

Name: K C A A Iroshan

Student Reference Number: 10638366

Module Code: CNET343SL	Module Name: Distributed Systems (19/AY/AU/M)
Coursework Title: Traffic Fine Handling System	
Deadline Date: 21/04/2020	Member of staff responsible for coursework: Mr. Rasika Alahakoon
Programme: BSc (Hons) Software Engineering	

Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook).

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

K. C. A. A. Iroshan 10638366  
A. A. Ayesh Dulanja 10638431  
M. D. S. Tharindu 10638387  
T.J.M Siriwardhana 10638374  
G H P Prabodhani 10638378  
T. Dasanayake 10638504

***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***

Signed on behalf of the group: K C A A Iroshan

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I \*have ~~used~~/not used translation software.

If used, please state name of software.....

Overall mark \_\_\_\_\_ % Assessors Initials \_\_\_\_\_ Date \_\_\_\_\_

## Acknowledgment

We are very grateful because we managed successfully to complete our Distributed Systems (CNET343SL) assignment within the time given by our module leader **Mr. Rasika Alahakoon**, who has through his knowledge has been able to guide us. The assignment cannot be completed without the effort and co-operation of our group members. We thank everybody who has, directly and indirectly, helped us in this assignment to make it successful.

## Table of contents

Acknowledgment.....	1
Table of contents.....	1
Table of figures .....	3
1. Introduction .....	4
1.1 Project Report Introduction.....	4
1.2 Description of the System .....	4
1.3 Scope of The Project.....	5
1.4 Functionalities .....	5
2. Project Background .....	6
2.1 Objectives .....	6
2.1.1 Background/Motivation .....	6
2.1.2 Project Goals .....	7
2.1.3 Project Objectives.....	7
2.2 Introduction to the Distributed System .....	8
2.2.1 What Is The Importance of Developing A Distributed System .....	8

3. Requirement Analysis .....	9
3.1 Functional Requirements .....	9
3.2 Non-Functional Requirements.....	9
4. System Design .....	11
4.1 System Architecture .....	11
4.1.1 Architecture Diagram .....	11
4.1.2 Hardware/Software Requirements.....	12
4.1.3 UML Diagrams.....	13
4.1.4 Database Design .....	15
5. System Development .....	18
9.1 System Workflow in Detail.....	18
9.1.1 Database Implementation.....	18
9.1.2 Database Usage and Load Balancing .....	18
9.1.3 Database Syncing .....	19
9.1.4 Symmetric DS.....	19
9.1.5 Fault Tolerance.....	22
9.2 Web Application Implementation.....	22
9.2..1 Features of Web Application.....	22
9.2..2 Web Application Workflow .....	25
9.3 Mobile Application Implementation.....	26
9.3.1 Features of Mobile Application .....	26
9.3.2 Mobile Application Work Flow.....	26
6. Tools and Technologies .....	30
6.1 Middleware Technology .....	31
6.2.1 API Implementation .....	31
7. Issues and Approach of Resolving .....	32
8. User Guide .....	33

8.1	User Guide for Mobile Application .....	33
8.2	User Guide for Web Application .....	33
9.	Conclusion .....	33
9.1	Summary.....	33
9.2	Future Enhancements.....	33
10.	Team Details .....	34
10.1	Workload Matrix .....	34
11.	Turnitin Report.....	35
12.	One Drive Link of The Project .....	35

## Table of figures

Figure I - Architecture Diagram .....	11
Figure II - Use Case Diagram.....	13
Figure III - Class Diagram .....	14
Figure IV -Extended Entity Relationship Diagram .....	15
Figure V- Relational Mapping.....	16
Figure VI - Normalized tables.....	17
Figure VII - Database Diagram.....	18
Figure XIV - Symmetric DS Control Software.....	20
Figure XV - Symmetric DS Dashboard.....	21
Figure XVI - Symmetric DS Nodes.....	21
Figure VIII- Login Page .....	25
Figure IX -My Account Page .....	25
Figure X - Splash Screen .....	27
Figure XI - Login Screen .....	28
Figure XII - Driver Home Screen .....	29
Figure XIII- Police Officer Home Screen .....	30

# 1. Introduction

## 1.1 Project Report Introduction

This the project report for the assignment of the module **CNET343SL Distributed Systems**. This document provides an overview of the developed system and further discusses the distributed architecture of the system and its component and software development technologies.

## 1.2 Description of the System

Enforcing “Traffic Laws” & preventing the violation of traffic regulations and prosecution of offenders are two of the main functions of Sri Lankan traffic police. In that process, the fining for violations of traffic regulations involves a lot of paperwork. The police officer fills forms and collects the driving license. Then the driver must go to the post office, make the payment and deliver the receipt to the police station to regain the driving license. This is a long, expensive and time-consuming process.

At present, most of the people use smartphones all the time. Most of the drivers are using GPS technology from their smartphones nowadays. Therefore, we can easily address this issue through smartphones. An automated system for this scenario does not exist yet in Sri Lanka. In order to address this issue, the “**Traffic Fine Handling System**” is proposed.

This is a distributed system with three user types,

- Police Officer
- Driver (Offender)
- Admin (Staff at the police station that handles the document work)

The user interaction with the system happens via a mobile app and a web application. The police officer uses the mobile app on his smartphone to generate a fine. The driver or the offender can either use the smartphone app or the web application to view the fines under his driving license and pay the fine online. The application tracks the fining deadline and if the fine is not paid within the 14 days, the staff at the office and the police officer who generated the fine are notified.

### 1.3 Scope of The Project

The system consists of a web application and a mobile app. The system has a web server that hosts the web application. There are two database servers. One server handles the retrievals of data and the other server handles the updating and insertion of data. The mobile application and the web application use these servers to store, update, delete and retrieve data. The servers are synced properly. The database in each server is identical to each other.

### 1.4 Functionalities

Functionalities of the Distributed Traffic Fine Handling System will be as follows.

From the police officer aspect,

- The police officer can log in to the mobile app or web app by using a username and password.
- The police officer enters the details (that would normally be inserted to a form) into the app and files the fine.
- They can view paid fine details as well as not paid fine details.
- Once a made a fine that police officers can see relevant drivers' details.
- They can view the fine details they have already made.
- The police officer can maintain his/her account.

From the driver aspect,

- The driver can log in with the mobile app or web app by using username and password
- The driver can see the relevant fine details that he/she should responsible for.
- Already paid fine details can be seen.
- Card settings can be done (payments) through the app.
- The driver can maintain his/her own account.

From the admin aspect,

- Admin can add police officers to the system or remove police officers from the system.
- Responsible for server side handling.
- Only the admin has the authority to access the database.

## 2. Project Background

### 2.1 Objectives

#### 2.1.1 Background/Motivation

The main purpose of this project is to develop skills regarding distributed system development and to give a solution to the problem defined. This project does not have a client to fund it. But, we believe there are potential clients for this solution when this project becomes successfully completed. In the existing process, the Sri Lankan Police and Sri Lankan Postal Department works jointly. The collection of the fines happen through the postal department. Under this process, the driver who's paying the fine undergoes a lot of difficulties. Other than the amount paid

as fines, they have to spend more time and money throughout the process. There are instances where the post office is closed and the police officers have to do all the tasks manually which involves a lot of paperwork. Our project aims to make this process easier and convenient through the application of modern technology.

### 2.1.2 Project Goals

The main goals of the project are,

- 1) Making it easy for drivers to pay the fines without having to undergo unnecessary difficulties.
- 2) Removal of the manually done paperwork of the existing system.
- 3) Reduce the time that it takes to process the fines (receiving the fine payments).
- 4) Reduce the overall cost involved with the process in the existing system, enabling the departments to save money.

Apart from the above project goals, to learn about distributed systems, related technologies and improve our technical skills and knowledge was our second goal as a team.

### 2.1.3 Project Objectives

- 1) Study and analyze existing processes and procedures in the world.
- 2) Requirements gathering.
- 3) Provide interoperability between the new application and the existing process.
- 4) Analyze relevant development technologies and deployment solutions.
- 5) Implement of the distributed system.
- 6) Introduce the application to the relevant structures. [If needed])



## 2.2 Introduction to the Distributed System

Networks of computers are all over the world today. The Internet is one of those, just like the other networks which are composed and available all over the world. Mobile phone networks, Personal area networks, Local area networks, corporate networks, virtual private networks, factory networks, storage processes on a network. Hence, all components together can perform single or set of corresponding area networks, campus networks, enterprise private networks, home networks, in-car networks and all of these, contains and is responsible for very similar characteristics which makes them a discussed topic under distributed systems. A distributed system can be explained as an application which executes a collection of protocols to synchronize the actions of various tasks. Computers that are connected by a network might be physically separated by any distance. This can vary from a couple of meters to thousands of kilometers. They can also be in separate countries or even in different continents far away from each other, or even in the same country, same building and next to each other in the same room.

### 2.2.1 What Is The Importance of Developing A Distributed System

This topic can be discussed very widely. There are many more advantages of developing a distributed system. Scalability and resource sharing are the two main features of why distributed systems are highly important. Furthermore, distributed systems can be changes to accommodate more users and resources whenever needed. In that case, a distributed system can be very large in size and very powerful when the overall capabilities of all the devices connected to the network are concerned.

## 3. Requirement Analysis

### 3.1 Functional Requirements

Functional requirements are the requirements that specify how the system should behave under different conditions.

- Services can be distributed within several API components.
- HTTP Request and HTTP Response method need to implement using NodeJS.
- The RESTful architectural pattern needs to use to make this distributed application.
- Backup of all services and API components need to implement with a backup server.
- Both web and mobile applications support JSON data types.
- Both web application and mobile application uses the MySql database.
- Supports continuous integration.

### 3.2 Non-Functional Requirements

Non-functional requirements are the requirements that specify the qualitative attributes of the system.

- Scalability

The capability of a system to manage the growing amount of workload is known as scalability. Traffic Fine Handling system must be a scalable system because it serves thousands of users spread out all over the country. The system must be scalable to enhance the performance of the system and to handle a large amount of users.

- Reliability

Traffic Fine Handling system must support 24\*7 for its customers. Therefore it must be a reliable system. The system is carefully designed to avoid or cover failures that can happen while running the system.

- Performance

The system must perform with short response time or low delayed time, high throughput, and low utilization.

- Availability

The system must be available when it capable of providing intended services. If the system down for a second it can be affected by millions of users. It is necessary to have efficient recovery or overprovisioning the resources and making the resources redundant. An additional web server can be deployed to host the service.

- Heterogeneity

The system must be able to interact with more than two devices that have different technologies, operating systems, programming languages, hardware platforms, and network protocols. The developed system interacts with the mobile device, Laptop, and PC that can communicate with each other in order to request a service and provide a service as a response.

- Security

Traffic Fine Handling system must be secure because it deals with police and driver details over the internet. Therefore we validate each and every form that related to the security of the whole system. We validate usernames over web service.

## 4. System Design

### 4.1 System Architecture

#### 4.1.1 Architecture Diagram

Architectural Diagram

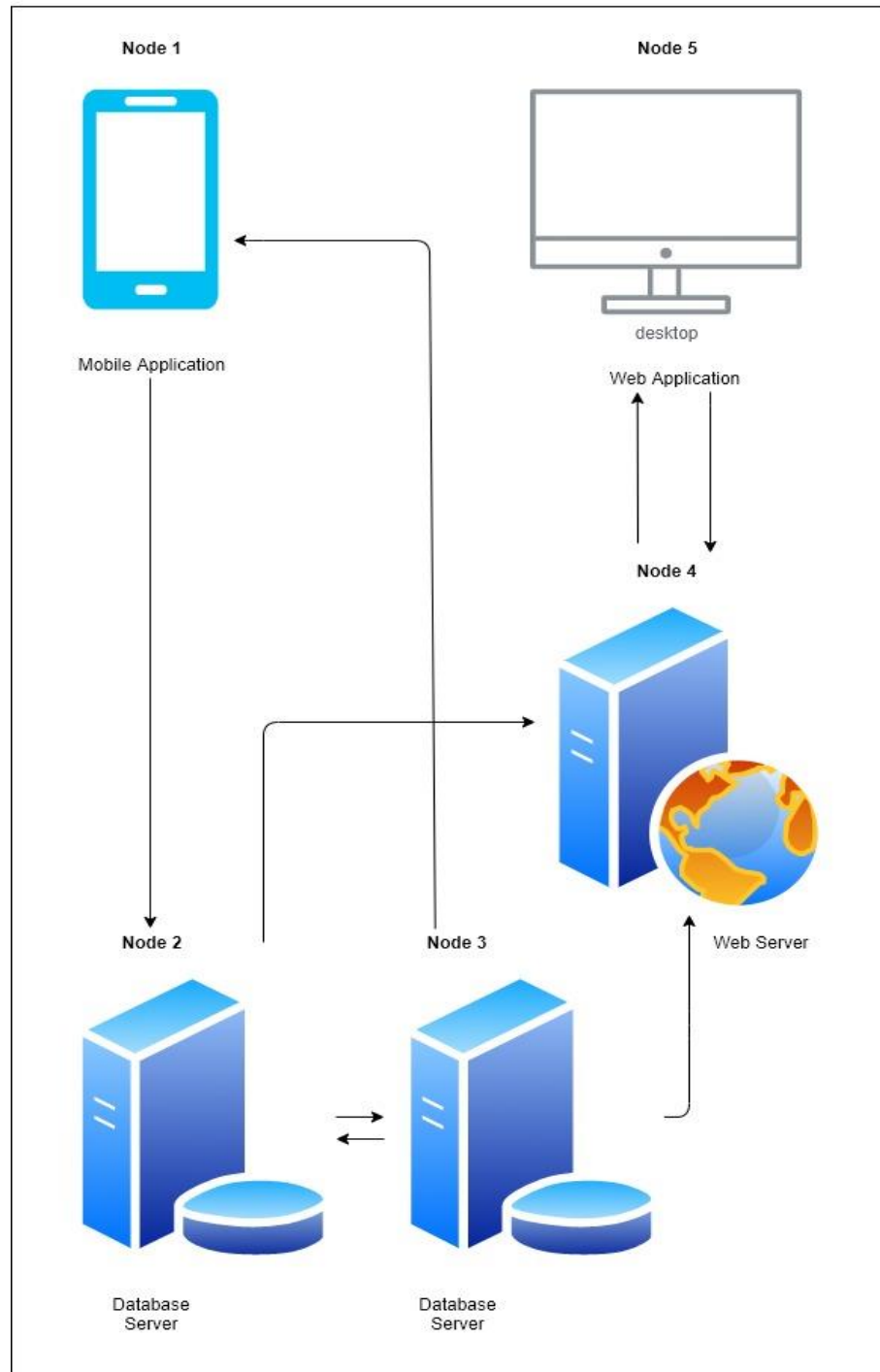


Figure 1 - Architecture Diagram

#### 4.1.2 Hardware/Software Requirements

these are the minimum requirements required for the Traffic Fine Handling system

##### Hardware Requirements

- **Web service**
  - Intel Core i3 2.5GHz or higher processor
  - Intel Xeon E52418L processor or higher (if using dedicated webserver)
  - 2GB Ram
  - 250MB hard disk space
- **Mobile application**
  - Android-based smartphone
  - Snapdragon 652 or higher processor
  - 2GB system ram

##### Software Requirements

- **Web service**
  - NetBeans IDE 8.1 or 6
  - MySQL
  - Glassfish server
- **Mobile application**
  - Android Marshmallow or higher OS
- **Website**
  - Local web hosting platform XAMP, WAMP

### 4.1.3 UML Diagrams

#### Use Case Diagram

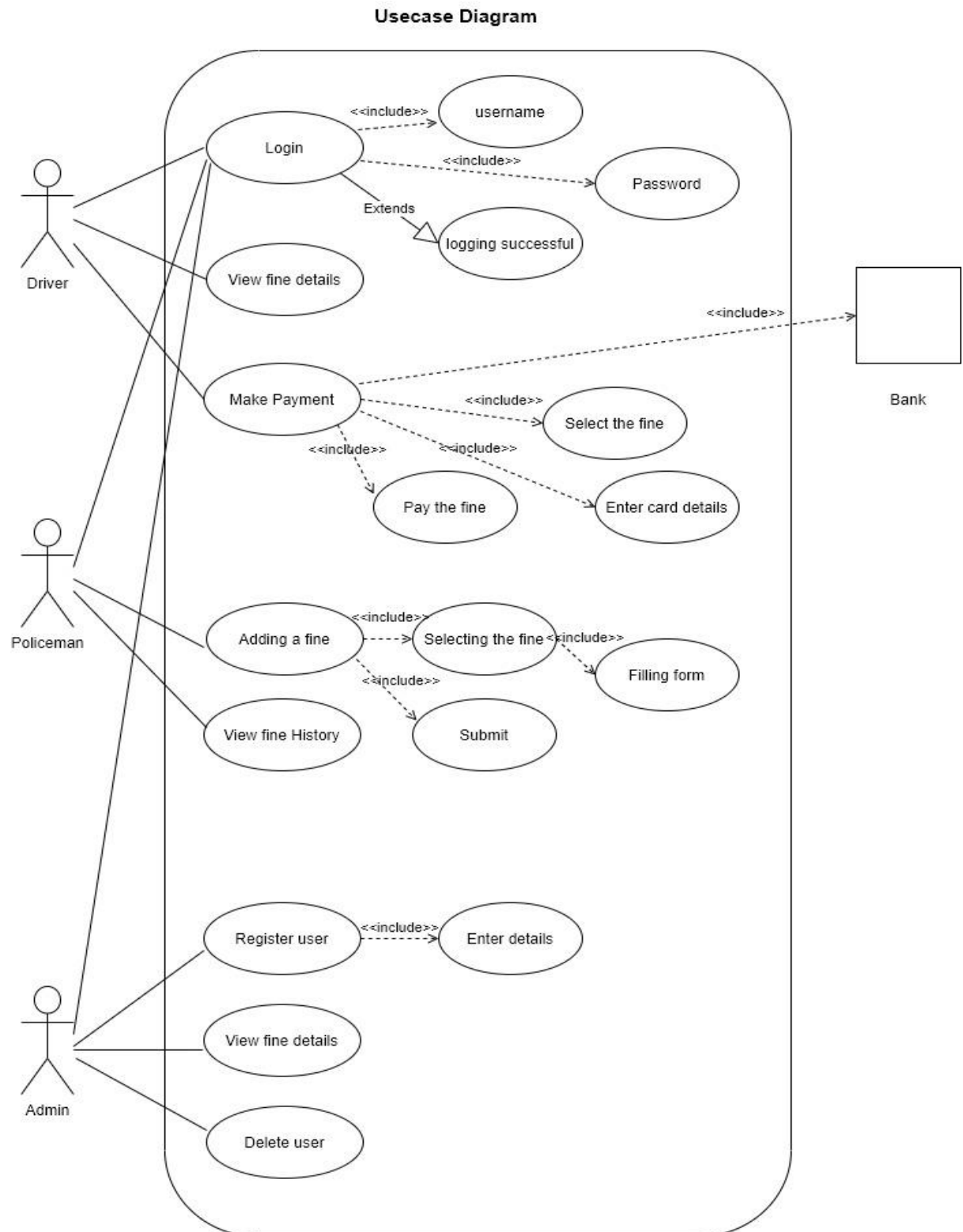


Figure II - Use Case Diagram

## Class Diagram

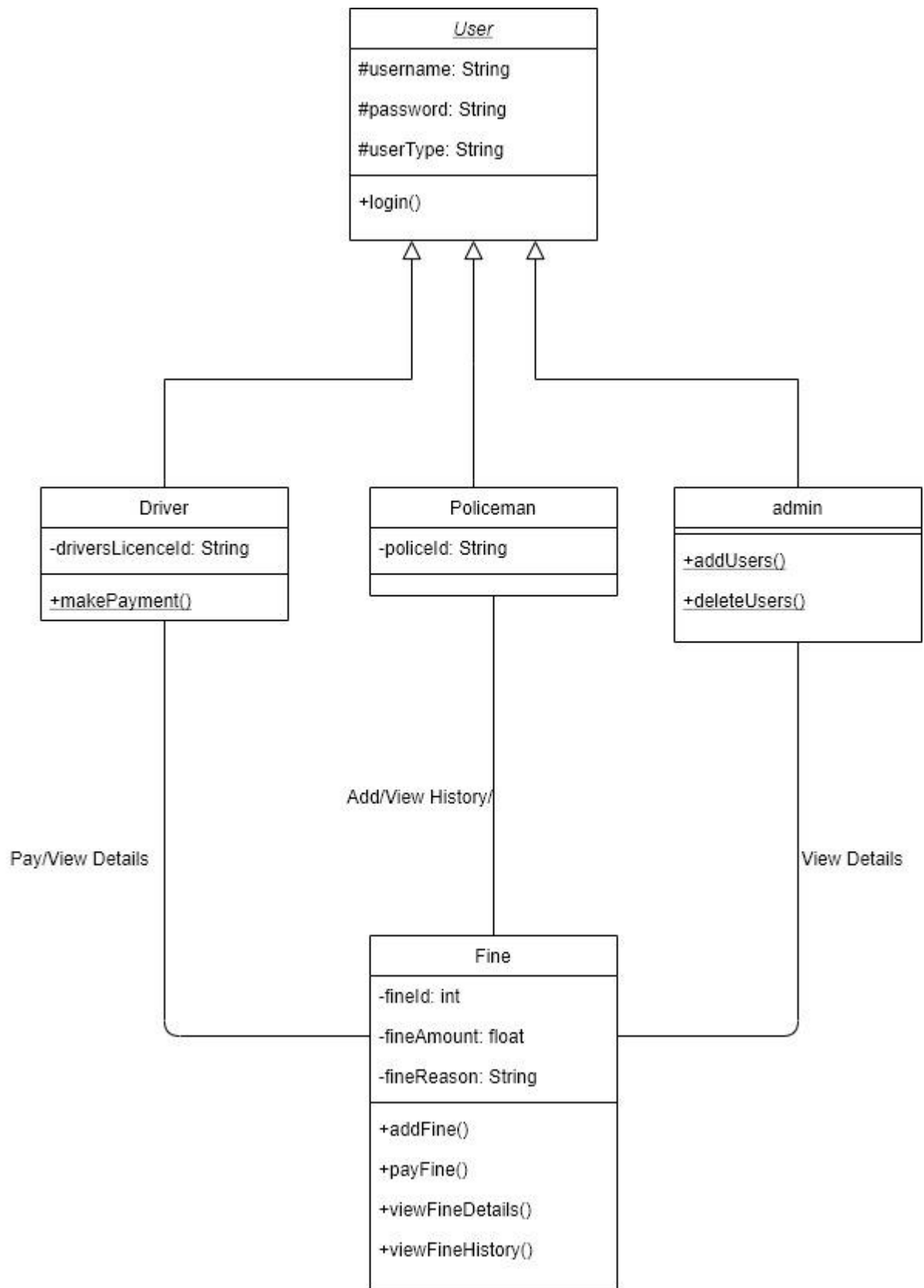


Figure III - Class Diagram

#### 4.1.4 Database Design

### Extended Entity Relationship Diagram

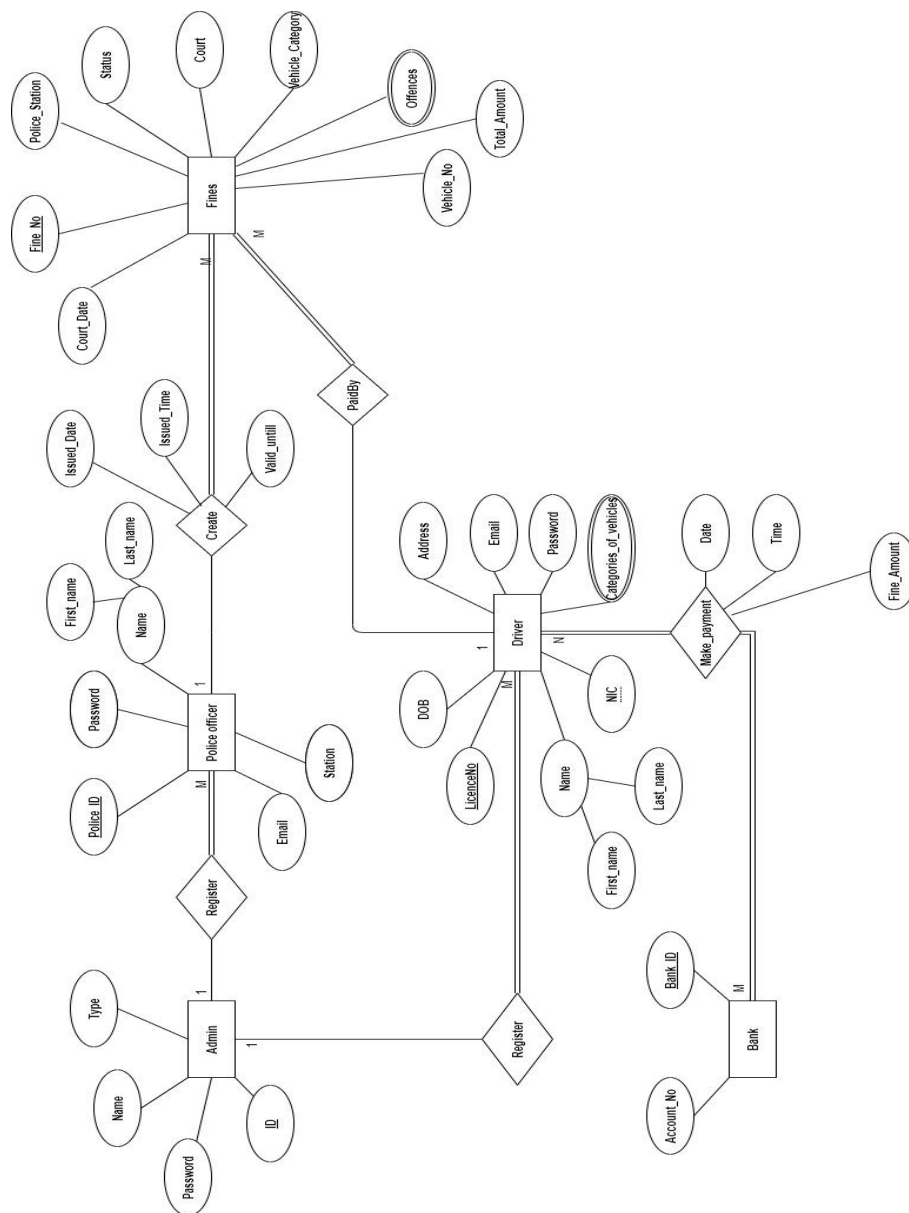


Figure IV -Extended Entity Relationship Diagram



## Relational Mapping

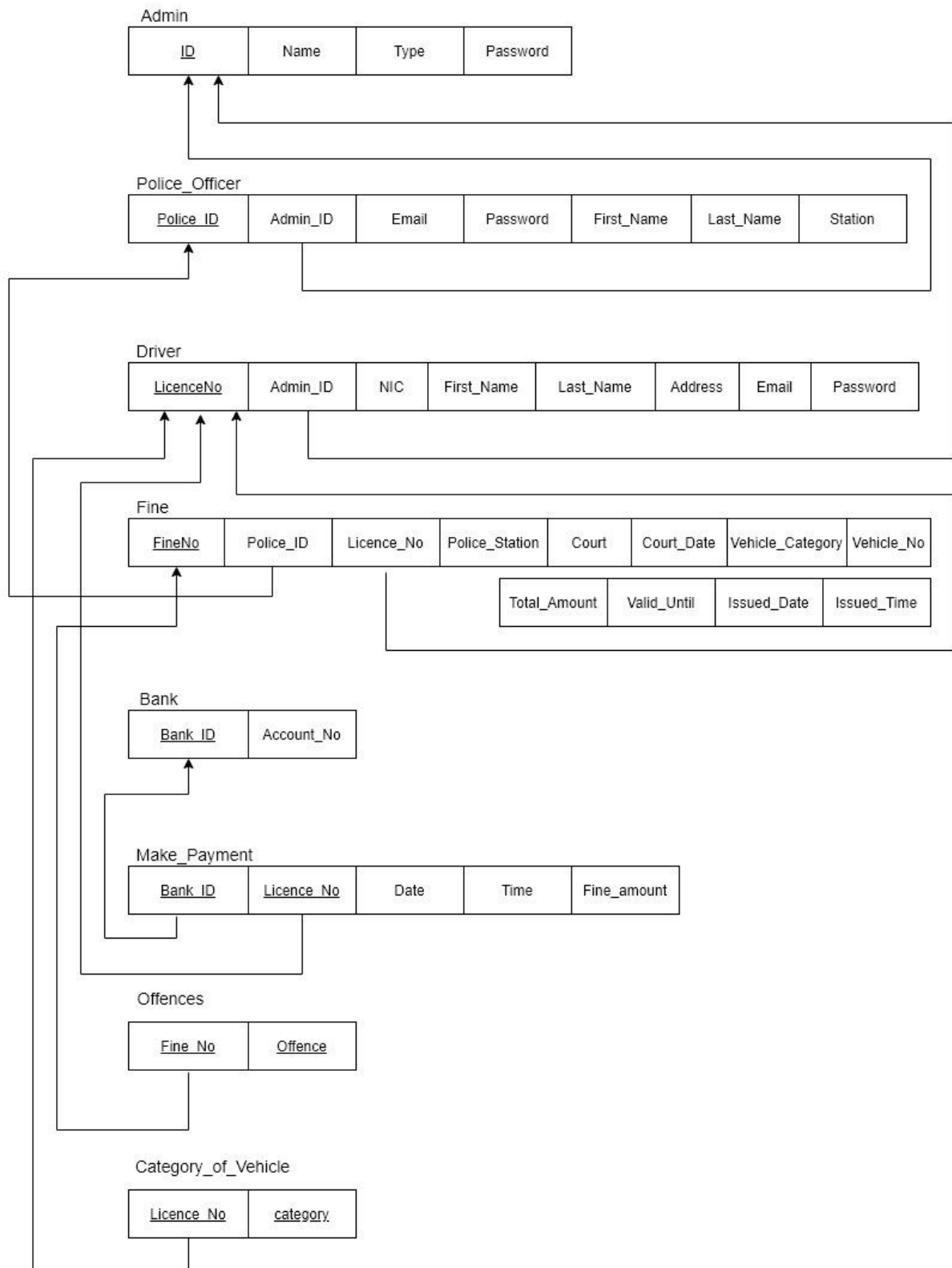


Figure V- Relational Mapping

## Normalized Table

Admin

<u>ID</u>	Name	Type	Password
-----------	------	------	----------

Police\_Officer

<u>Police_ID</u>	Admin_ID	Email	Password	First_Name	Last_Name	Station
------------------	----------	-------	----------	------------	-----------	---------

Driver

<u>LicenceNo</u>	Admin_ID	NIC	First_Name	Last_Name	Address	Email	Password
------------------	----------	-----	------------	-----------	---------	-------	----------

Fine

<u>FineNo</u>	Police_ID	Licence_No	Police_Station	Court	Court_Date	Vehicle_Category	Vehicle_No
				Total_Amount	Valid_Until	Issued_Date	Issued_Time

Bank

<u>Bank_ID</u>	Account_No
----------------	------------

Make\_Payment

<u>Bank_ID</u>	<u>Licence_No</u>	Date	Time	Fine_amount
----------------	-------------------	------	------	-------------

Offences

<u>Fine_No</u>	<u>Offence</u>
----------------	----------------

Category\_of\_Vehicle

<u>Licence_No</u>	<u>category</u>
-------------------	-----------------

Figure VI - Normalized tables

## Database Diagram

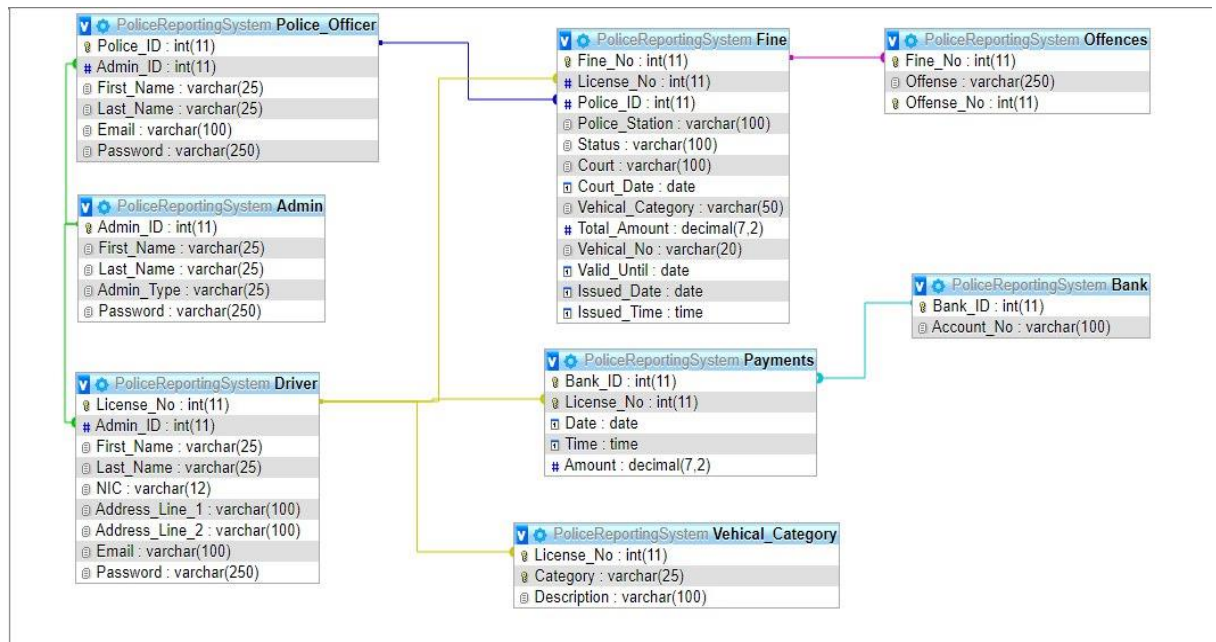


Figure VII - Database Diagram

## 5. System Development

### 9.1 System Workflow in Detail

#### 9.1.1 Database Implementation

The system is consisted of two MySQL databases. The two databases are exact copies of each other. The two databases are stored on two computers/servers. CRUD operations of the two databases are handled by RESTFUL CRUD APIs developed by Node JS. Each database run on express servers.

#### 9.1.2 Database Usage and Load Balancing

If we refer the two databases as Database\_1 and Database\_2, Database\_1 is used for CREATE, UPDATE and DELETE operations. Database\_2 is used for RETRIEVE operations. In other words, all data

retrievals are done on Database\_2 and data insertion, updating and removal is done on Database\_1. The purpose for this approach is load balancing between the two databases. Load here is the traffic for each database connection.

### 9.1.3 Database Syncing

The two databases are synchronized with the use of a replication tool called Symmetric DS. The two databases have two roles, which are master database and the client/slave db. The two databases are connected through the Symmetric DS server. The replication process is done with a trigger based capture method. whenever the source database (Master db. in this case) executes a CRUD operation, a function created by the symmetric ds is triggered and it collects the operation details (operation type, table etc.) and relevant data of the operation. Then the details and data is sent to the Symmetric DS server where it is queued and routed to the target database(slave db in this case). after routing is done, the same operation is executed with the same data at the target database replicating the data from the master database

### 9.1.4 Symmetric DS

Symmetric DS is a standalone tool server with a web dashboard for data replication between databases. For this system, we have used a Master-Slave node setup for data replication and synchronization between databases. The replication process is a trigger-based capture process; when the master node database executes an insert/update/delete query, that triggers a procedure automatically made into the database by the symmetric DS server which triggers the replication process. Replication is done in three phases: extraction, routing, and loading. First, the server extracts the relevant data from the source/master database, then it routes the data along with the query to the target/slave database via the

symmetric DS server and finally, the data is loaded to the target database.

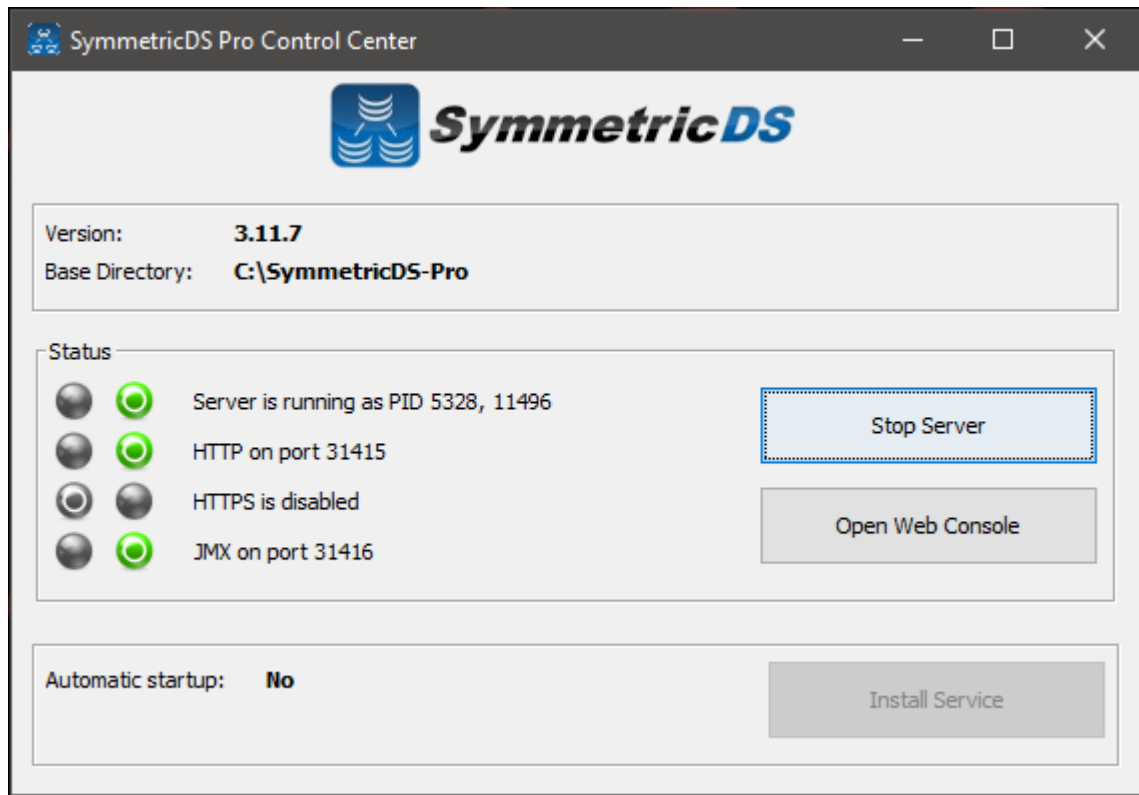


Figure VIII - Symmetric DS Control Software

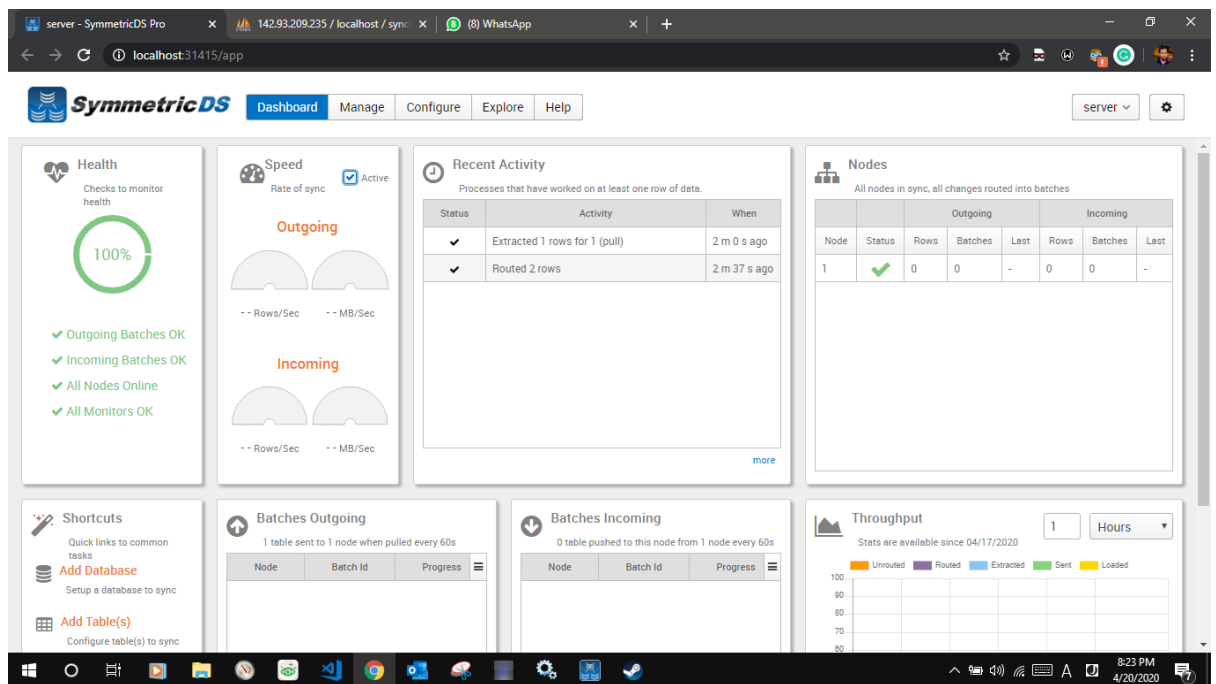


Figure IX - Symmetric DS Dashboard

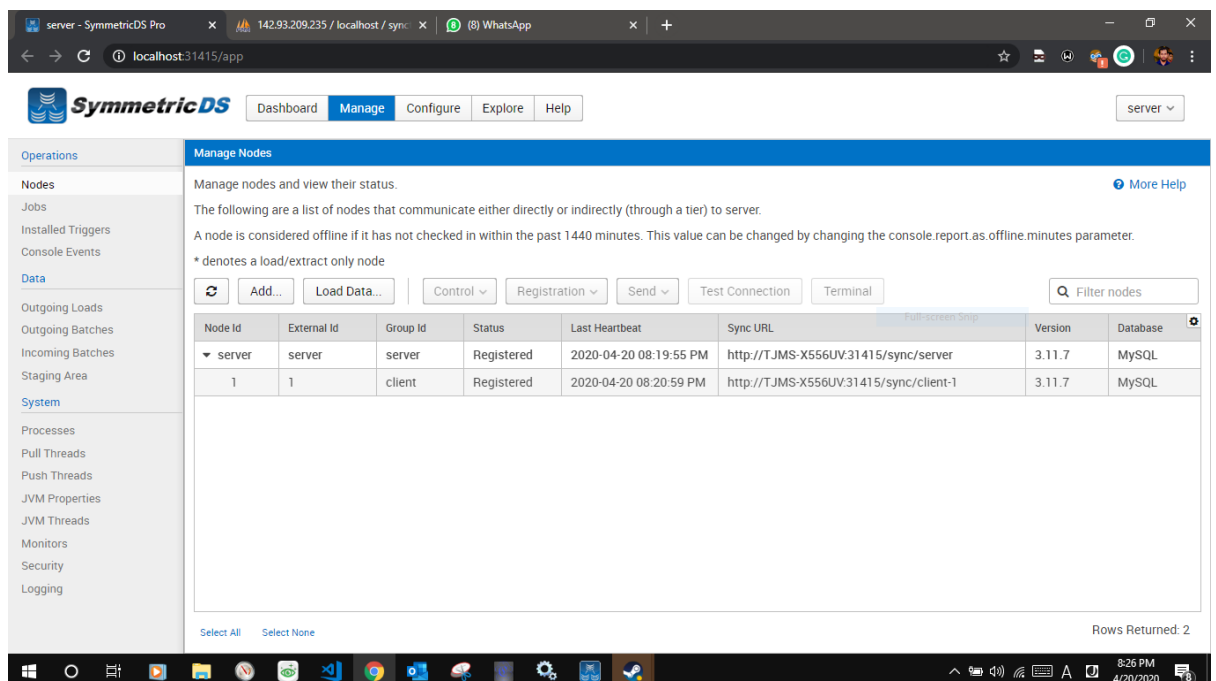


Figure X - Symmetric DS Nodes

## Instruction for setup Symmetric DS

- Download the executable jar file from the provided link and install it(Required JDK).
- <https://www.jumpmind.com/products/symmetricts/download>

- Then start the server using Symmetric DS control panel from the start menu shortcut and open the web dashboard.
- In the web dashboard enter the trial key provided to continue.
- <https://github.com/IRoDevO/PoliceReportingSystemDistributed/tree/master/SymmetricDS>
- Create a server on from the dashboard and add the Master DB by providing database URL and credentials
- Then create a Slave DB on the local node by providing the db URL and credentials
- After that the dashboard will prompt to select tables to replicate, select the tables and finish the setup.
- If everything was done, there should be no errors and the databases should be in sync.

#### 9.1.5 Fault Tolerance

In case where Database\_1 fails, Database\_2 would be utilized for all CRUD operations. If Database\_2 fails, Database\_1 goes for full usage. The redundant aspects of having two databases is utilized here. When this happens, when a single database fails, the system would sacrifice its redundancy and load balancing for the continuous operation, for maintaining Availability.

## 9.2 Web Application Implementation

### 9.2..1 Features of Web Application

- Index Page

When the user first accesses the web site, it will navigate to the Index Page. It will contain details about the site.

- Login Page

The user has to select the user category while logging to the website. Users are categorized as Admin, Driver and Police officer. Registered users(driver/police officer) can log in to the system using a user name and password.

- **Register Page**

If someone is not a registered user they can sign up for the system by given required details.

- **My Account**

After the successful login process of any user, the application will navigate to My Account page. According to the category of user, there are three types of My Account pages.

- Admin My Account
- Driver My Account
- Police Officer My Account

### **1) Admin My Account**

After a successful login process of the Admin, the application will navigate the Admin My account which is the main screen design for admin of the system. The following are the features available on the admin my account.

- **Add User**

Admin can add users to the system by giving the required details.

- **Edit Users**

Admin can view registered users' details using this feature and also he has the authority of removing users from the system.

### **2) Driver My Account**



After a successful login process of the driver, the application will navigate the Driver My Account which is the main screen design for drivers. The following are the features available on the page.

- View fines

By using this feature the driver can view all the fine details that he/she is responsible for.

- History

Using this feature driver can see the previous fine details that he/she already paid.

- Pay fine

The driver can pay his/her fines through the website.

### **1) Police Officer My Account**

After a successful login process of the police officer, the application will navigate the Police Officer My Account Page which is the main screen design for the police officer. The following are the features available on the page.

- History

Using this feature police officers can see all the fine details that he had made.

## 9.2..2 Web Application Workflow

**POLICE REPORTING SYSTEM**  
Alert today - Alive tomorrow

HOMELOG IN

### LOG IN

Login Type \*  
Police Officer

User Name \*  
Enter Your Username

Password \*  
Enter Your Password

Login

Search...

Search

#### Recent Posts

Welcome to Police Reporting System

#### Archives

April 2020

#### Meta


Site Admin  
Log out  
Entries feed  
Comments feed  
WordPress.org

Figure XI- Login Page

**POLICE REPORTING SYSTEM**  
Alert today - Alive tomorrow

HOMELOG INMY ACCOUNT

### My Account



Type	Driver
First Name	Asini
Last Name	Appuhami
License Number	7896541
NIC	19961120015
Address	a23 Supermarket Street, Veyangoda 11100
Email	appuhami@gmail.com
Contact No.	0768386669

Edit My Details

Search...

Search

#### Recent Posts

Welcome to Police Reporting System

#### Archives

April 2020

#### Meta

Site Admin  
Log out  
Entries feed  
Comments feed  
WordPress.org

Figure XII -My Account Page

## 9.3 Mobile Application Implementation

### 9.3.1 Features of Mobile Application

- Splash screen

When the user first launch the EasyFine app, it will navigate to login screen through splash screen. Splash will contain the application name animation.

- Common login

After the walkthrough of splash screen, it will navigate to the login screen which includes the username and password fields for authentication process of the user.

- Driver's home page

Driver can see his/her fines through this page and he/she can make payment using "pay" button on the relevent fine. Also, he/she can see the past fines, which are paid before.

- Officer's home page

Officer can see all the fines that he/she created before for the drivers using "History" tab. If there are court cases for the drivers who didn't pay the fine within 14 days,

it'll desplayed under the "court cases" tab.

Also, officer can creates new fines using the "ADD" button.

### 9.3.2 Mobile Application Work Flow



Figure XIII - Splash Screen

← Sign In To Easy ... Sign Up

Email

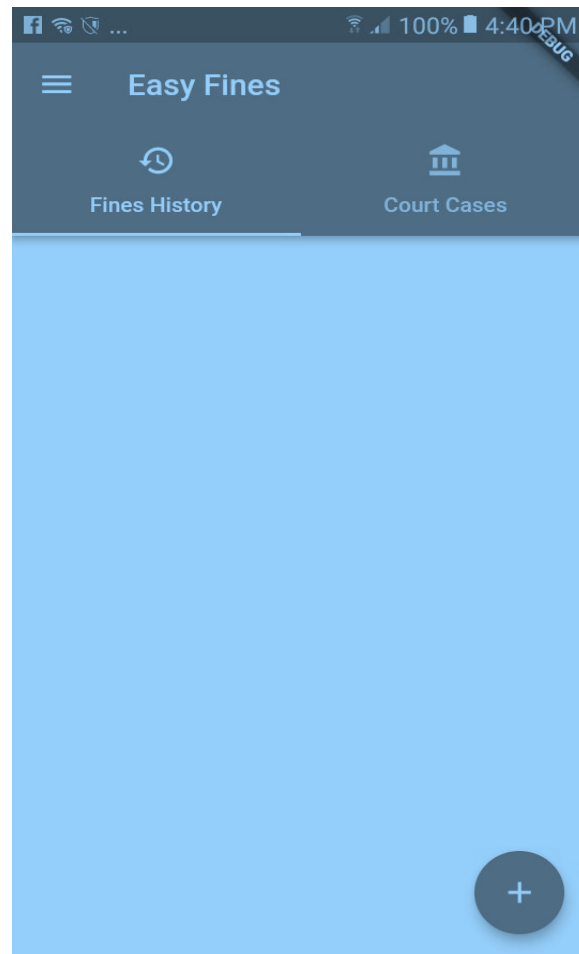
Password

Sign In

Figure XIV - Login Screen



Figure XV - Driver Home Screen



*Figure XVI- Police Officer Home Screen*

## 6. Tools and Technologies

For Web Application :

- PHP
- CSS
- HTML
- nodeJS
- MySQL
- JavaScript

For Mobile Application :

- Flutter
- Dart
- Node JS

- MySQL

Tools :

- WordPress
- Sublime Text
- VS Code
- Postman

## 6.1 Middleware Technology

A middleware is a piece of software that offers a set of services and capabilities to applications outside of what's offered by the operating system, it can interconnect applications, data, and users. In this system, the restful API acts as the middleware by exposing services, resources, and functionalities to the applications using endpoints. It allows the applications to send and retrieve data from the databases by consuming the services provided by the API.

### 6.2.1 API Implementation

#### **API Development Technologies**

Mobile Application and Web Application use NodeJS REST API for controlling the database and distributed application. Rest API (Representational State Transfer) APIs are web standards-based architecture and use the HTTP Protocol for exchanging data between applications or systems. In RESTFUL web service HTTP methods like GET, POST, PUT and DELETE can be used to perform CRUD operations. REST is very simple compare to other methods like SOAP, CORBA, and WSDL, etc.

#### **API Development Server Details**

Mainly here we have created two databases that are run in the localhost of each server along with their APIs. One is used to retrieve data and another database is used to insert, update, and delete data. Here we



used the same database and the same API and two databases are synchronized in the background. We are using the Localhost as our server.

Host : Localhost

USER : "root",

PASSWORD : "iroshan",

DATABASE : "Police\_Reporting\_System"

HOST, USER is custom names given no variables.

### **API Documentation**

<https://documenter.getpostman.com/view/8511782/Szf81SyM>

## **7. Issues and Approach of Resolving**

Some issues occurred while implementing APIs. We had problems with technologies that we used to implement the APIs. NodeJS used as implementing technology. We also did our best in trying to implement Remote Method Invocation in the NodeJS API. but due to limitations of javascript we were unable to successfully implement it. Some issues were raised when synchronizing the databases and we used SymmetricDS to solve it.

Connecting android application to the service was another issue that we faced while implementing. Some tutorials were followed to get some understanding of how to handle android technologies.

## 8. User Guide

### 8.1 User Guide for Mobile Application

Requirements :      Android-based smartphone

### 8.2 User Guide for Web Application

Requirements :      web browser

The web app will open in your default browser.

## 9. Conclusion

### 9.1 Summary

At the end of the project, the outcome of the project development was as expected. All most all the functionalities have the capability of function according to the functional specification. Overall the objective of this project was achieved which emphasizes that the proposed concept can be developed in order to distribute the functionalities of the Traffic Fine Handling system with enhancements.

### 9.2 Future Enhancements

Android application and web application can be further improved to increase system usability.

# 10. Team Details

## 10.1 Workload Matrix

Workload	10638366	10638374	10638431	10638378	10638387	10638504
API Development	✓					
Web Application Development			✓			
Database design and development				✓	✓	
Mobile Application Development						✓
Database server, distribution and replication		✓				

## 11. Turnitin Report

CNET343SL Team 64.docx			
ORIGINALITY REPORT			
9%	2%	1%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	Submitted to National Institute of Business Management Sri Lanka Student Paper	2%	
2	Submitted to Liverpool John Moores University Student Paper	1%	
3	www.clickessays.com	1 or	

## 12. One Drive Link of The Project

<https://drive.google.com/open?id=1xBEoEV2QZFLbKdvRX5GZYbv7a9YgPW47>

