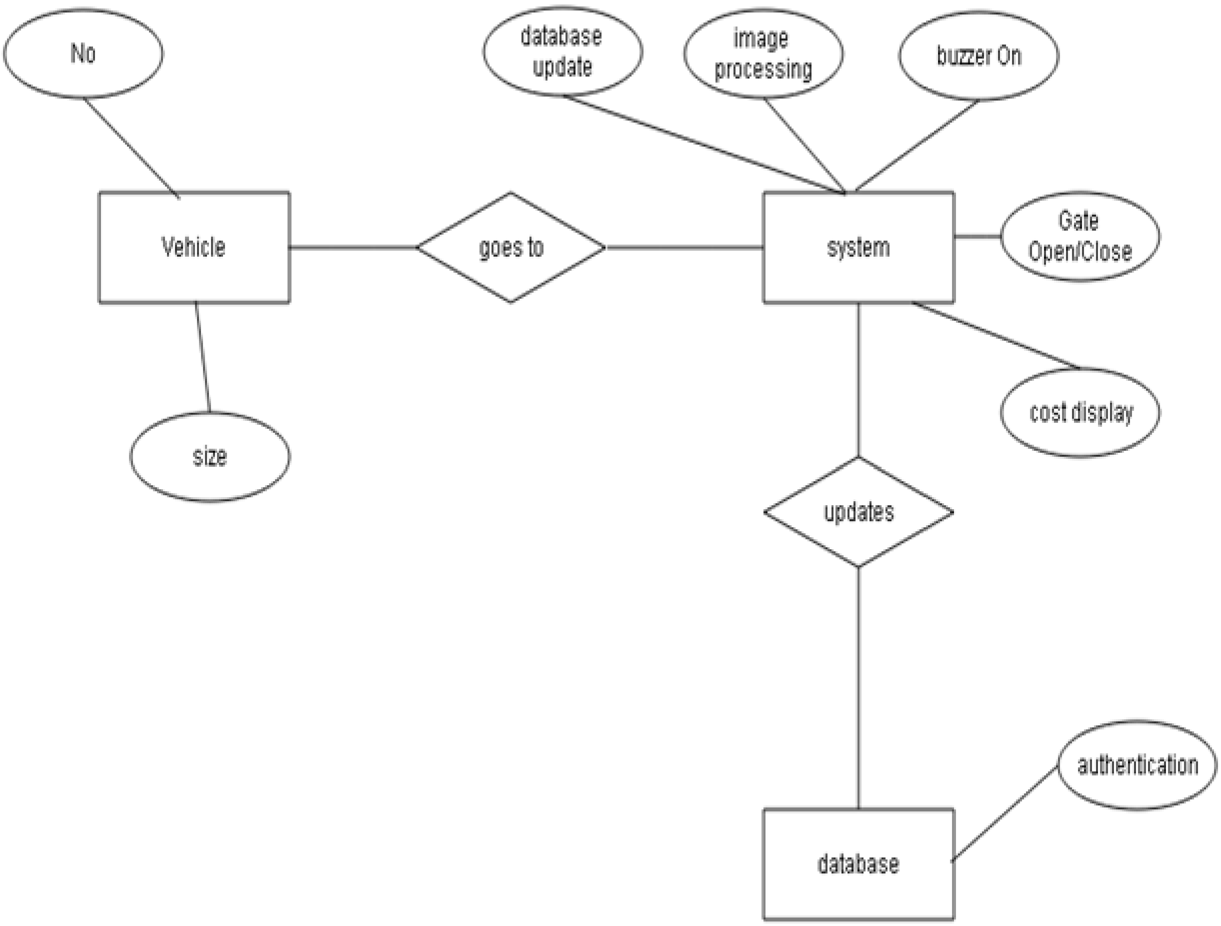


Figure 8.1: ER Diagram

Flow Chart

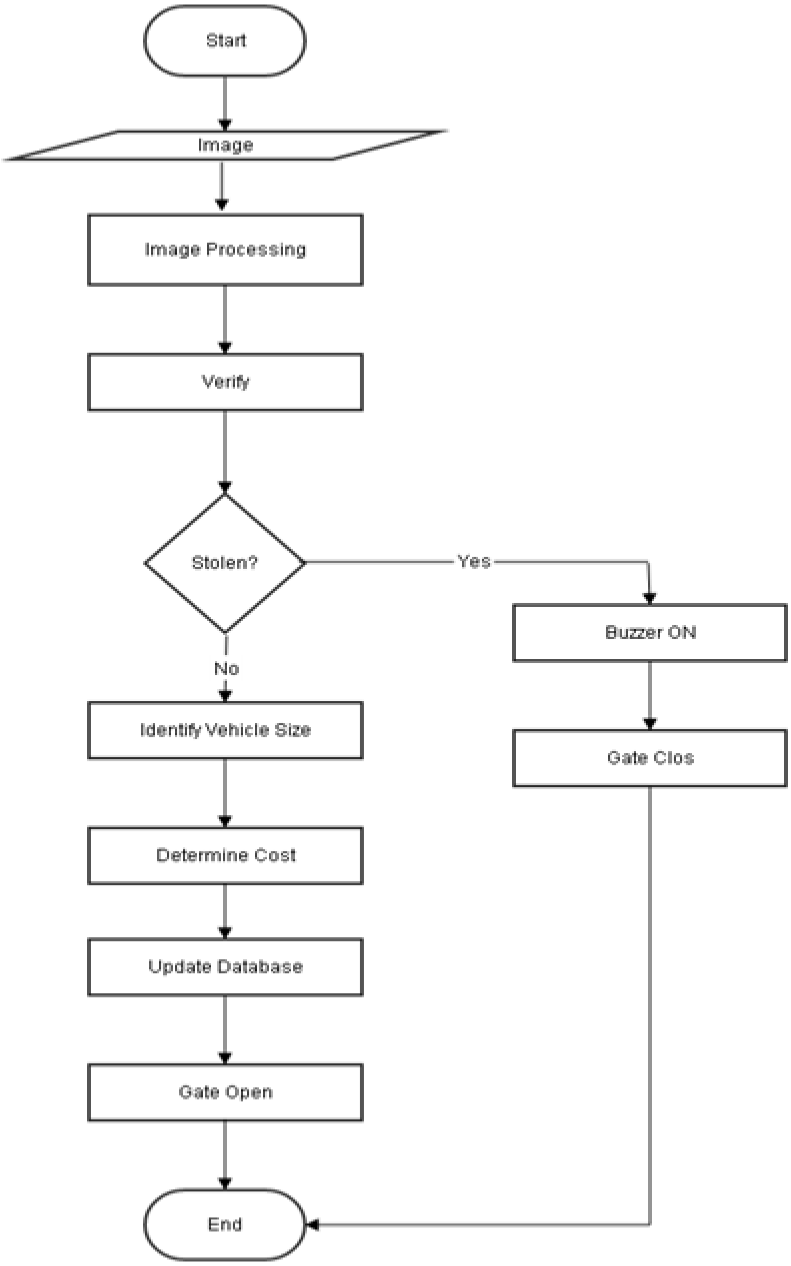


Figure 8.2: Flow Chart

Data Flow Diagram

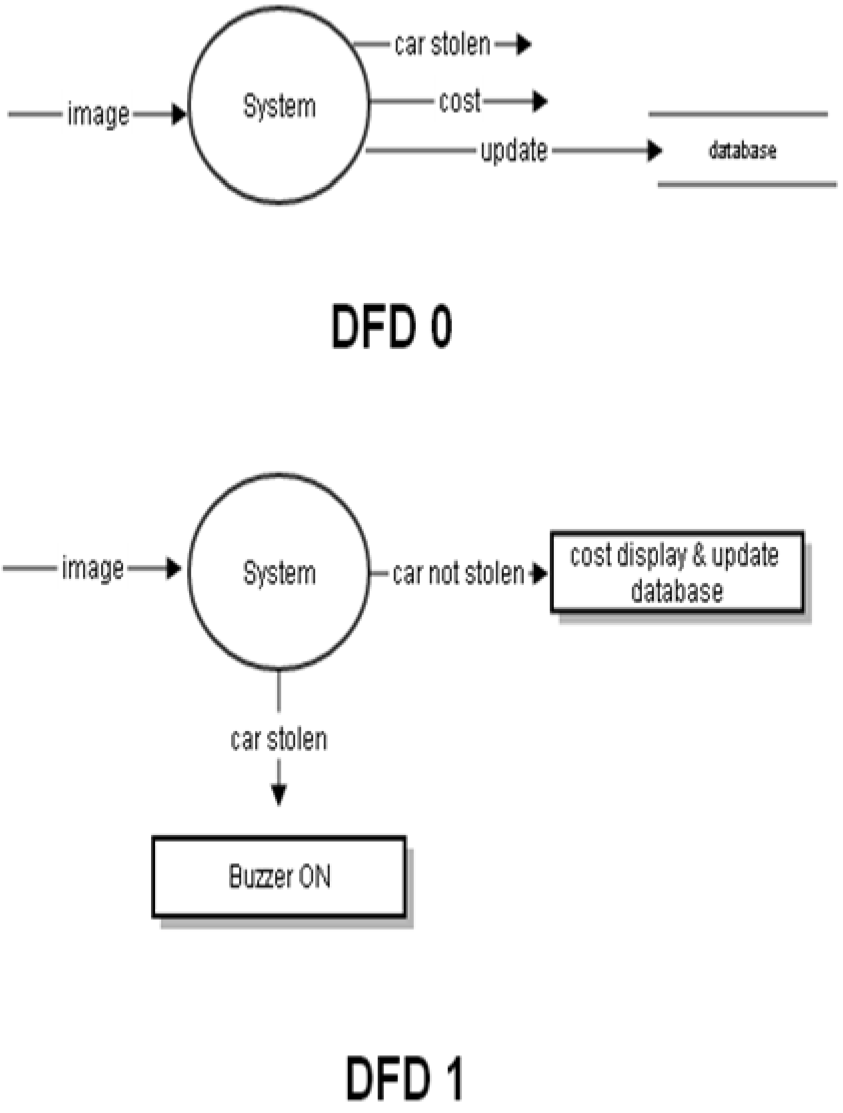


Figure 8.3: Data Flow Diagram

UML Diagram: Class Diagram

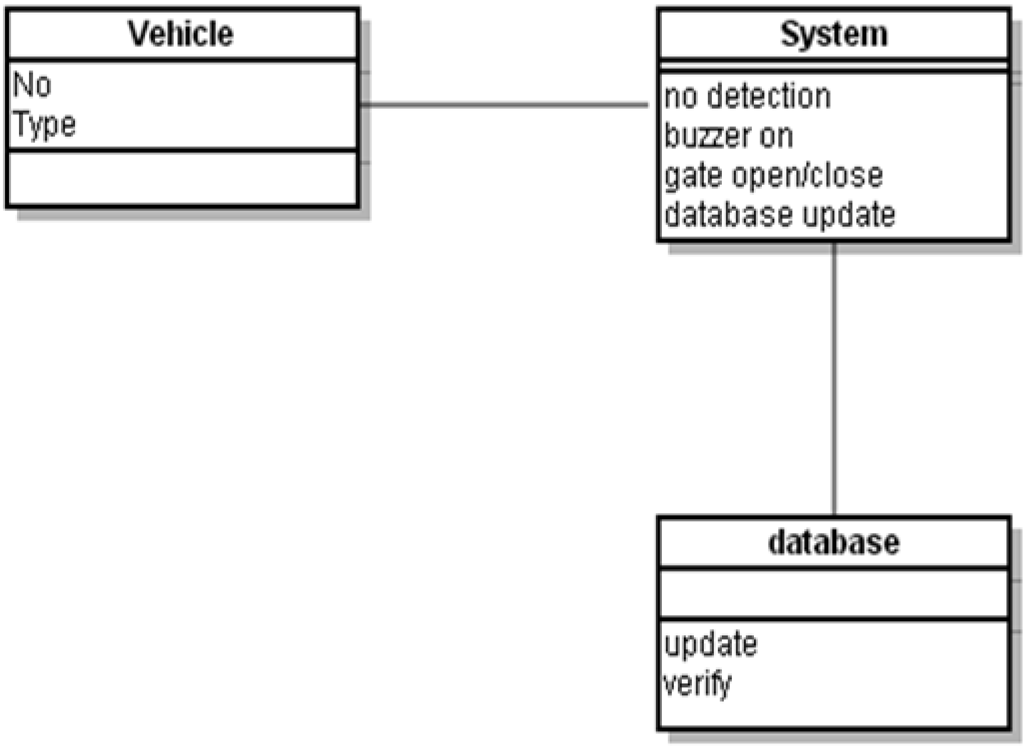


Figure 8.4: UML Diagram

UML Diagram: Use Case Diagram

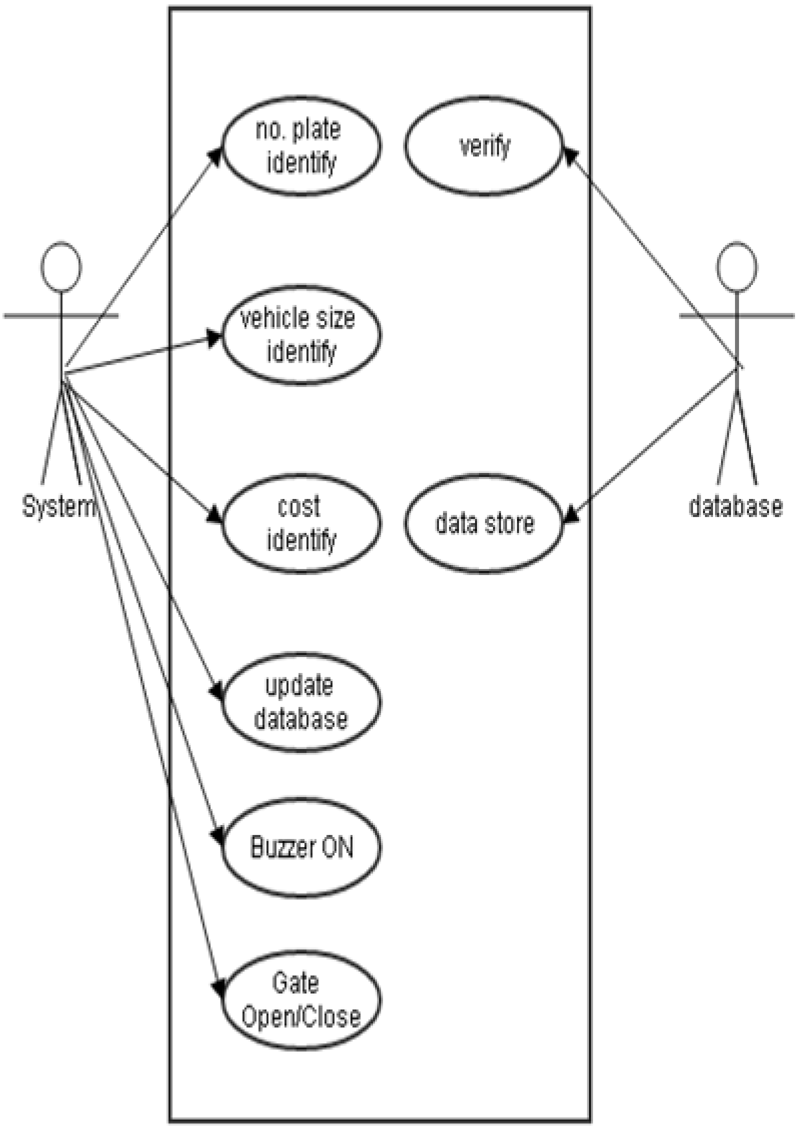


Figure 8.5: Use Case Diagram

Sequence Diagram

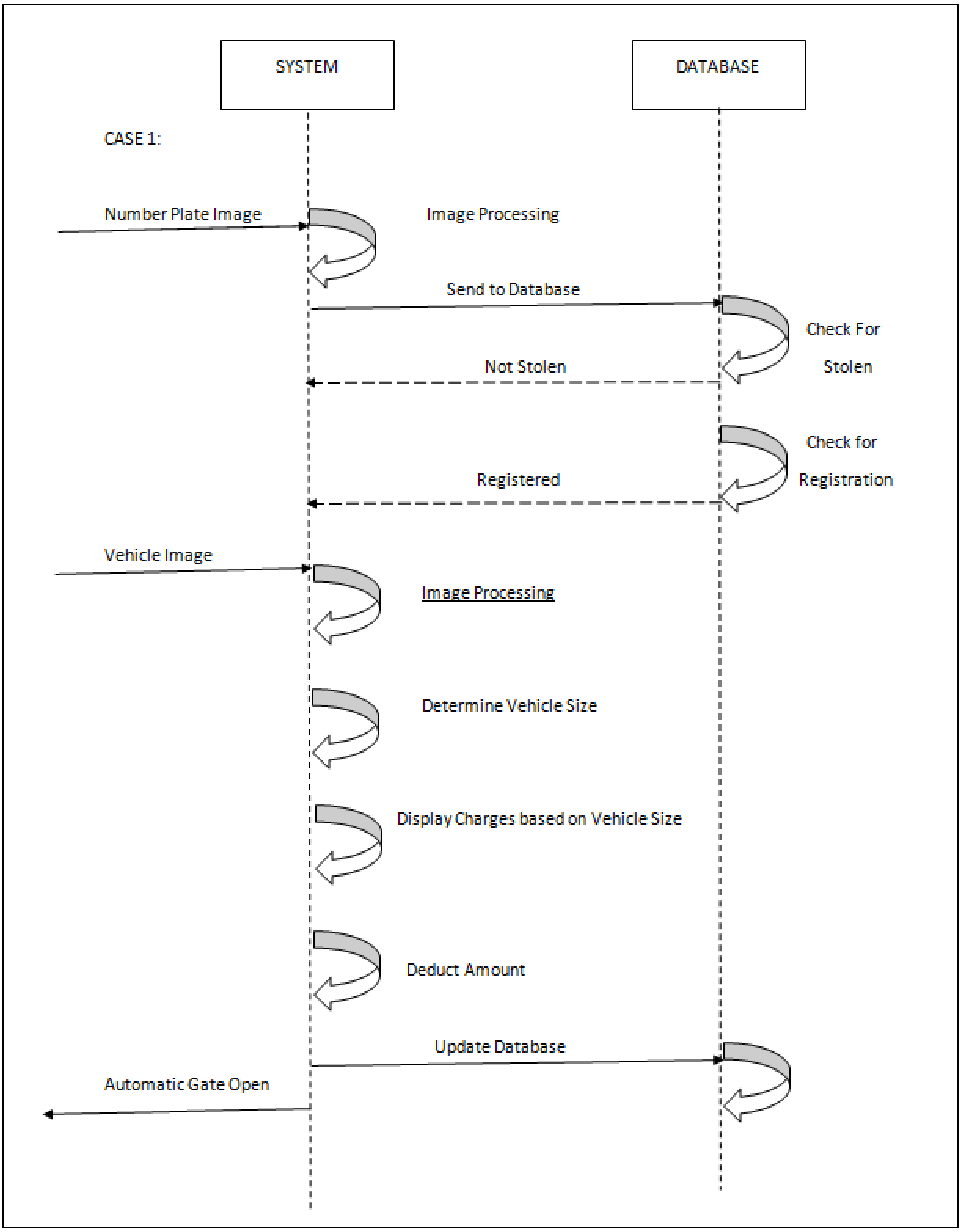


Figure 8.6: Sequence Diagram

Sequence Diagram

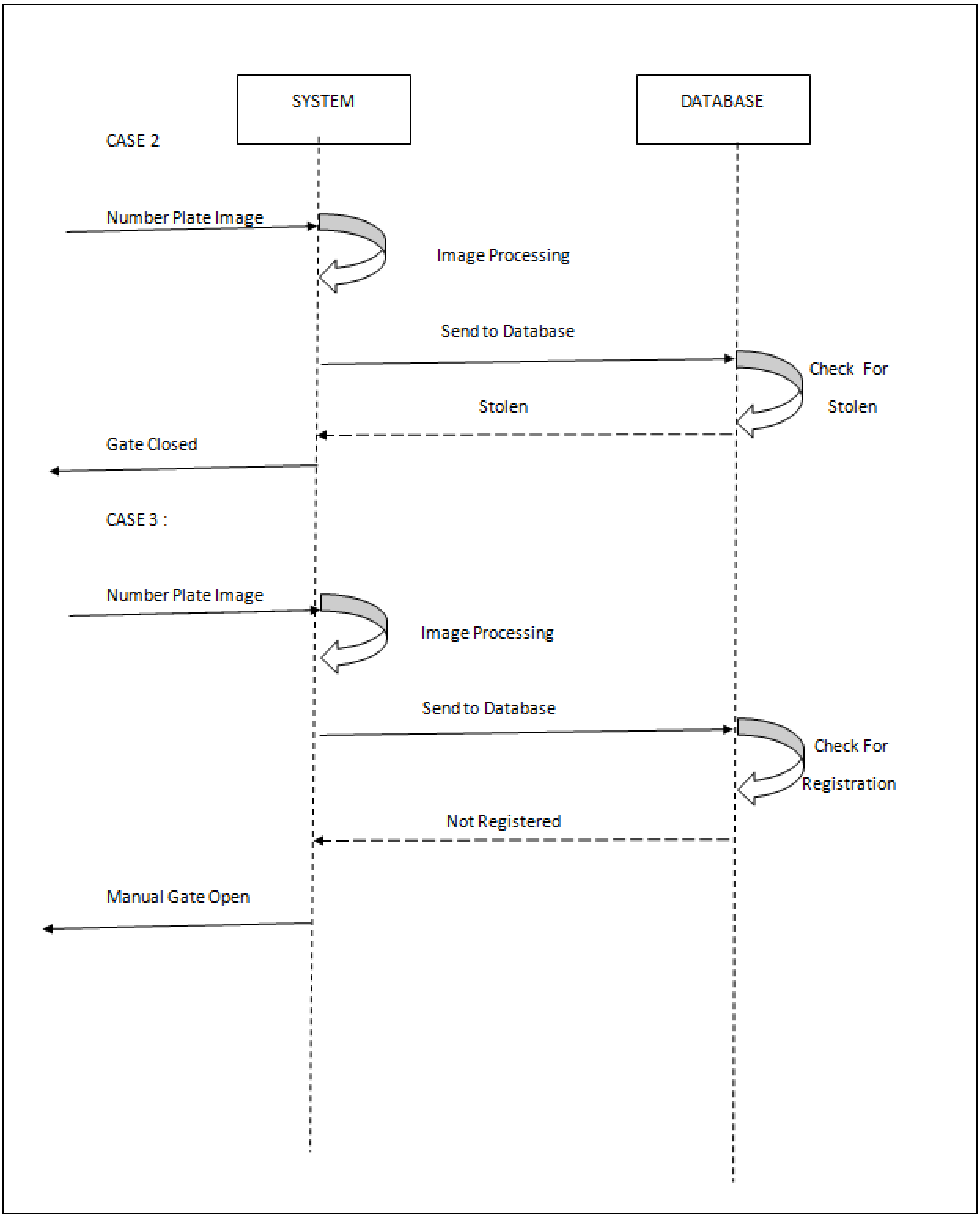


Figure 8.7: Sequence Diagram

Implementation

9.1 Modules

9.1.1 System model Diagram

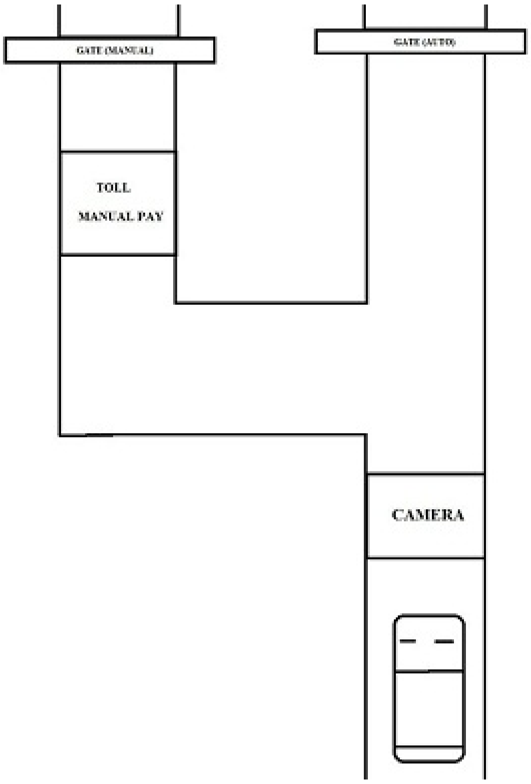


Figure 9.1: System Model Diagram

9.1.2 Camera

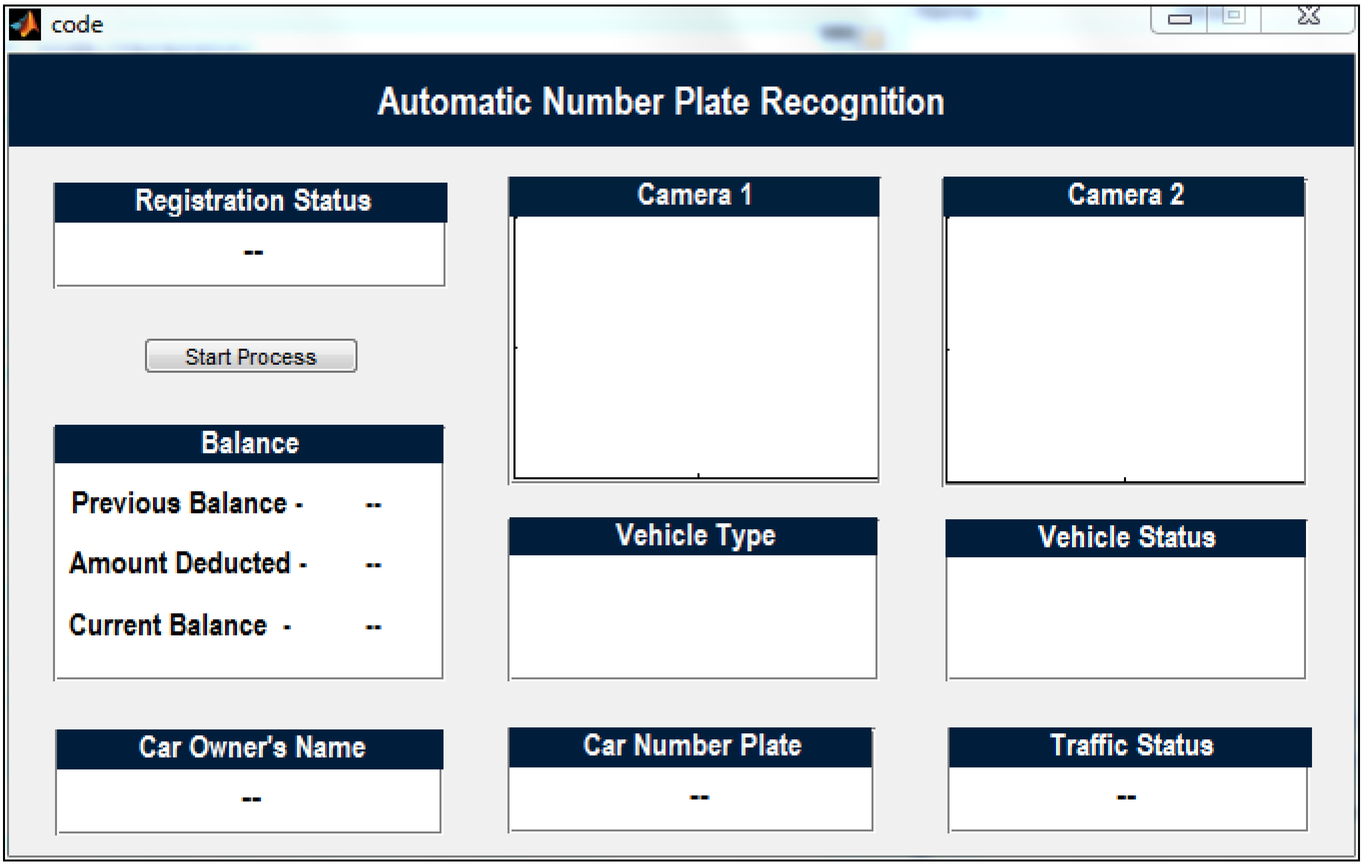
Camera is the first module of the system which works as a input for the system by capturing images of the car.It sends the captured image to the System's main application

9.1.3 System's Main Application

This is the main part of the whole system were all the image processing manipulation takes place. It takes the captured images as input from the camera modules and then process it and gives number plate in the text form and the size of the car as the output. This application is heart of the system were different image processing algorithms are implemented. It also has GUI so that the user can interact with the system easily. The Application gives information about the following thing:

- **Registration Status:** It displays whether the car is registered or not
- **Vehicle Type:** It displays size of the car : light weight or heavy weight
- **Vehicle Status:** It displays whether the car is safe or stolen
- **Car Owner's Name:** It displays the name of the car owner
- **Car Number Plate:** It displays the extracted cars number plate through image processing
- **Traffic:** It display the current traffic based on the frequency
- **Vehicle Balance:** It displays the vehicles previous balance, current balance, and the amount deducted.
- **Cameras:** It shows the Camera 1 and Camera 2

System's Main Application UI



Automatic Number Plate Recognition

Registration Status: --

Camera 1

Camera 2

Balance

Previous Balance - --

Amount Deducted - --

Current Balance - --

Vehicle Type

Vehicle Status

Car Owner's Name: --

Car Number Plate: --

Traffic Status: --

Figure 9.2: System's Main Application User Interface

9.1.4 Micro-Controller

This is supplementary part of the system which takes input from the system's main application and responds accordingly. It consist of:

- **LCD:** To Display the vehicle and transaction information
- **Gates:** This are the gates of the toll system control by the motor driver.

There are three scenario's of the System

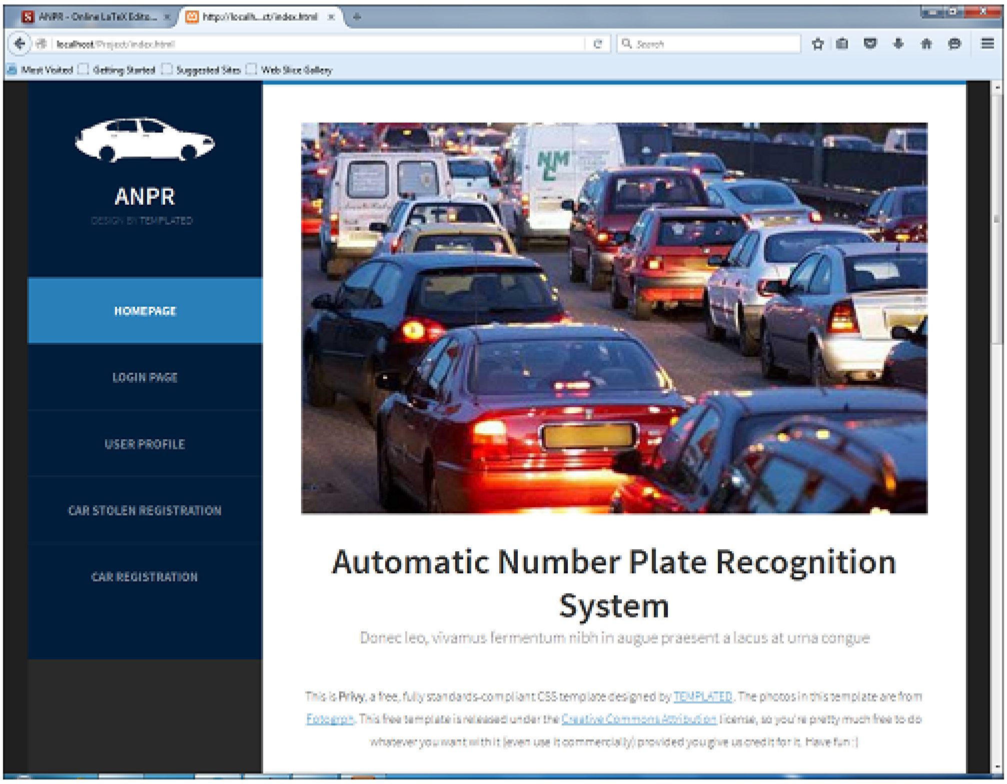
- **Registered Car:** In this scenario the microcontroller will give the LCD information about the Vehicle and will open the Automatic Gate were the payment is done online.
- **Unregistered Car:** In this scenario the microcontroller will not give Information about the vehicle to the LCD and will open the manual gate were the payment is done manually.
- **Stolen Car:** In this scenario the microcontroller will give the information about the vehicle to the LCD and will not open any gate as it is a stolen vehicle.

9.1.5 User Website

User Website is client side application of our system in which the user can Register, Login, Check Balance and Details, Register Stolen Vehicles. This website would be accessible to any user even at his home.
It consist of the following information:

- **Registration Page:** It is the webpage which consist of form were the users can register themselves for the ANPR service.
- **Login Page:** It is the webpage which consist of authentication for the user.
- **User Details:** It consist of webpage which shows the users vehicle information.
- **Stolen Vehicle Registration:** It consist of webpage were user can register themselves if their car is stolen

Website Home Page



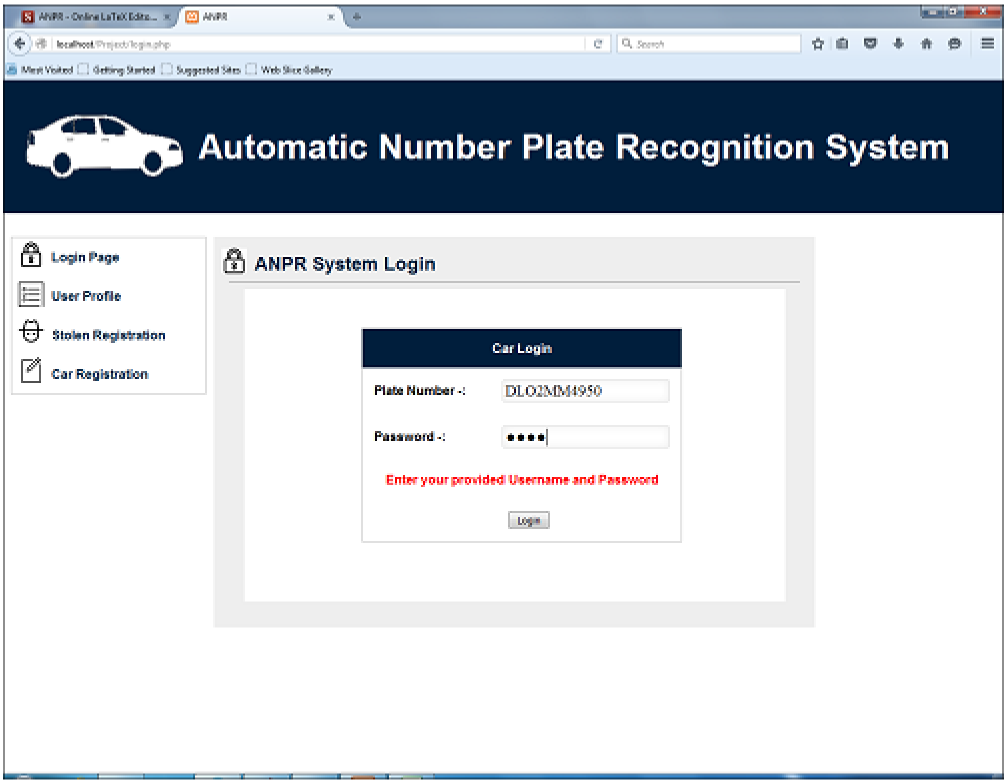
Automatic Number Plate Recognition System

Donec leo, vivamus fermentum nibh in augue praesent a lacus at urna congue

This is Privy, a free, fully standards-compliant CSS template designed by TEMPLATED. The photos in this template are from Fotogrph. This free template is released under the Creative Commons Attribution license, so you're pretty much free to do whatever you want with it (even use it commercially) provided you give us credit for it. Have fun :)

Figure 9.3: Website Home Page

Website Login Page



Automatic Number Plate Recognition System

ANPR System Login

Car Login

Plate Number -: DLO2MM4950

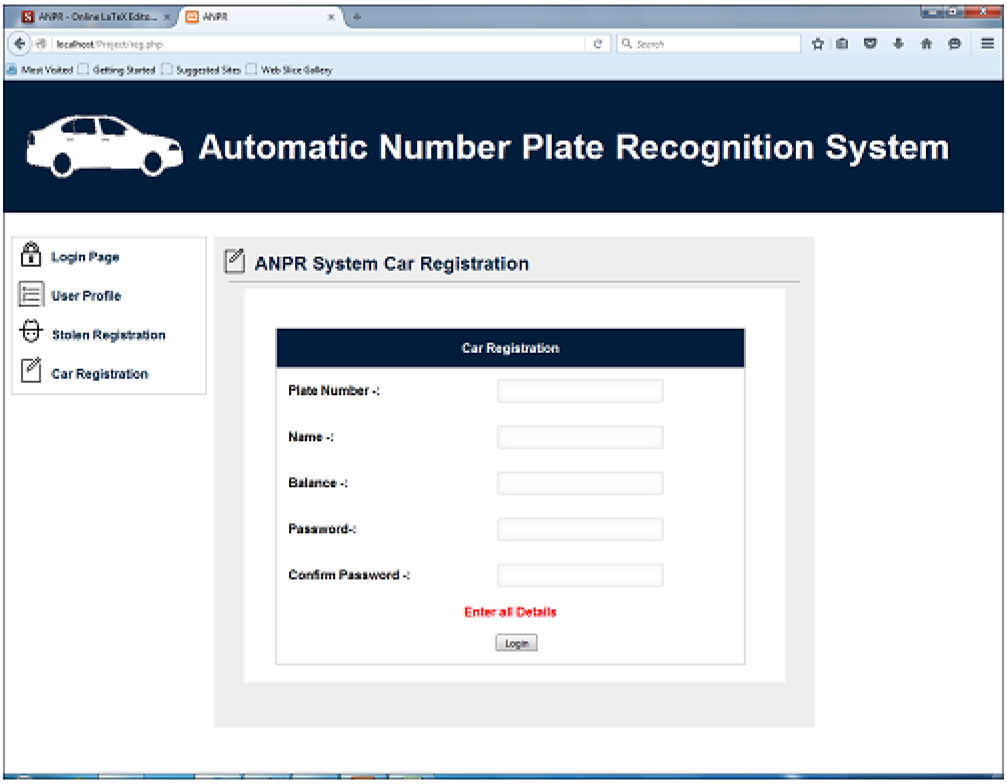
Password -: ●●●●

Enter your provided Username and Password

Login

Figure 9.4: Website Login Page

Website Registration Page



Automatic Number Plate Recognition System

ANPR System Car Registration

Car Registration

Plate Number -:

Name -:

Balance -:

Password-:

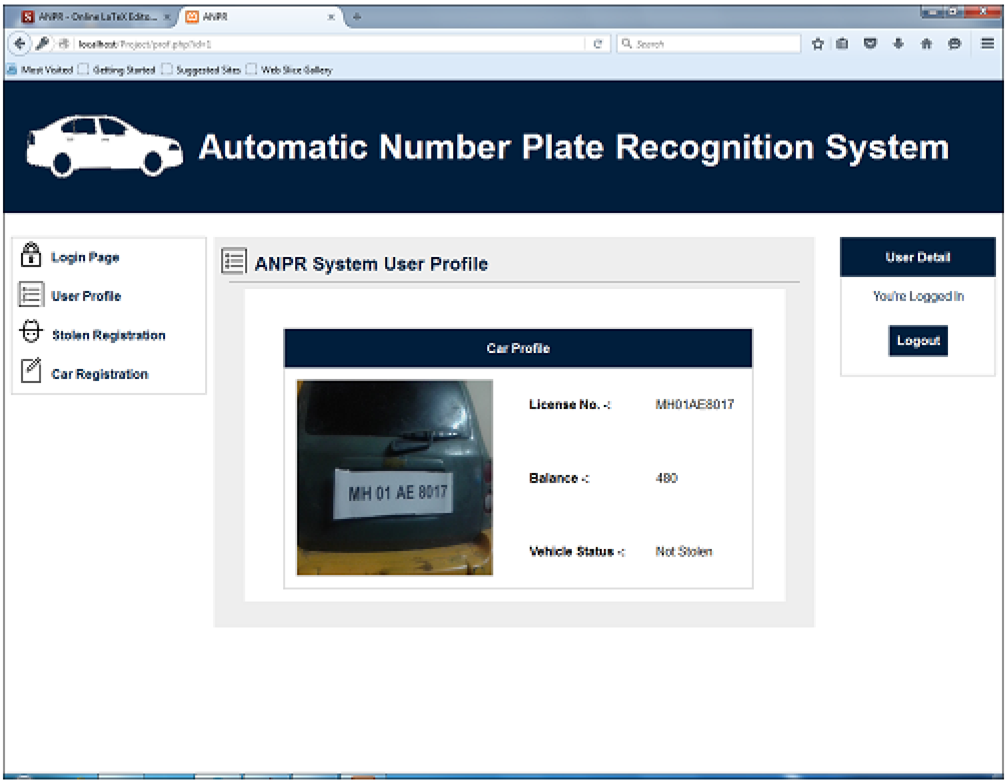
Confirm Password -:

Enter all Details

Login

Figure 9.5: Website Registration Page

Website User Profile Page



ANPR System User Profile

Car Profile

License No. -: MH01AE8017

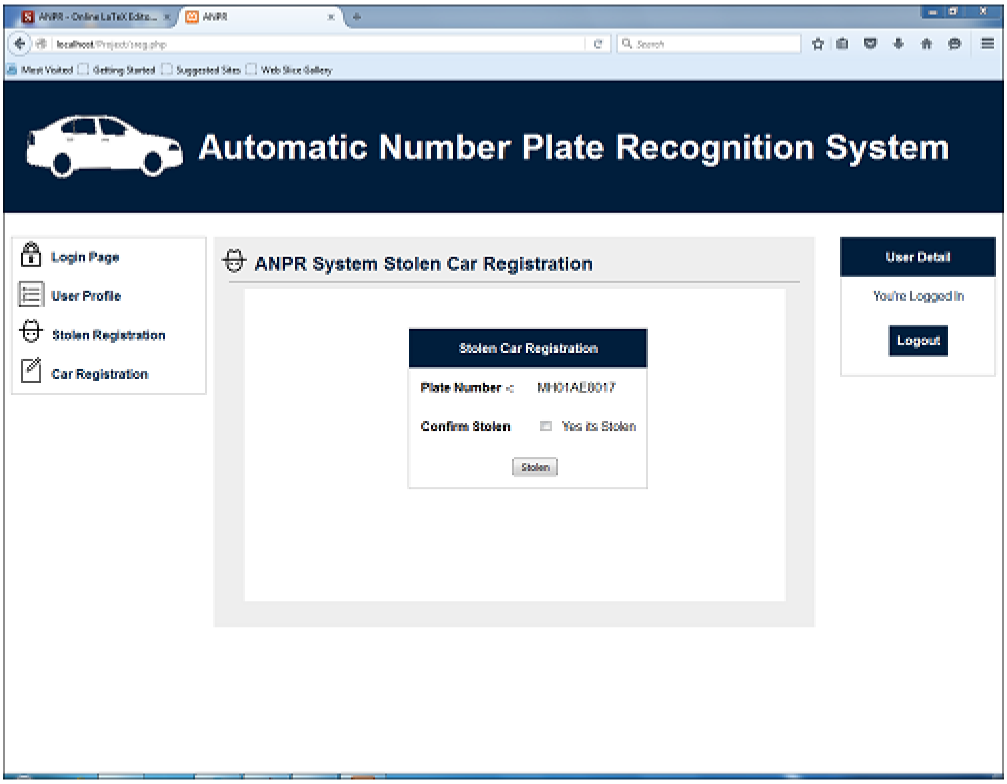
Balance -: 480

Vehicle Status -: Not Stolen

You're Logged In

Figure 9.6: Website Home Page

Website Stolen Vehicle Registration Page



ANPR System Stolen Car Registration

Stolen Car Registration

Plate Number -: MH01AE8017

Confirm Stolen □ Yes its Stolen

You're Logged In

Figure 9.7: Website Stolen Vehicle Registration Page

9.2 Algorithm

In mathematics and computer science, an algorithm is a self-contained step-by-step set of operations to be performed. Algorithms exist that perform calculation, data processing, and automated reasoning. An algorithm is an effective method that can be expressed within a finite amount of space and time and in a well-defined formal language for calculating a function.

9.2.1 RGB to Gray Conversion

What is RGB Image?
The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.
If the RGB image is 24-bit (the industry standard as of 2005), each channel has 8 bits, for red, green, and blue in other words, the image is composed of three images (one for each channel), where each image can store discrete pixels with conventional brightness intensities between 0 and 255.

What is Grayscale Image?
In computing, a grayscale image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale image is of 8-bit which stores pixels brightness intensity information between 0 to 255

Conversion: RBG to Gray

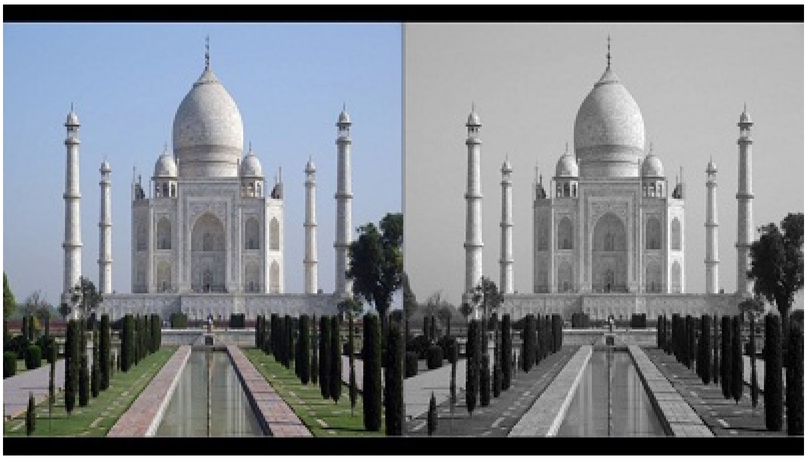


Figure 9.8: **Right Side:** RGB Image **Left Side:** Grayscale Image

9.2.2 Noise Reduction

What is Noise?
Image noise is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise.

Noise Reduction using Linear Smoothing Filter?
Here value of each pixel is set equal to the average value of its 8 neighboring pixels. This type of

filter for noise removal is called linear smoothing filter. Such a filter is very effective in terms of performance and speed.

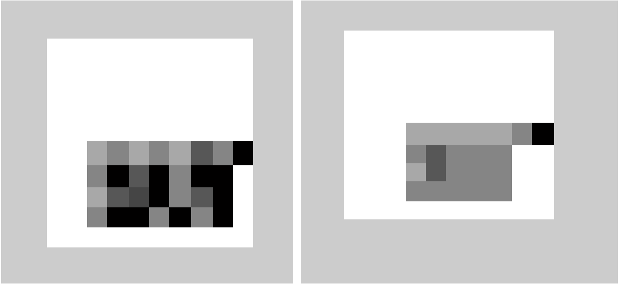


Figure 9.9: **Right Side:** Image with Noise **Left Side:** Image without Noise

9.2.3 Edge Detection

In this process we have used Canny Edge Detector for Edge Detection

Canny Edge Detector
The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

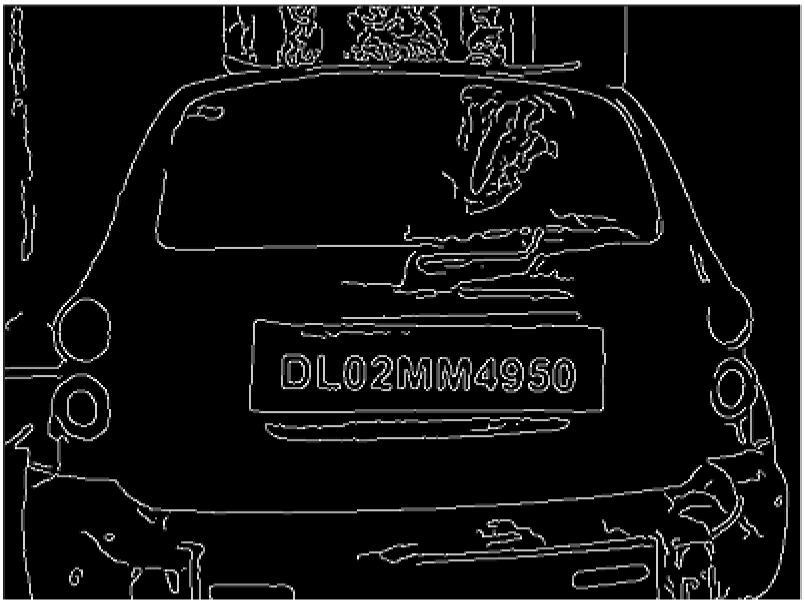


Figure 9.10: Filled Image

9.2.4 Filling Holes

In this algorithm we use imfill function.
imfill(BW,'holes') fills holes in the input binary image BW. In this syntax, a hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image.

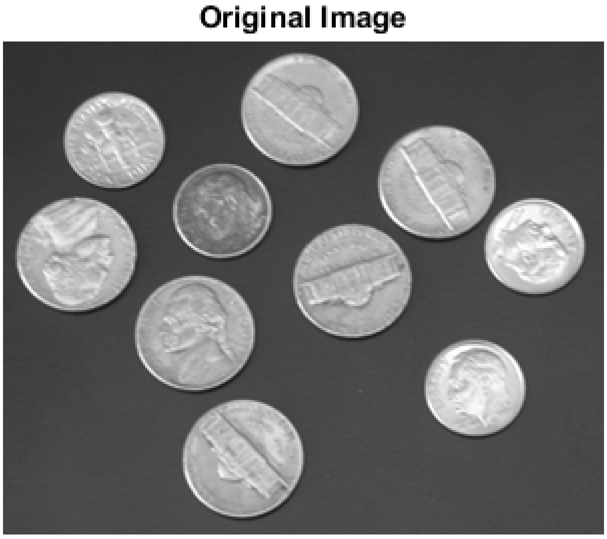


Figure 9.11: Original Image

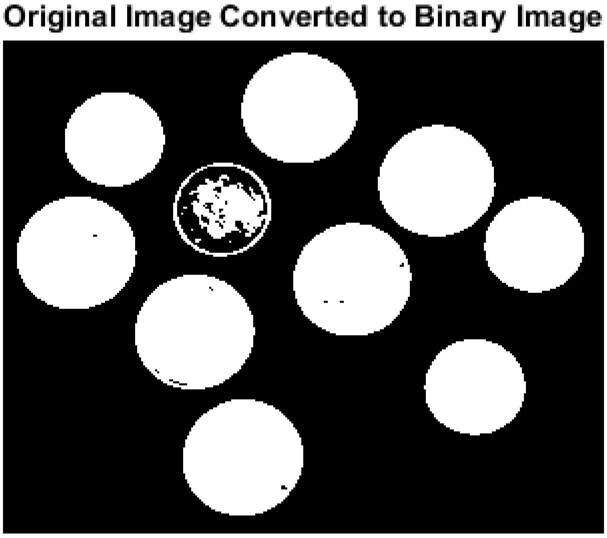


Figure 9.12: Filled Image

9.2.5 Finding the Plate Area

In this Algorithm, we take input the Filled image from which we have to find the plate area. For this we use bwlabel function. bwlabel(BW) returns the label matrix that contains labels for the 8-connected objects found in inputted image. From the label matrix we have to find out the area whose area length is close to the plate area. This can find by finding the biggest area in the label because the plate area is generally the biggest area. After finding the biggest area, black out all the pixels of the image that are not the part of the biggest area. Thus we get plate area as the output in the image.

Original Image

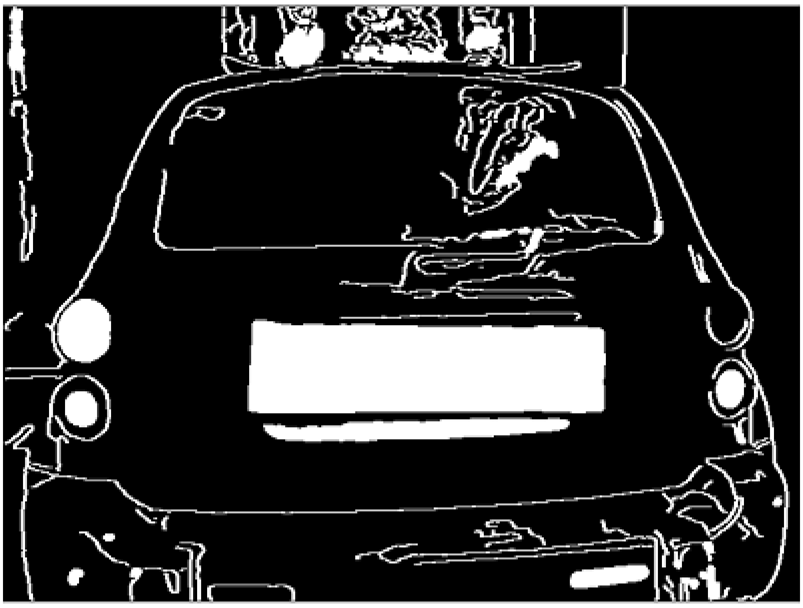


Figure 9.13: Original Image

Image with Plate Area

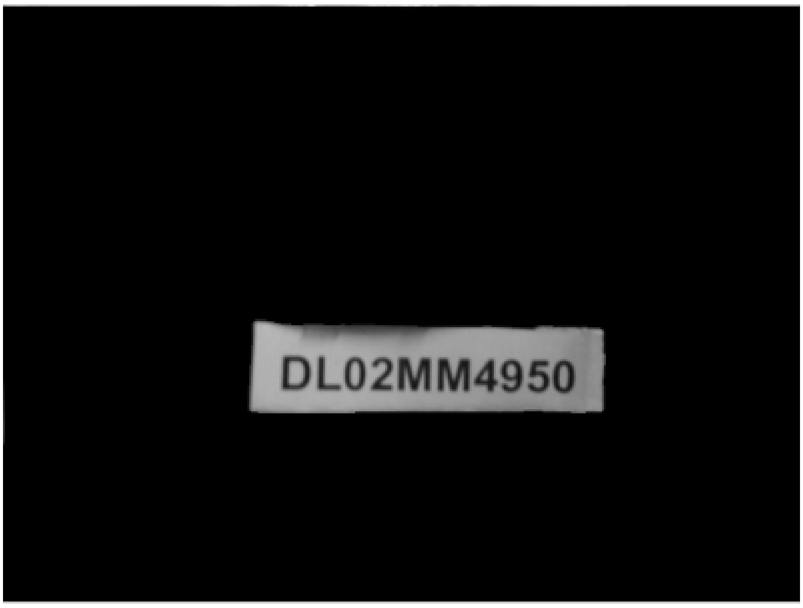


Figure 9.14: Image with plate area

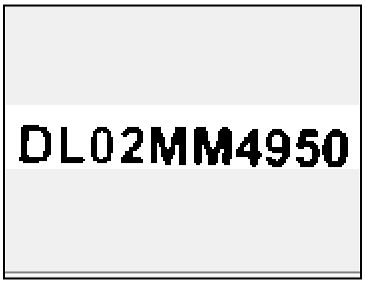
9.2.6 Cropping

In this process the image for more accuracy the plate area is crop for removing the black poriotn of the image.
This is done using imcrop() function in which we specify the starting pixel and the length for cropping.

9.2.7 Optical Character Recognition

Optical character recognition (optical character reader) (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text

OCR Image



DL02MM4950

Figure 9.15: OCR Image

9.2.8 Traffic

This algorithm is used to calculate the traffic at the toll area. In this algorithm first we find what is the current time. Then take all vehicles timing that came at the toll one hour before the timing. Then we calculate the frequency between each value(vehicle timing). depending upon the frequency we decalre whether the traffic is low, medium or high

9.3 Coding

9.3.1 RGB to Gray Conversion

```
I = imread ('path');
figure(1); imshow(I);
Igray = rgb2gray(I);
```

9.3.2 Noise Reduction

In this code BW1 is image to be Noise Reduced

```
Idilate = BW1;
for i = 1:rows
    for j = 2:cols-1
        temp = max(BW1(i,j-1), BW1(i,j));
        Idilate(i,j) = max(temp, BW1(i,j+1));
    end
end
I = Idilate;
```

9.3.3 Edge Detection

```
BW1 = edge(Igray,'canny',0.15);
```

9.3.4 Filling Holes

```
BW2=imfill(I,'holes');
```

9.3.5 Finding the Plate Region

```
L = bwlabel(BW2);
CC(1000)=0;
for i=1:rows
    for j=1:cols
        CC(L(i,j)+1) = CC(L(i,j)+1)+1;
    end
end

big=0;
pos=1;

for i=2:300
    if(CC(i)¿big)
        big=CC(i);
        pos=i-1;
    end
end
```

```
for i=1:rows
    for j=1:cols
        if(L(i,j) =pos)
            L(i,j)=0;
       end
   end
end

for i = 2:cols
    for j = 2:rows
        if(L(j,i) =pos)
            Igray(j,i)=0;
        end
    end
end
```

9.3.6 Cropping the Image

```
for i = 2:cols
    for j = 2:rows
        if(Igray(j,i) =0)
            tempx=i;
            tempy=j;
            break;
        end
    end
    if(Igray(j,i) =0)
        break;
    end
end
x=tempx;
y=tempy;

for i=2:rows
    if(Igray(i,tempx+10) =0)
        if(Igray(j,i) =0)
            tempy=i;
            break;
        end
    end
end

for i=tempx+5:cols
    for j=tempy+5:tempy+70
        if(Igray(j,i) =0)
            sx=i;
            sy=j;
            pos=i;
        end
    end
end

for j=1:rows
    if(Igray(j,pos-10) =0)
        sy=j;
    end
end

cl=sx-x;
ch=sy-y;
a=imcrop(Igray,[x+10 y+5 cl-20 ch]);
```

9.3.7 Optical Character Recognition

```
g=a(:,:,1);
log=roicolor(g,0,100);
imagen =clip(log);
axes(handles.axes3)
imshow(imagen);

if length(size(imagen))==3
   imagen=rgb2gray(imagen);
end

imagen = medfilt2(imagen);
[f c]=size(imagen);
imagen (1,1)=255;
imagen (f,1)=255;
imagen (1,c)=255;
imagen (f,c)=255;
word=[];
re=imagen;
fid = fopen('text.txt', 'wt');

while 1
    [fl re]=lines(re);
     rq=fl;

    while 1
        [fl1 rq]=columns(rq);
        imgn= fl1;
        L = bwlabel(imgn);
        mx=max(max(L));
        BW = edge(double(imgn),'sobel');
        [imx,imy]=size(BW);
```

```
for n=1:mx
    [r,c] = find(L==n);
    rc = [r c];
    [sx sy]=size(rc);
    n1=zeros(imx,imy);
    for i=1:sx
        x1=rc(i,1);
        y1=rc(i,2);
        n1(x1,y1)=255;
    end
    n1= n1;
    n1= clip(n1);
    imgr=samedim(n1);
   letter=genetic(imgr);
   word=[word letter];
end

if isempty(rq)
    fid = fopen(’text.txt’, ’wt’);
    fprintf(fid,’s’,word);
else
    fprintf(fid,’s’,lower(word));
   fprintf(fid,’s’,word);
 end

if isempty(rq)
    break
end
end
```

9.3.8 Size of the Vehicle

```
g=a1(:,:,1);
imtool(g);
log=roicolor(g,210,255);
axes(handles.axes2)
imshow(log);
[lab1,num1]=bwlabel(log,8);
sizeBlob1 = zeros(1,num1);
j = 0;

for i=1:num1,
    sizeblob1(i) = length(find(lab1==i));
end

[maxno largestBlobNo] = max(sizeblob1);
outim = zeros(size(log),’uint8’);
outim(find(lab1==largestBlobNo)) = 1;
```

```
last=255*outim;
vehicle = 0;
sizeblob1(largestBlobNo)

if (sizeblob1(largestBlobNo)¿3000)
    set(handles.txtvehicle,’string’,’Vehicle Type:HMV’);
    vehicle = 80;
else
    set(handles.txtvehicle,’string’,’Vehicle Type:LMV’);
    vehicle = 40;
end
```

9.3.9 Traffic

```
time=rem(now,1);
qwe = xlswrite(’database.xls’,time, ’Sheet1’, tcell);
ctime=rem(now,1);
hour=ctime-0.0416;
j=0;
sumdif=0;
trows=size(C,1)-1;
tempt=cell2mat(C(27,3));
for i=2:trows
    tempt=cell2mat(C(i,3));
    if( (tempt¡ctime)  (tempt¿hour) )
        diff=cell2mat(C(i+1,3))-cell2mat(C(i,3));
        sumdif=sumdif+diff;
        j=j+1;
    end
end
avdif=sumdif/j;
if(avdif¡0.002)
    set(handles.traffic,’string’,’High’);
elseif(avdif¡0.005)
    set(handles.traffic,’string’,’Medium’);
elseif(avdif¿0.005)
    set(handles.traffic,’string’,’Low’);
end
```

9.3.10 Micro-Controller :LCD Interfacing and Motor Driving Interfacing

```
DB4 EQU P1.4
DB5 EQU P1.5
DB6 EQU P1.6
DB7 EQU P1.7
EN EQU P1.2
RS EQU P1.0
RW EQU P1.1
DAT EQU P1
bz EQU P1.3
led EQU p2.7
buzzer EQU p2.0
mdi1 EQU p0.0
mdi2 EQU p0.1

org 0000h
ljmp main1

org 0023h
lcall serial

serial:
jb ti, txs
mov a,sbuf
cjne a,#’Q’,nx1

lcall clockwise
lcall delay
clr ri
reti
nx1: cjne a,#"W",nx2
lcall anticlockwise
lcall delay

clr ri
reti
nx2:
clr mdi1
clr mdi2
setb buzzer
setb led
clr ri
ret

txs: reti

org 0100h
main1:
mov p1,#0ffh
```

```
 mov p0,#00h
 mov ie,#90h ; to enable data receive interrupts
lcall ser ;to initiale uc serial port, 9600,8,n,1
setb p3.0
setb p3.1
setb buzzer
setb led
lcall INIT_LCD ; to initialise lcd
lcall disp1
lcall delay
lcall clear_lcd
lcall delay
ljmp main

main:sjmp main

ser: mov tmod,#20h
mov th1,#0fdh
mov scon,#50h
setb TR1
ret

clockwise:
clr mdi1
setb mdi2
clr led
mov a,#80h
lcall command
mov a,#' '
lcall data_display
mov a,#' '
lcall data_display
mov a,#'S'
lcall data_display
mov a,#'T'
lcall data_display
mov a,#'A'
lcall data_display
mov a,#'T'
lcall data_display
mov a,#'U'
lcall data_display
mov a,#'S'
lcall data_display
mov a,#0c0h
lcall command
mov a,#'C'
lcall data_display
ret
```

```
disp1:
mov a,#80h
lcall command
mov a,#'H'
lcall data_display
mov a,#'E'
lcall data_display
mov a,#'L'
lcall data_display
mov a,#'L'
lcall data_display
mov a,#'L'
lcall data_display
mov a,#'O'
lcall data_display
ret

anticlockwise:
setb mdi1
clr mdi2
clr buzzer
setb led
mov a,#80h
lcall command
mov a,#' '
lcall data_display
mov a,#' '
lcall data_display
mov a,#'S'
lcall data_display
mov a,#'T'
lcall data_display
mov a,#'A'
lcall data_display
mov a,#'T'
lcall data_display
mov a,#'U'
lcall data_display
mov a,#'S'
lcall data_display
mov a,#0c0h
lcall command
mov a,#'A'
lcall data_display
mov a,#'C'
lcall data_display
ret
```

```
INIT_LCD:
CLR RS
CLR RW
CLR EN
SETB EN
mov a,p1
anl a,#0fh
orl a,#20h
mov p1,a
CLR EN
LCALL ready
MOV A,#28h
LCALL WRITE_2_NIBBLES
LCALL ready
MOV A,#0Eh
LCALL WRITE_2_NIBBLES
LCALL ready
MOV A,#06h

LCALL WRITE_2_NIBBLES
LCALL ready
MOV A,#01h
LCALL WRITE_2_NIBBLES ;Write A as two separate nibbles to LCD
LCALL ready
RET
cursor_on:
MOV A,#0eh
LCALL command
ret
cursor_off:
MOV A,#0ch
lcall command
ret
clear_lcd:
MOV A,#01h
LCALL WRITE_2_NIBBLES ;Write A as two separate nibbles to LCD
LCALL ready
ret
command:
LCALL WRITE_2_NIBBLES
clr RS
clr RW
lcall ready
ret
data_display:
setb RS
clr RW
LCALL WRITE_2_NIBBLES
lcall ready
```

```
ret
READ_2_NIBBLES:
ORL DAT,#0F0h ;Be sure to release datalines (set output latches to '1')

SETB EN
MOV A,DAT ;Read first part of the return value (high nibble)

CLR EN
ANL A,#0F0h ;Only high nibble is usable
PUSH ACC
SETB EN
MOV A,DAT ;Read second part of the return value (low nibble)
CLR EN
ANL A,#0F0h ;Only high nibble is usable
SWAP A ;Last received is actually low nibble, so put it in place
MOV R3,A
POP ACC
ORL A,R3 ;And combine it with low nibble
RET
WRITE_2_NIBBLES:
mov r3,a
anl a,#0f0h
mov r4,a
mov a,p1
anl a,#0fh
orl a,r4
mov p1, a
SETB EN
CLR EN
mov a,r3
swap a
anl a,#0f0h
mov r4,a
mov a,p1
anl a,#0fh
orl a,r4
mov p1, a
SETB EN
CLR EN
RET
ready:
CLR RS ;It's a command
SETB RW ;It's a read command
LCALL READ_2_NIBBLES ;Take two nibbles from LCD in A
JB ACC.7,ready ;If bit 7 high, LCD still busy

CLR RW ;Turn off RW for future commands
RET
```

```
delay: mov tmod,#21h
mov r1,#14
back2: mov tl0,#00h
mov th0,#00h
setb tr0
again2: jnb tf0,again2
clr tr0
clr tf0
djnz r1,back2
ret
END
```

9.3.11 Website Code

Session Page

```
<?php
ob_start();
session_save_path('tmp/');
@session_start();
$filename=$_SERVER['SCRIPT_NAME'];
$referer=@$_SERVER['HTTP_REFERER'];
?>
```

Index Page

```
<?php
ini_set("display_errors",1);
require_once 'excel_reader2.php';
require_once 'db.php';
$data = new Spreadsheet_Excel_Reader("database.xls");
    if(@count($data->sheets[2][cells])>0) // checking sheet not empty
    {
        @$datatable[@count($data->sheets[2][cells])][2];
        for($j=1;$j<=@count($data->sheets[2][cells]);$j++) // loop used to get each row of
        {
            for($k=1;$k<=@count($data->sheets[2][cells][$j]);$k++) // This loop is created
            {
                $datatable[$j][$k]=@$data->sheets[2][cells][$j][$k];
            }
        }
    }

for($i=1;$i<=6;$i++)
{
    for($j=1;$j<=3;$j++)
    {
         //echo $datatable[$i][$j].’<br>’;
    }
}
?>
```

Login Page

```
<?php
    include(’core.php’);
    if(isset($_SESSION[’user_id’]) && !empty($_SESSION[’user_id’]))
    {
        $url="prof.php?id=".$_SESSION[’user_id’];
        echo ’<script type="text/javascript">’;
        echo ’window.location.href="’.$url.’";’;
        echo ’</script>’;
    }

    if(isset($_POST[’plate’]) && isset($_POST[’password’]))
    {
        echo $plate=$_POST[’plate’];
        echo $password=$_POST[’password’];
        $found=0;
        if(!empty($plate) && !empty($password))
        {
            ini_set("display_errors",1);
            require_once ’excel_reader2.php’;
            require_once ’db.php’;
            $data = new Spreadsheet_Excel_Reader("database.xls");
                for($j=2;$j<=@count($data->sheets[2][cells]);$j++)
                {
                    if(@$data->sheets[2][cells][$j][1]==$plate)
                    {
                        if(@$data->sheets[2][cells][$j][4]==$password)
                        {
                            $found=1;
                            $_SESSION[’user_id’]=$data->sheets[2][cells][$j][3];
                            $url="prof.php?id=".$data->sheets[2][cells][$j][3];
                            echo ’<script type="text/javascript">’;
                            echo ’window.location.href="’.$url.’";’;
                            echo ’</script>’;
                            break;
                        }
                    }
                }

                    if($found==0)
                    {
                        echo ’Invalid Password’;
                    }
        }
    }
?>
```

Logout Page

```
<?php
require ’core.php’;
unset($_SESSION[’user_id’]);
header(’Location:login.php’);
?>
```

Registration Page

```
<?php
    include ’core.php’;
    if(isset($_SESSION[’user_id’]) && !empty($_SESSION[’user_id’]))
    {
        $url="prof.php?id=".$_SESSION[’user_id’];
        echo ’<script type="text/javascript">’;
        echo ’window.location.href="’.$url.’";’;
        echo ’</script>’;
    }

    if(isset($_POST[’name’]) && isset($_POST[’cno’]) && isset($_POST[’npass’]) && isset($_
    isset($_POST[’bal’]))
    {
        include ’index.php’;
        echo $count=@count($data->sheets[2][cells])+1;
        $name=$_POST[’name’];
        $cno=$_POST[’cno’];
        $npass=$_POST[’npass’];
        $pass=$_POST[’pass’];
        $bal=$_POST[’bal’];
            if(!empty($name) && !empty($cno) && !empty($pass) && !empty($npass) &&
            !empty($bal))
            {
                if($npass==$pass)
                {
                    $url="regtest.php?name=".$name."&cno=".$cno."&pass=".$pass.
                    "&count=".$count."&bal=".$bal;
                    echo ’<script type="text/javascript">
                    echo ’window.location.href="’.$url.’";’;
                    echo ’</script>’;
                    }
                    else
                    {
                        echo ’<center><b style="color:red;position:absolute;top:670px;
                        left:590px;z-index:1;">Password Doesn\’t Match</b></center>’;
```

```
          }
      }
      else
      {
        echo '<center><b style="color:red;position:absolute;top:670px;left:590px;
        z-index:1;">Enter all Details</b></center>';
      }

  }
  else
  {
    echo '<center><b style="color:red;position:absolute;top:670px;left:590px;
    z-index:1;">Enter all Details</b></center>';
  }
?>
```

Stolen Vehicle Registration Page

```
<?php
  include 'core.php';
  if(isset($_SESSION['user_id']) && !empty($_SESSION['user_id']))
  {
  }
  else
  {
   $url="login.php";
   echo '<script type="text/javascript">';
   echo 'window.location.href="'.$url.'";';
   echo '</script>';
  }

  $id=$_SESSION['user_id'];
  $pos=0;
  include('index.php');
  for($i=2;$i<=@count($data->sheets[2][cells]);$i++)
  {
    if($datatable[$i][3]==$id)
    {
      $pos=$i;
    }
  }

  if(@$_POST['yon']=="Y")
  {
    $count=@count($data->sheets[1][cells])+1;
    $url="test.php?id=".$datatable[$pos][1]."&count=".$count;
    echo '<script type="text/javascript">';
    echo 'window.location.href="'.$url.'";';
    echo '</script>';
    }

?>
```

Stolen Vehicle identification

```
<?php
$stat="Not Stolen";
    if(@count($data->sheets[1][cells])>1)
    {
       for($j=2;$j<=@count($data->sheets[1][cells]);$j++)
       {
            if(@$data->sheets[1][cells][$j][1]==$datatable[$pos][1])
            {
                $stat="Stolen";
                break;
            }
       }
    }
?>
```

User Profile Page

```
<?php
include('core.php');
 if(@isset($_SESSION['user_id']) && @!empty($_SESSION['user_id']))
 {
 //echo $_SESSION['user_id'];
   if($_SESSION['user_id']!=$_GET['id'])
{
$url="prof.php?id=".$_SESSION['user_id'];
echo $_SESSION['user_id'];
echo '<script type="text/javascript">';
echo 'window.location.href="'.$url.'";';
echo '</script>';
}
 }else
 {

$url="login.php";
echo $_SESSION['user_id'];
echo '<script type="text/javascript">';
echo 'window.location.href="'.$url.'";';
echo '</script>';

 }


?>
```

```
<html>
<head>
<title>
ANPR
</title>
<style type="text/css">
.link:hover{
background-color:#EEEEEE;
}
</style>
</head>
<body style="margin:0px;font-family:arial;">
<div style="position:absolute;width:100%;height:170px;background-color:#001E3C;">
<br></br>
<table style="position:absolute;left:10px;" cellpadding="0" cellspacing="0">
<tr>
<td>
<img src="logo.jpg" height="80px" width="200px" style="padding-left:20px;">
</td>
<td><center><h1 style="color:white;font-size:45px;font-family:Arial;margin:20px;">Automatic
</td></tr></table>
</div>
<div style="position:absolute;top:200px;left:10px;">
<table style="border:1px solid silver;width:250px;"  cellpadding="15px" cellspacing="0px">
<tr class="link">
<td>
<img src="padlock.png" style="position:absolute;left:10px;top:7px;">
</td>
<td>
<b><a href="login.php" style="text-decoration:none;color:#001E3C;">Login Page</a></b>
</td>
</tr>
<tr class="link">
<td>
<img src="note.png" style="position:absolute;left:10px;top:58px;">
</td>
<td>
<b><a href="prof.php"  style="text-decoration:none;color:#001E3C;">User Profile</a></b>
</td>
</tr>
    <tr class="link">
<td>
<img src="spy.png" style="position:absolute;left:10px;top:105px;">
</td>
<td>
<b><a href="sreg.php"  style="text-decoration:none;color:#001E3C;">Stolen Registration</a><
</td>
</tr>
    <tr class="link">
```

```
<td>
<img src="new-file.png" style="position:absolute;left:10px;top:155px;">
</td>
<td>
<b><a href="reg.php"  style="text-decoration:none;color:#001E3C;">Car Registration</a></b>
</td>
</tr>
</table>
</div>
<div style="position:absolute;top:200px;left:270px;background-color:#EEEEEE;width:60%;heigh
<br>
<img src="note.png" style="position:absolute;left:10px;top:15px;">&nbsp &nbsp &nbsp
<b style="color:#001E3C;font-size:24px;font-weight:bold;">&nbsp &nbsp ANPR System User Prof
<hr color="silver" width="95%">
<?php
  $id=$_GET['id'];
  $pos=0;
  include('index.php');
  for($i=2;$i<=@count($data->sheets[2][cells]);$i++)
  {
  if($datatable[$i][3]==$id)
  {
  $pos=$i;
  }
  }
  include('stol.php');


?>
<div style="width:90%;height:400px;background-color:white;margin-left:39px;">
<form>
<table style="position:relative;top:50px;left:50px;border:1px solid silver;width:600px;" bo
<tr>
<th colspan="3" style="background-color:#001E3C;color:white;">
Car Profile
</th>
</tr>
<tr>
<td rowspan="4">
     <img src="users/<?php echo $id;?>.jpg" height="250px" width="250px" style="border:1px
</td>
</tr>
<tr>
<td>
     <b>License No. -:</b>
</td>
<td>
<?php echo $datatable[$pos][1]?>
</td>
```

```
</tr>
<tr>
<td >
<b>Balance -: </b>
</td>
<td>
<?php echo $datatable[$pos][2]?>
</td>
</tr>
<tr>
<td >
<b>Vehicle Status  -:</b>
</td>
<td>
<?php
echo $stat;
?>
</td>
</tr>
</table>
</form>
</div>
</div>
<div style="position:absolute;top:200px;right:10px;">
<table style="border:1px solid silver;width:200px;"  cellpadding="15px" cellspacing="0">
<tr>
<th style="background-color:#001E3C;">
<b><b href="login.php" style="text-decoration:none;color:white;padding:0px;">User Detail</b
</th>
</tr>
<tr >
<td>
<center><font href="prof.php"  style="text-decoration:none;color:#001E3C;">
<?php
if(isset($_SESSION['user_id']) && !empty($_SESSION['user_id']))
echo "You're Logged In";
?>
</font></center>
<br><br>
<center><b><a href="logout.php" style="color:white;background-color:#001E3C;padding:10px;te
<br>
</td>
</tr>
</table>
</div>
</body>
</html>
```

9.3.12 Interface Coding

```
function varargout = code(varargin)
% CODE M-file for code.fig
%      CODE, by itself, creates a new CODE or raises the existing
%      singleton*.
%
%      H = CODE returns the handle to a new CODE or the handle to
%      the existing singleton*.
%
%      CODE('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in CODE.M with the given input arguments.
%
%      CODE('Property','Value',...) creates a new CODE or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before code_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to code_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help code

% Last Modified by GUIDE v2.5 19-Feb-2016 19:14:18

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @code_OpeningFcn, ...
                   'gui_OutputFcn',  @code_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before code is made visible.
function code_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to code (see VARARGIN)

% Choose default command line output for code
handles.output = hObject;
clc
ser = serial('COM4');
% Update handles structure
handles.ser = ser;
guidata(hObject, handles);

% UIWAIT makes code wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = code_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in cmd_start.
function cmd_start_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.output = hObject;

ser = handles.ser;
set(handles.txt_break,'string','--');
set(handles.pbal,'string','--');
set(handles.adud,'string','--');
set(handles.cbal,'string','--');
set(handles.oname,'string','--');
set(handles.txt_plate,'string','--');
set(handles.traffic,'string','--');
set(handles.txt_stolen,'string','--');
set(handles.txt_vehicle,'string','--');
```

Testing

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete.

Testing Types :

- Uniting Testing
- Integration Testing
- Validation Testing
- System Testing

10.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design (i.e.) the module. Unit testing exercise specific paths in a modules control structure to ensure complete coverage and maximum error detection .This test focus on each module individually , ensure that it functions properly as a unit. Hence, the name is unit testing. During unit testing some errors were raised and all of them were rectified and handled well. The result was quiet satisfactory and it worked well.

10.2 Integration Testing

Integration testing address the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of High-order tests are conducted. The main objective in this testing process is to take unit tested modules and build a program structure that has been dictated by design.
The following are the types of Integration Testing:

10.2.1 Top-Down Integration:

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first of breadth-first manner.

10.2.2 Bottom-Up integration:

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is elimination. The bottom-up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters that perform a specific software sub-function.
- A drive (i.e.,), the control program for testing is written to co-ordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure.

10.3 Validation Testing

At the end of the Integration Testing, software is completely assembled as a package, interfacing errors have been uncovered and correction testing begins.

Validation Test Criteria:
Software testing and validation is achieved through series of black box tests that demonstrate conformity with the requirements. A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both, the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are meet.

10.4 System Testing

System testing is series of different tests whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all the work should verify that all system element have been properly integrated and perform allocated functions.

10.4.1 Recovery Testing

It is a system that forces the software to fail in a variety of ways and verifies that the recovery is properly performed.

10.4.2 Security Testing

It attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration. The systems security must of course be tested from in vulnerability form frontal attack.

10.4.3 Stress Testing

Stress tools are designed to confront programs with abnormal situations. Stress testing executes a system in a manner that demands resources in abnormal quantity and volume.

10.4.4 Regression Testing

Regression testing is a type of software testing that verifies that software that was previously developed and tested still performs correctly after it was changed or interfaced with other software. Changes may include software enhancements, patches, configuration changes, etc. During regression testing new software bugs or regressions may be uncovered.

10.5 White Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to de- sign test cases.

10.6 Black box Testing

Black-box testing, also called behavioral testing, focuses on the functional requirements of the software. That is, black-box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. It examines the functionality of an application without peering into its internal structures or workings. Black-box testing finds errors in the following categories: incorrect or missing functions, interface errors, access, behavior or performance errors, initialization and termination errors.

Working Example

Following is the Working of the ANPR System

1. When the Car enters the Toll Area, there are two cameras to click the picture of the car

   Camera 1: It is located at the top of the toll which takes picture of the car from the top view to detect its size

   Eg:

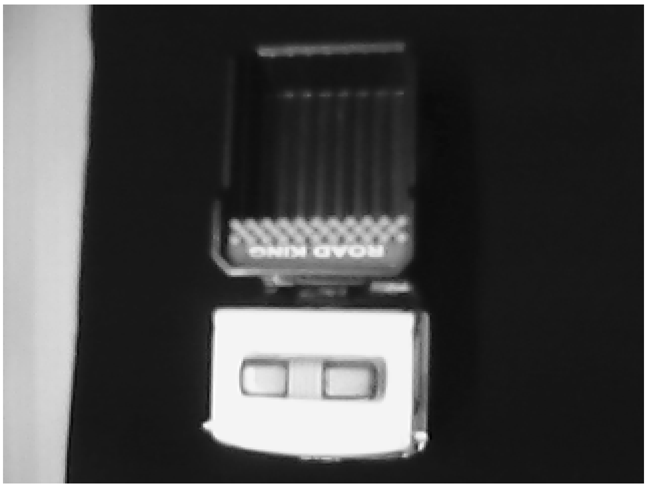


Figure 11.1: Camera 1

Camera 2: It is located behind the car fixed inside the speed breaker which takes picture of car from behind for number plate extraction process

Eg:

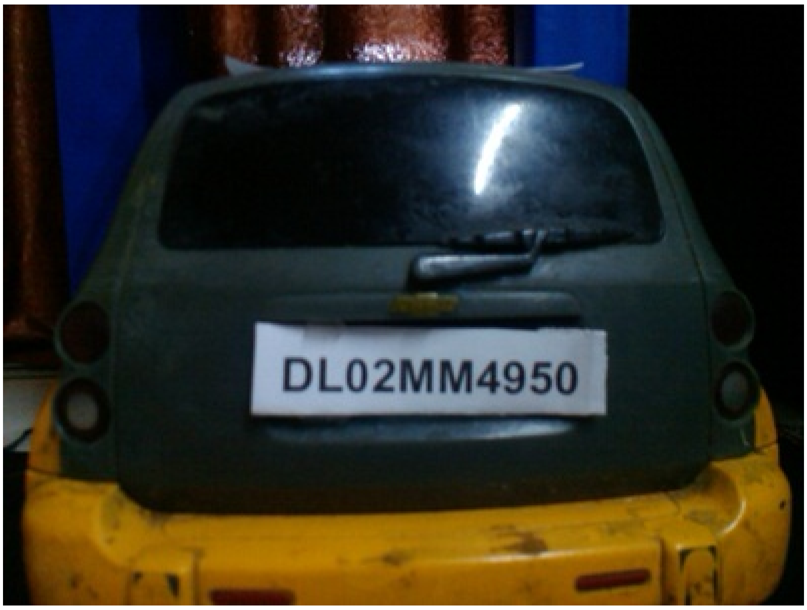


Figure 11.2: Camera 2

2. After Taking the picture, the manipulation process is conducted were different image processing algorithms are applied described in the algorithm section. So following are the images of the result of the whole process

Edge Detection

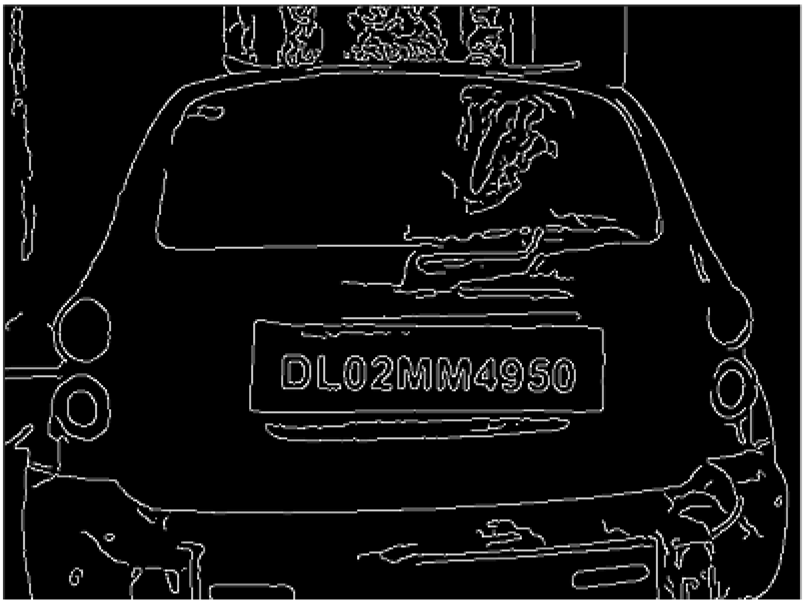


Figure 11.3: Canny Edge Detection

Filling Holes

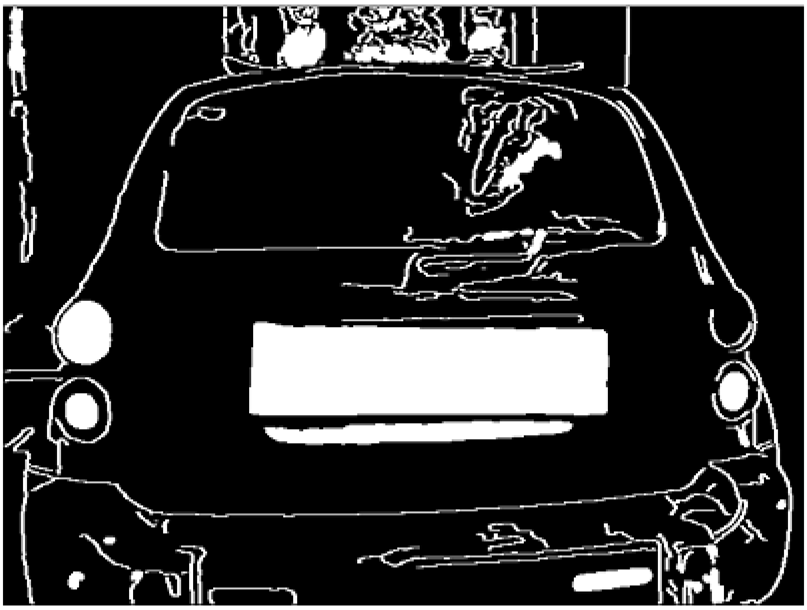


Figure 11.4: Filling Holes using imfill()

Finding the Plate Location

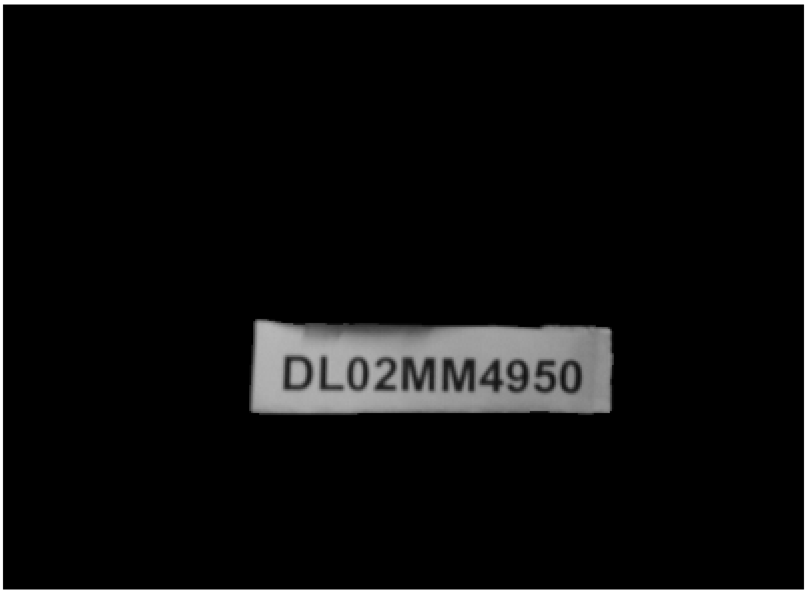


Figure 11.5: Finding the Plate Location

Cropping the plate and Performing OCR

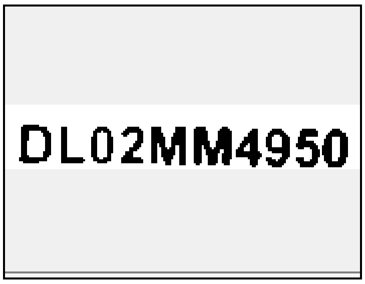
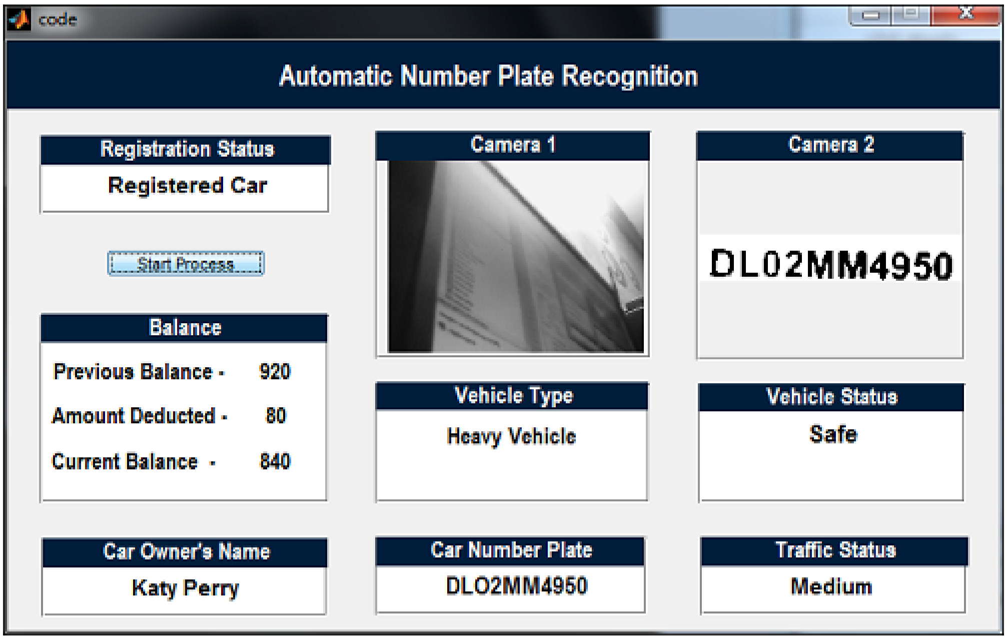


Figure 11.6: Cropping the plate and Performing OCR

Final Result



Automatic Number Plate Recognition

Registration Status: Registered Car

Balance

Previous Balance - 920

Amount Deducted - 80

Current Balance - 840

Vehicle Type: Heavy Vehicle

Vehicle Status: Safe

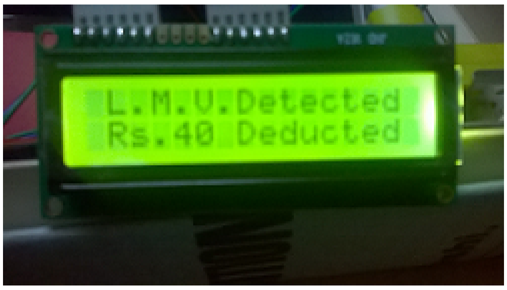
Car Owner's Name: Katy Perry

Car Number Plate: DLO2MM4950

Traffic Status: Medium

Figure 11.7: Final Result

Final Result LCD



L.M.V.Detected
Rs.40 Deducted

Figure 11.8: Final Result LCD

Conclusion

- The proposed system will help in reducing the human intervention in the traffic System.
- The purpose of this project is to maintain traffic disciplines.
- The ANPR system plays an important role in detecting security threat.

Future Work

- To work on images that contains more noise.
- To work with the regional Indian languages like Hindi, Marathi, Bengali...etc.
- To extract the number plate from fast moving vehicle.
- If Image processing algorithms improves further than along with number plate recognition we can also perform face detection to detect the face of any suspicious criminal.
- It can also detect whether the driver is using the seat belt or not

Reference

- Muhammad Tahir Qadri, Muhammad Asif, Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition IEEE 2009
- OptasiaSystem Pte Ltd, The World Leader in License Plate Recognitioz Technology Source from: www.singaporegateway.com
- MATLAB official Website: in.mathworks.com/help
- Wikipedia: www.wikipedia.org