PAGE INDEX

CHAPTER 1

INTRODUCTION

As the name suggests "Semi-Automatic Toll Plaza" the key theme of our project is the semi-automation. So here we will just take the over look of what is mean by Semi-Automation. So in very simple language the Semi-Automation means to replace the human being from the process with the machines upto a certain extent. Before moving further we will just take the overlook of history of the toll plazas. In early days toll booths means there are two people for opening & closing of the gate & another two are for reception of the money & data keeping etc.But by using semi automatic toll plazas data is stored in computers only two personals are required for single booth.

1.1 MySQL and Relational DBMS

A Database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Nowadays, we use relational database management systems (RDBMS) to store andmanage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys. Relational Database Management System (RDBMS) is a software that-Enables you to implement a database with tables, columns and indexes.Guarantees the Referential Integrity between rows of various tables.Updates the indexes automatically.

1.2 PHP

PHP (PHP: Hypertext Preprocessor) is a scripting language that helps people make web pages more interactive by allowing them to do more things. Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something". The PHP code is enclosed in special start and end processing instructions <? php and?> that allow you to jump into and out of "PHP mode." What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP.A website programmed with PHP can have pages that are password protected. A website with no programming cannot do this without other complex things.

1.3 HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

1.4 XAMPP

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows.
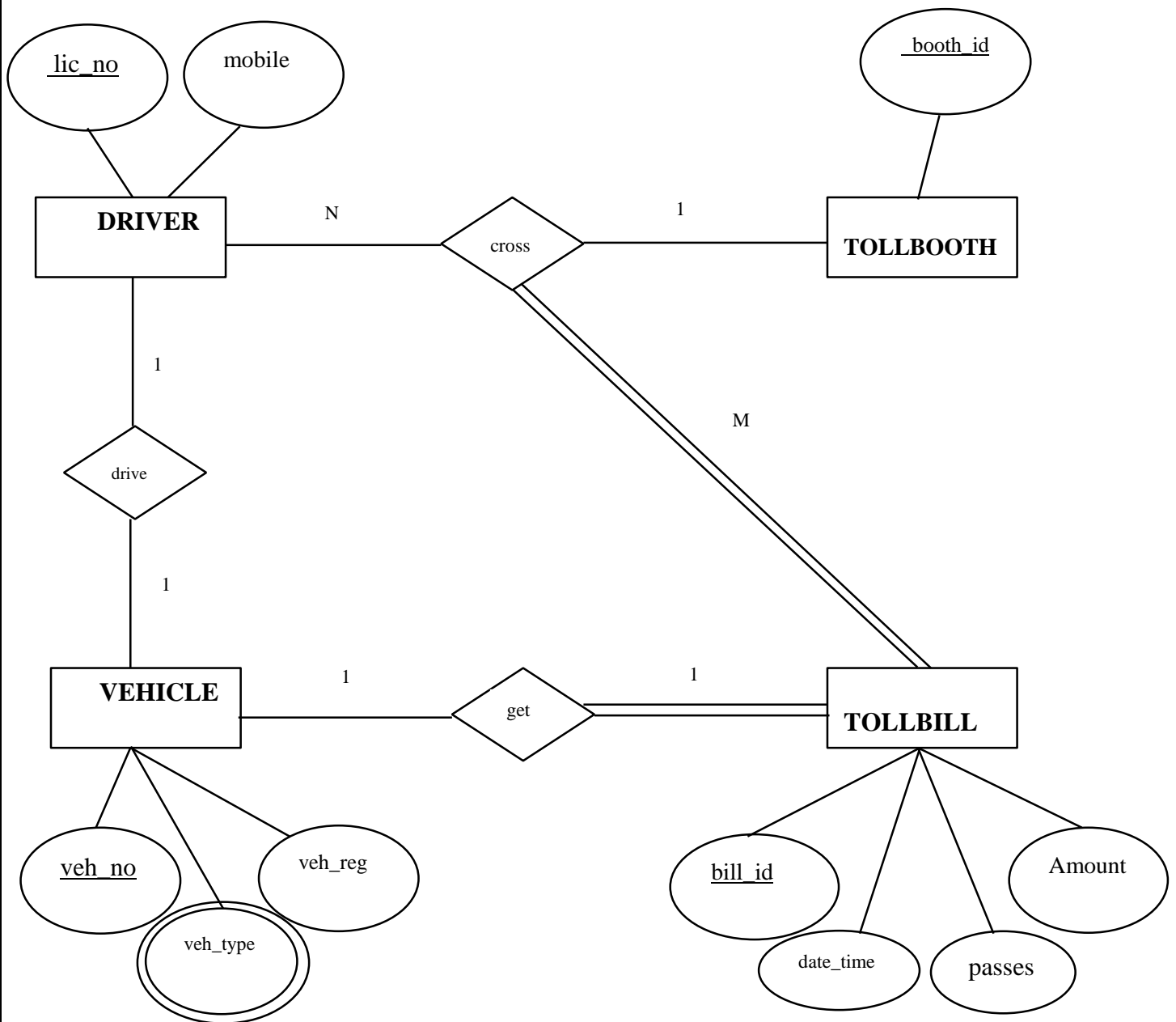
CHAPTER 2

DESIGN

2.1 ER-diagram:

Fig 2.1 ER-diagram

2.2 RELATIONAL SCHEMA (E-R TO RELATIONAL SCHEMA):

STEP 1: MAPPING OF REGULAR ENTITY TYPES

VEHICLE

veh_no	veh_state	veh_type	lic_no	date_time

DRIVER

lic_no	mobile	veh_no

TOLLBOOTH

booth_id	email

crosses

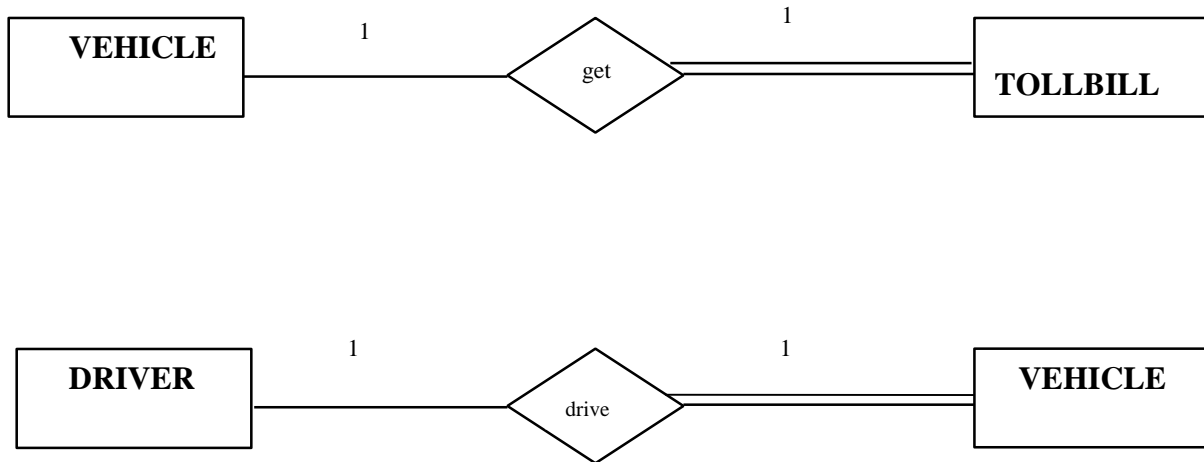| lic_no	booth_id	bill_id

Fig 2.2.a Regular entity

STEP 2: MAPPING OF WEAK ENTITY TYPES

TOLLBILL

bill_id	passes	amount	veh_no	lic_no	date_time

Fig 2.2.b Weak entity

STEP 3: MAPPING OF BINARY ENTITY TYPES

Fig 2.2.c Binary entity

STEP 4: MAPPING OF 1: N RELATION TYPES

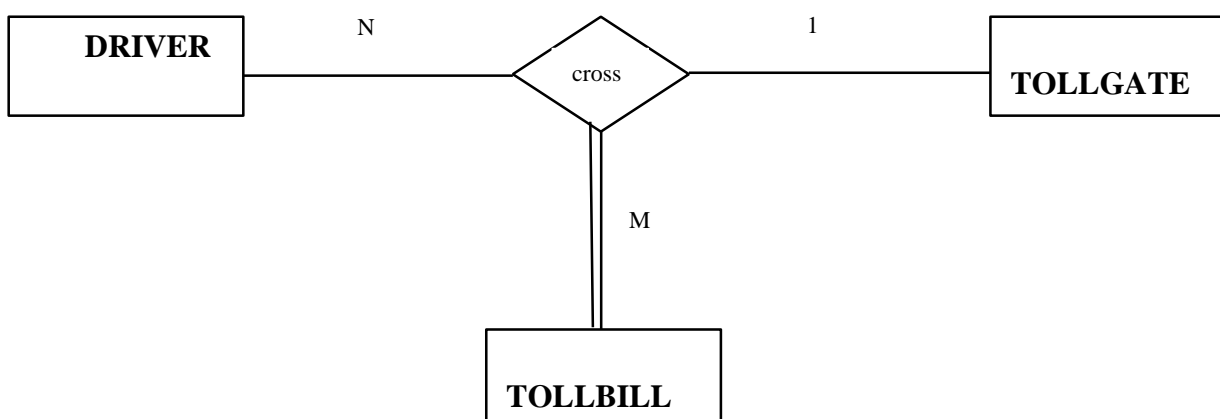The ERD of our project does not contain any 1:N relation types.

STEP 5: MAPPING OF M: N RELATION TYPES

The ERD of our project does not contain any M:N relation types.

STEP 6: MAPPING OF MULTIVALUED ATTRIBUTES

VEHICLE_TYPE

type_id	price

Fig 2.2.f Multivalued relationship

STEP 7: MAPPING OF N-ARY RELATION TYPES

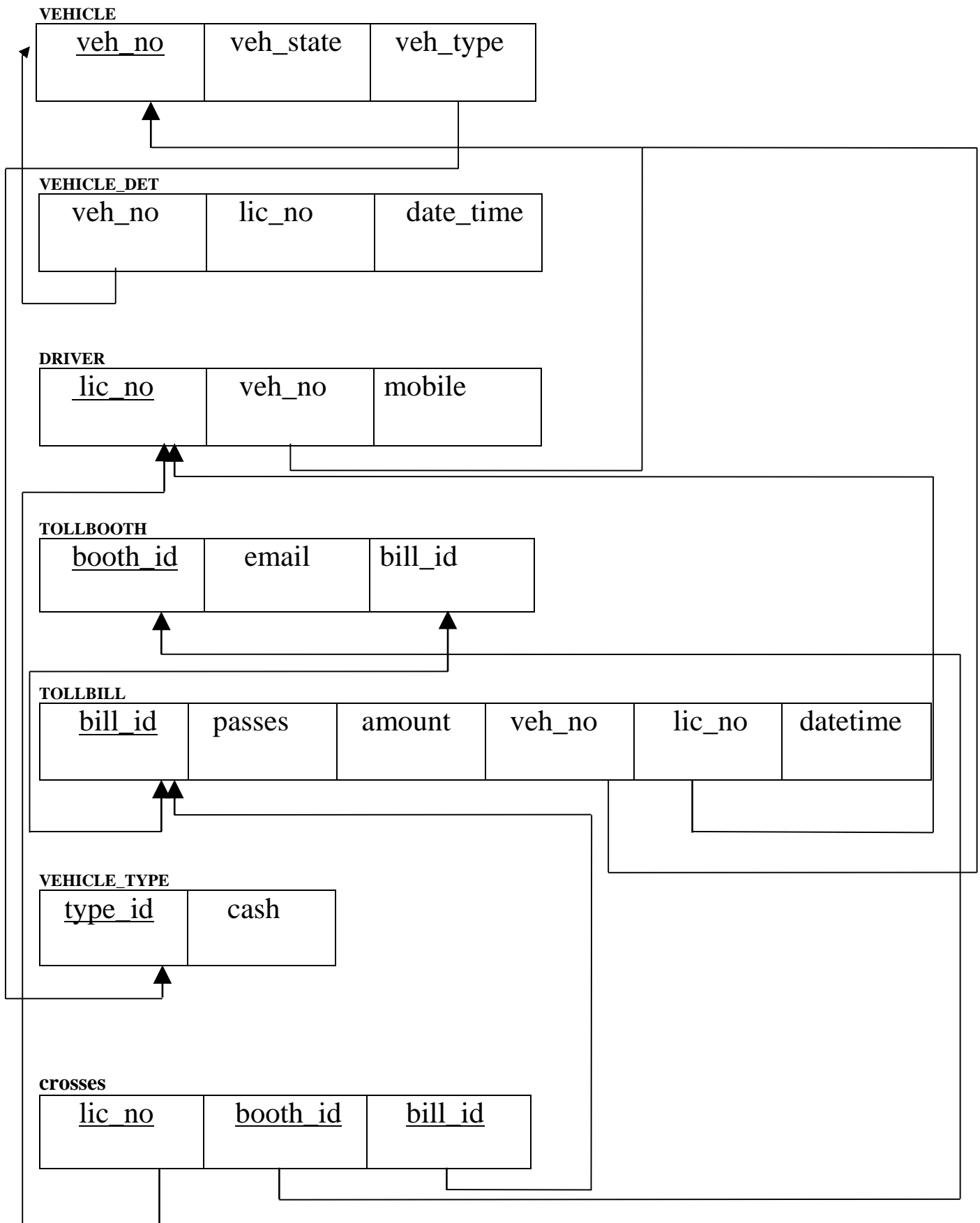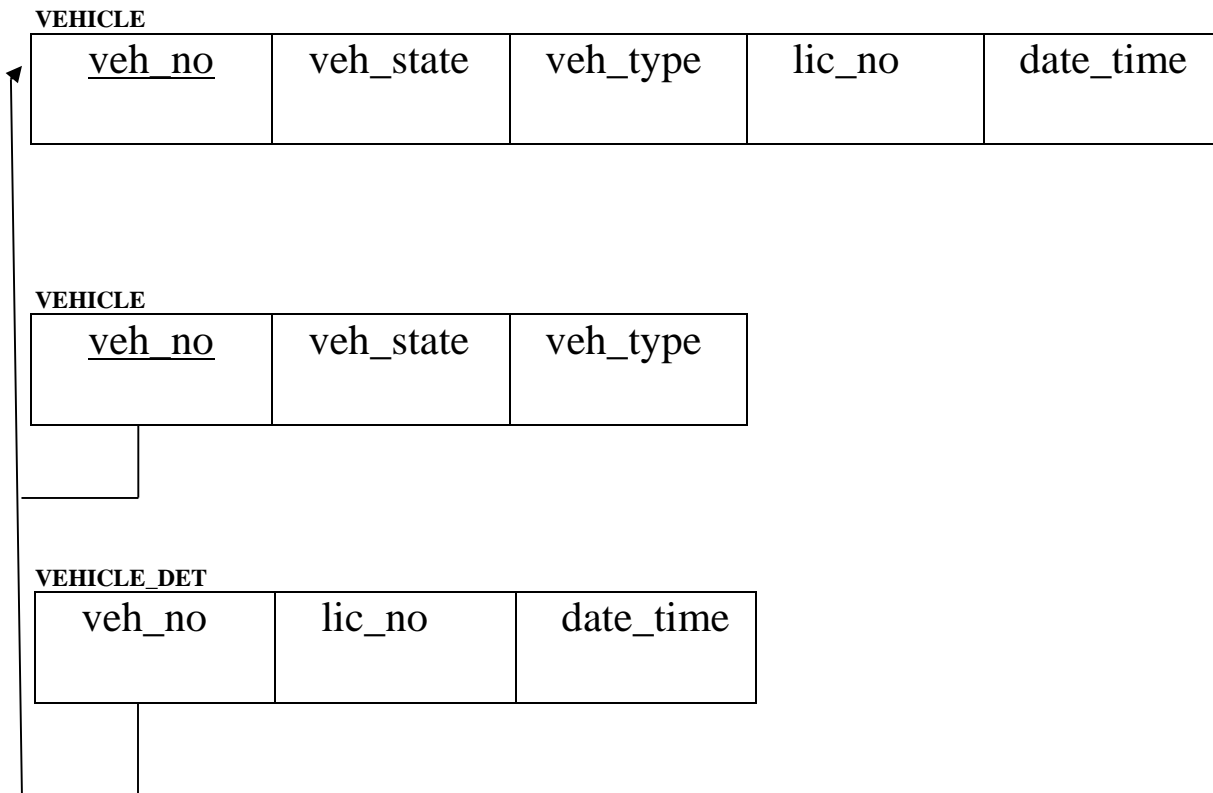**Fig 2.2.g N-ary relationship**

2.3 SCHEMA DIAGRAM:

VEHICLE

veh_no	veh_state	veh_type

VEHICLE_DET

veh_no	lic_no	date_time

DRIVER

lic_no	veh_no	mobile

TOLLBOOTH

booth_id	email	bill_id

TOLLBILL

bill_id	passes	amount	veh_no	lic_no	datetime

VEHICLE_TYPE

type_id	cash

crosses

lic_no	booth_id	bill_id

Fig 2.3 Schema diagram

2.4 NORMALIZATION:

From the table vehicle, we have the following FD:

VEHICLE

veh_no	veh_state	veh_type	lic_no	date_time

VEHICLE

veh_no	veh_state	veh_type

VEHICLE_DET

veh_no	lic_no	date_time

Fig 2.4 Normalisation

The above relation is in 1NF and also all the nonprime attribute (veh_state, veh_type ,lic_no, date_time) are fully functionally dependent on the prime attribute (ID). Hence it satisfies 2 NF.
The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key(ID). Hence it satisfies 3 NF.

CHAPTER 3

IMPLEMENTATION

COMPUTERIZED TOLL COLLECTION SYSTEM: SOFTWARE

System Requirements

Operating System: Windows XP/7/8/10 / MacOS / Linux (any distro).

Memory: Minimum of 256Mb of RAM, Minimum of 1GB harddisk space.

Software requirement: MySQL server 5.0 or above, php Web Browser (any).

3.1 Login

In proposed system, there is a provision of login for administrator and booth, admin and booths have to enter their ID and Password before entering to the database. If all the details match, then operator can login to the toll system. If a new booth has to register into the database, then it has to fill the registration form and need to login again.

3.2 Toll Operating Software

When vehicle enter in the toll plaza firstly the vehicle license plate number has to be entered along with other details. After filling the complete form, the toll operator submits the data to the database. This action will save all the information regarding the vehicle into the database and prints the bill. This data can be retrieved from the database by the authorized persons whenever needed.

Database of toll system is divided into mainly two parts:

1. Database Admin

2. Booth database

The admin database contains details of all toll booths. The booth database includes all registered vehicles with details like vehicle number, driver license number, date and time of passing etc. Both the database admin and booth database is connected to a centralized database where all the data is saved. All toll-booths records stored at central server and these records could be seen and printed by day, date, month, and year. All these records are maintained at corresponding toll-booth.

3.3 Table structure

ADMIN

Name	Type
admin_name	Varchar(20)
email	Varchar(20)
password	Varchar(10)

TOLLBOOTH

Name	Type
booth_name	Varchar(20)
email	Varchar(20)
password	Varchar(10)

VEHICLE

Name	Type
vehno	Varchar(20)
vehtype	Varchar(20)
vehreg	Varchar(20)
date_time	date

DRIVER

Name	Type
licno	Varchar(20)
vehno	Varchar(20)
mobile	Int(10)

TOLLBILL

Name	Type
billid	Varchar(20)
vehno	Varchar(20)
licno	Varchar(20)
passes	Varchar(20)
amount	Int(10)
date_time	date

VEHICLE_TYPE

Name	Type
type_id	Varchar(20)
price	Int(10)

The front end is done using HTML and the back end is done using MySQL. The connection between the HTML and MySQL is done using the following syntax:

mysql_select_db("tollplazadb");

Where db refers to the database which is tollplazadb. The booth as well as the vehicle details can be added, modified and deleted .

INSERT

Insert operation is used for booth which is not already registered and is new to the system. Also, when a vehicle detail needs to be entered into database insertion operation takes place.

UPDATE

Update operation is used to update the price of the vehicle types. This can be done only by admin.

DELETE

Delete operation is used when a booth has to be removed from the toll plaza. Also, admin can use this operation to delete any data from the database.

TRIGGER

Trigger is given to update a new table whenever a new booth registers to the system. Boothid, date and time is updated by this trigger.

```
CREATE TRIGGER upd_check BEFORE UPDATE ON vehicletype
    FOR EACH ROW
    BEGIN
      IF NEW.price < 20 THEN
        SET NEW.price = 30;
      ELSEIF NEW.price > 100 THEN
        SET NEW.price = 110;
      END IF;
    END;
```

PROCEDURE

Procedure is used to count the total number of booths who have come to perform a transaction.

The count is incremented by one whenever a new booth comes and login/signup to the system.

The command to accomplish trigger is as follows:

CREATE DEFINER= 'root' @ 'localhost' PROCEDURE 'total' (OUT 'counts' INT) NO SQL

SELECT count(*) INTO counts FROM booth;

CHAPTER 4

RESULTS

If we implement this method on toll plazas having traditional manual toll collecting method the whole time duration for payment of toll or retrieval of data will be less. We can also get vehicle information at customer account registration time and when vehicles crosses booth.These records can be seen by using driver name or vehicle number. This computarized toll plaza system will also results in less usage of man power, fuel saving, reduced complexity in storing and retrieving data etc.

4.1 Login page

There is a login page for both admin and the booth.
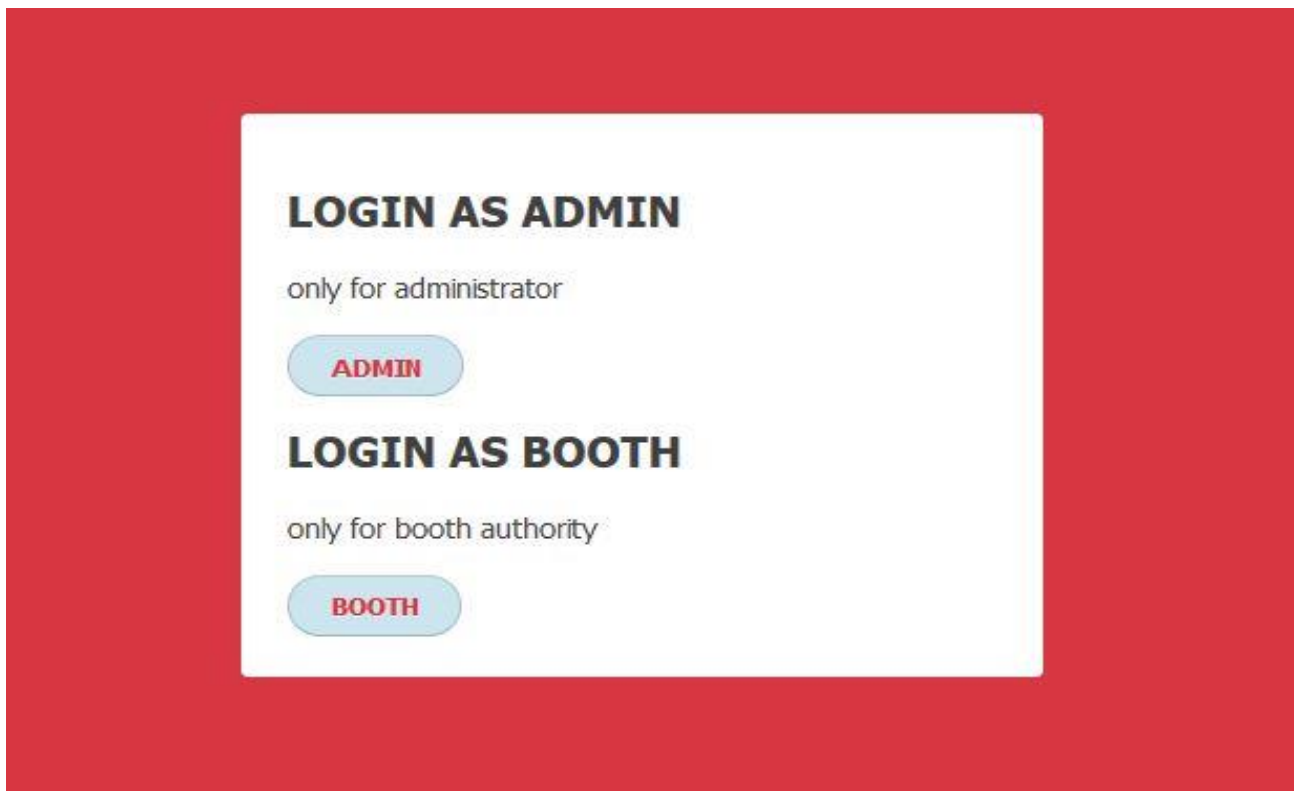
Fig 4.1.a Login page
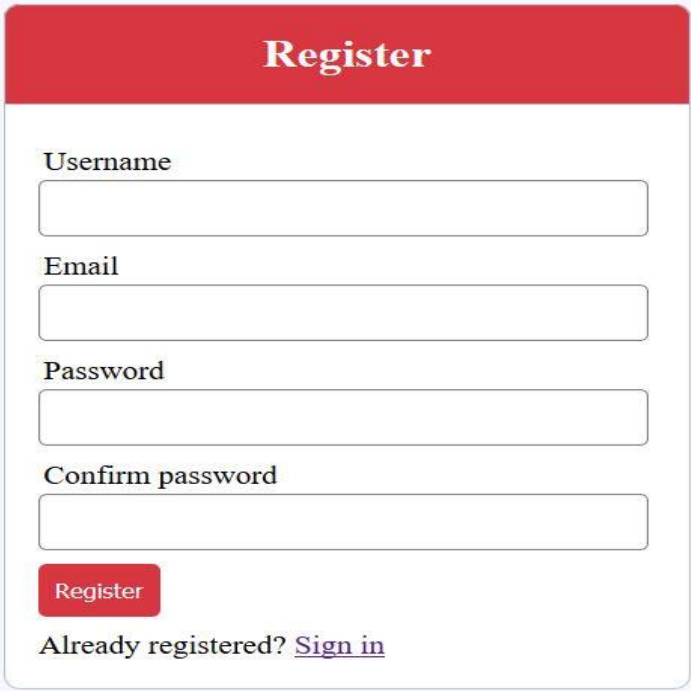
Enter the correct username and password to continue.

Fig 4.1.b Login booth

New booth can register to the Database. Without logging in one can't move forward.

Fig 4.1.c Registration booth

4.2 Home page

Admin home page is provided with options for both viewing and updating different vehicle prices.
But booth home page has only viewing option.
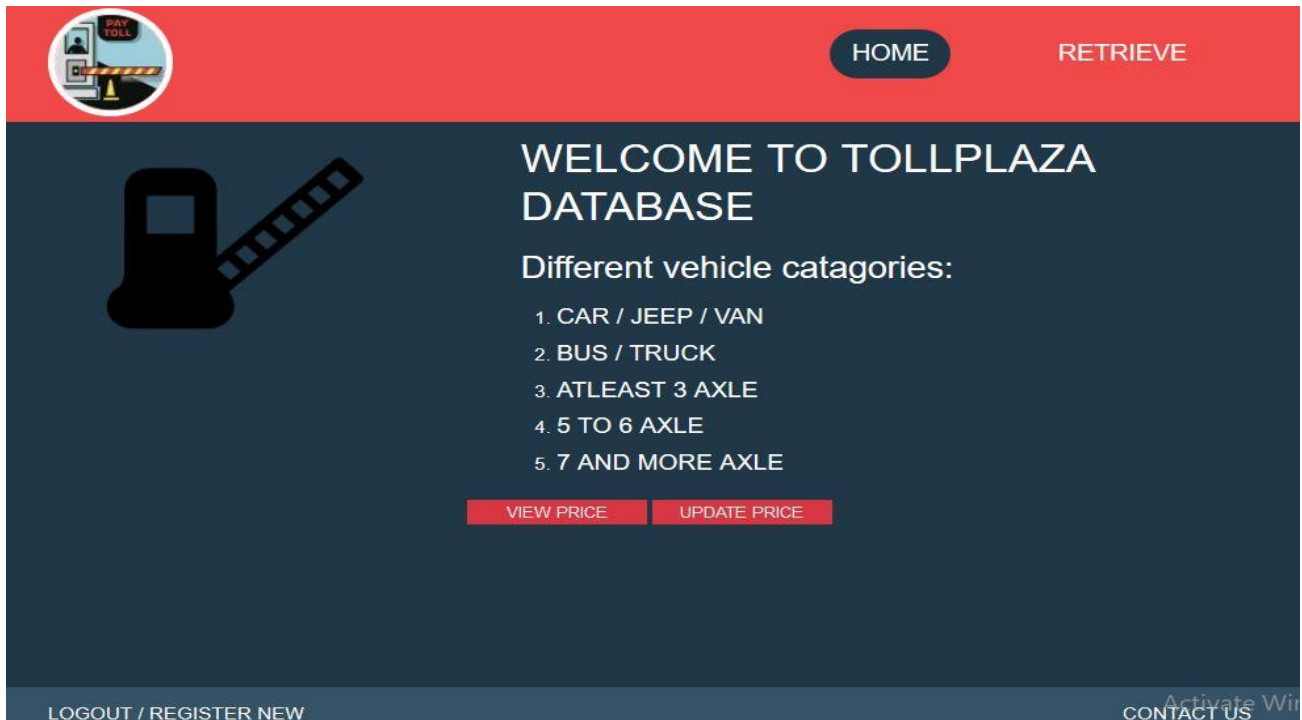
Fig 4.2 Admin home

4.3 Get bill

This page is restricted only for booth. Different informations about the vehicle is entered into the
database in this page.

Fig 4.3 Get bill

4.4 Retrieve data

Retrieval of the data from the database is possible for both admin and booth. Options to search by vehno and licno are provided in booth account.

Fig 4.4 Retrieve data

4.5 Retrieve and modify

This page is accessible only for the admin. In this page the admin can view the details of all the information about the vehicles that are entered into the database by the booths. Admin can also delete the booths and other user information from the database.

Fig 4.5 Modify

CHAPTER 5

CONCLUSION

By doing semi-automation of toll plaza we can have the best solution over money loss at toll plaza by reducing the man power and also can reduce the traffic indirectly resulting in reduction of time at toll plaza. In our project we have implemented the computarized way of storing the information regarding thevehicles that passes through the toll gate. By implementing this project the benefits include better audit control by centralizing accounts, expand capacity without building more infrastructures, easy way to retrieve vehicle data.

REFERENCES

1. MySQL Documentation and Reference Manual:http://dev.mysql.com/docs/en
2. Php Documentation:http://php.net/manual/en/book.com.php
3. Vehicle_Registration_Plates_of_India.Available:http://en.wikipedia.org/wiki/Vehicle_registration_plates_of_India.