

**TOLLGATE MANAGEMENT SYSTEM**

**A MINI-PROJECT REPORT**

*Submitted by*

SAKETHRAM BS                    241901094

SUSHANTH REDDY PV              241901115

*in partial fulfillment of the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**(CYBER SECURITY)**



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND CYBER SECURITY

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

November 2025

## **BONAFIDE CERTIFICATE**

Certified that this project “**TOLLGATE MANAGEMENT SYSTEM**” is the bonafide work of “SAKETHRAM BS (241901094), SUSHANTH REDDY PV(241901115),” who carried out the project work under my supervision.

### **SIGNATURE**

**Ms.RUPMALA**

**ASSISTANT PROFESSOR**

Dept. of Computer Science and Engineering  
and Cyber Security  
Rajalakshmi Engineering College  
Chennai

This mini project report is submitted for the viva voce examination to be held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **ABSTRACT**

The Tollgate Management System is a database-driven application designed to automate and streamline toll collection processes at highway toll plazas. It aims to reduce manual intervention, improve operational efficiency, and ensure accurate financial tracking. The system records vehicle entries and exits, calculates toll charges based on vehicle type and route, and manages payment transactions securely.

Key modules include vehicle registration, owner details, toll rate configuration, transaction logging, and staff management. The solution improves transparency, lowers the amount of labor that needs to be done by hand, and helps data-driven decision-making by combining dynamic scheduling with live attendance monitoring. In the future, this approach could get even better by adding RFID technology to ID cards or mobile devices that can use NFC. The backend database ensures data integrity through normalization and relational design, supporting real-time queries and report generation. The system also integrates modern payment methods such as RFID and FASTag, promoting digital tolling and reducing congestion

Keywords: Java, DBMS, Automated attendance tracking, Data-driven decision-making, Educational technology

## **ACKNOWLEDGEMENT**

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal**Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Mr. BENEDICT** for being ever supporting force during our project work.

We also extend our sincere and hearty thanks to our internal guide **Ms.RUPMALA** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

Rajalakshmi Engineering College,

Chennai November 2025

**SAKETHRAM B S**

**SUSHANTH REDDY P V**

## TABLE OF CONTENTS

| <b>CHAPTER NO</b> | <b>TOPIC</b>                      | <b>PAGE NO</b> |
|-------------------|-----------------------------------|----------------|
| 1.1               | INTRODUCTION                      | 1              |
| 1.2               | SCOPE OF WORK                     | 1              |
| 1.3               | PROBLEM STATEMENT                 | 1              |
| 1.4               | AIM AND OBJECTIVES                | 2              |
| 2.1               | HARDWARE SPECIFICATIONS           | 3              |
| 2.2               | SOFTWARE SPECIFICATIONS           | 3              |
| 3                 | MODULE DESCRIPTION                | 4              |
| 4                 | SAMPLE CODING                     | 5              |
| 5                 | SCREENSHOTS                       | 12             |
| 6                 | CONCLUSION AND FUTURE ENHANCEMENT | 15             |
| 7                 | REFERENCES                        | 16             |

## LIST OF FIGURES

| <b>FIGURE NO.</b> | <b>TITLE</b>              | <b>PAGE NO.</b> |
|-------------------|---------------------------|-----------------|
| <b>5.1</b>        | INTRODUCTION PAGE         | <b>12</b>       |
| <b>5.2</b>        | LOGIN PAGE                | <b>12</b>       |
| <b>5.3</b>        | VEHICLE ENTRY             | <b>13</b>       |
| <b>5.4</b>        | PAYMENT EDITOR            | <b>13</b>       |
| <b>5.5</b>        | RECEIPT PRINTER           | <b>14</b>       |
| <b>5.6</b>        | VEHICLE ENTRY DATE FILTER | <b>14</b>       |

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

The Tollgate Management System is designed to automate toll operations. It records vehicle entry/exit, calculates toll charges based on vehicle type, and stores payment details. It reduces manual work and speeds up toll processing through digital methods like RFID / FASTag.

### **1.2 SCOPE OF THE WORK**

This system helps to manage toll collection in busy highways. It provides quick access to vehicle data, accurate charge calculation, and secure transaction handling. It supports better traffic flow and modernizes toll management.

### **1.3 PROBLEM STATEMENT**

Many toll plazas still use manual methods which cause delay, errors in charge calculation, long queues, and improper record handling. Due to increasing vehicles, manual systems are not efficient. Hence, a computerized toll management system is needed.

## **1.4 AIM AND OBJECTIVES OF THE PROJECT**

The aim of this project is to automate toll billing and maintain proper records of every vehicle and payment. The system ensures fast toll processing, accurate data storage, digital payment support, and reduces human error, making toll operations more efficient.

## **CHAPTER 2**

### **SYSTEM SPECIFICATIONS**

#### **2.1 HARDWARE SPECIFICATIONS**

|             |   |                 |
|-------------|---|-----------------|
| Processor   | : | Intel i5        |
| Memory Size | : | 8GB (Minimum)   |
| HDD         | : | 256 GB(Minimum) |

#### **2.2 SOFTWARE SPECIFICATIONS**

|                  |   |               |
|------------------|---|---------------|
| Operating System | : | WINDOWS<br>10 |
| Front – End      | : | JAVA          |
| Back - End       | : | MySql         |
| Language         | : | JAVA,SQL      |

# **CHAPTER 3**

## **MODULE DESCRIPTION**

This application consists of two modules. When the program runs, it will ask for a confirmation to the login window. The person who interacts can login either as an Administrator or as a Staff/User. The description of the modules are as follows:

### **1. Admin Login**

When the person who interacts tries to login as an Admin, then he needs to enter his username and password. The administrator has the full access to modify, update, configure toll rates, manage staff details, view complete transaction logs, manage vehicle registration details, and manipulate all data in the database.

### **2. User / Staff Login**

When the person tries to login as a User/Staff, then he/she will be prompted to enter vehicle details such as vehicle number, vehicle type, entry point, etc. Based on the selected route, the final toll amount will be generated and printed. The staff can only perform toll processing operations and view related transaction details but cannot alter backend configuration or database values.

## **CHAPTER 4**

### **SAMPLE CODING**

#### **LOGINFORM.JAVA CODE**

```
package com.tollgate;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class LoginForm extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;

    public LoginForm() {
        setTitle("Login");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(3, 2));

        add(new JLabel("Username:"));
        usernameField = new JTextField();
        add(usernameField);

        add(new JLabel("Password:"));
        passwordField = new JPasswordField();
        add(passwordField);
    }
}
```

```

loginButton = new JButton("Login");
add(new JLabel()); // empty cell
add(loginButton);

loginButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        authenticateUser();
    }
});

setLocationRelativeTo(null); // center window
setVisible(true);
}

private void authenticateUser() {
    String username = usernameField.getText();
    String password = new String(passwordField.getPassword());

    try (Connection conn = DatabaseConnector.connect()) {
        String sql = "SELECT * FROM users WHERE username = ? AND password = ?";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setString(1, username);
        stmt.setString(2, password);
        ResultSet rs = stmt.executeQuery();

        if (rs.next()) {
            JOptionPane.showMessageDialog(this, "Login successful!");
            dispose(); // close login window
            new VehicleEntryform(); // launch main GUI
        } else {
            JOptionPane.showMessageDialog(this, "+ Invalid credentials.");
        }
    }
}

```

```

    }
} catch (Exception ex) {
    ex.printStackTrace();
    JOptionPane.showMessageDialog(this, "+ Error connecting to database.");
}
}
}
}
}
```

## **VEHICLEENTRYFORM.JAVA CODE**

```

package com.tollgate;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class VehicleEntryform extends JFrame {
    private JTextField numberField;
    private JComboBox<String> typeBox;
    private JButton submitButton;
    private JTable historyTable;
    private DefaultTableModel tableModel;

    // ◆ Filter Components
    private JComboBox<String> filterTypeBox;
    private JTextField filterDateField;
    private JButton filterButton;
```

```

public VehicleEntryform() {
    setTitle("Vehicle Entry");
    setSize(700, 500);
    setLayout(new BorderLayout());

    // ◆ Entry Form Panel
    JPanel formPanel = new JPanel(new GridLayout(3, 2));
    formPanel.add(new JLabel("Number Plate:"));
    numberField = new JTextField();
    formPanel.add(numberField);

    formPanel.add(new JLabel("Vehicle Type:"));
    typeBox = new JComboBox<>(new String[] {"Car", "Truck", "Bike"});
    formPanel.add(typeBox);

    submitButton = new JButton("Submit");
    formPanel.add(submitButton);

    add(formPanel, BorderLayout.NORTH);

    // ◆ Filter Panel
    JPanel filterPanel = new JPanel(new FlowLayout());
    filterPanel.add(new JLabel("Filter Type:"));
    filterTypeBox = new JComboBox<>(new String[] {"All", "Car", "Truck", "Bike"});
    filterPanel.add(filterTypeBox);

    filterPanel.add(new JLabel("Date (YYYY-MM-DD):"));
    filterTextField = new JTextField(10);
    filterPanel.add(filterTextField);

    filterButton = new JButton("Filter");
}

```

```

filterPanel.add(filterButton);

add(filterPanel, BorderLayout.SOUTH);

// ♦ Table for Vehicle History
tableModel = new DefaultTableModel(new String[]{"Vehicle ID", "Number", "Type",
"Entry Time"}, 0);
historyTable = new JTable(tableModel);
JScrollPane scrollPane = new JScrollPane(historyTable);
add(scrollPane, BorderLayout.CENTER);

// ♦ Actions
submitButton.addActionListener(e -> insertVehicle());
filterButton.addActionListener(e -> applyFilter());

historyTable.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
        int row = historyTable.getSelectedRow();
        if (row != -1) {
            String vehicleNumber = tableModel.getValueAt(row, 1).toString();
            new PaymentEditor(vehicleNumber);
        }
    }
});

```

## MY SQL DATABASE CREATION AND TABLE CREATION CODE

DROP DATABASE IF EXISTS tollgate\_db;

```
CREATE DATABASE tollgate_db; USE tollgate_db;
```

```
CREATE TABLE vehicles (
    vehicle_id INT AUTO_INCREMENT PRIMARY KEY,
    vehicle_number VARCHAR(20),
    type VARCHAR(10),
    entry_time TIMESTAMP
);
```

```
CREATE TABLE tollgates (
    tollgate_id INT AUTO_INCREMENT PRIMARY KEY,
    tollgate_name VARCHAR(50),
    lane_count INT
);
```

```
CREATE TABLE transactions (
    transaction_id INT AUTO_INCREMENT PRIMARY
    KEY, vehicle_id INT,
    tollgate_id INT,
    timestamp TIMESTAMP DEFAULT
    CURRENT_TIMESTAMP, amount DECIMAL(10,2),
    FOREIGN KEY (vehicle_id) REFERENCES vehicles(vehicle_id),
    FOREIGN KEY (tollgate_id) REFERENCES tollgates(tollgate_id)
);
USE tollgate_db;
```

```
INSERT INTO tollgates (tollgate_name, lane_count) VALUES ('Main Gate', 4);
CREATE TABLE users (
```

```
user_id INT AUTO_INCREMENT PRIMARY KEY,  
username VARCHAR(50) UNIQUE,  
password VARCHAR(50)  
);  
  
-- Insert a test user  
INSERT INTO users (username, password) VALUES ('admin', 'admin123');
```

# CHAPTER 5

## SCREEN SHOTS

### JAVA GUI

The screenshot shows an IDE interface with the following details:

- Project:** Tollgate management system
- File:** Main.java
- Code:**

```
1 package com.tollgate;
2
3 public class Main {
4     public static void main(String[] args) {
5         new Loginform(); // Launches the GUI
6     }
7 }
```

- External Libraries:** mysql-connector-j-8.0.22

Fig:5.1

### BACKEND MYSQL DATABASE

The screenshot shows MySQL Workbench with the following details:

- Schemas:** sakila, sys, tollgate\_db, world
- SQL Editor:** SQL updated code
- Code:**

```
16
17
18 * CREATE TABLE transactions (
19     transaction_id INT AUTO_INCREMENT PRIMARY KEY,
20     vehicle_id INT,
21     tollgate_id INT,
22     timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
23     amount DECIMAL(10,2),
24     FOREIGN KEY (vehicle_id) REFERENCES vehicles(vehicle_id),
25     FOREIGN KEY (tollgate_id) REFERENCES tollgates(tollgate_id)
26 )
27 * USE tollgate_db;
28
29 * INSERT INTO tollgates (tollgate_name, lane_count) VALUES ('Main');
30 * CREATE TABLE users (
31     user_id INT AUTO_INCREMENT PRIMARY KEY,
32     username VARCHAR(50) UNIQUE,
33     password VARCHAR(50)
34 );
35
36 -- Insert a test user
37 * INSERT INTO users (username, password) VALUES ('admin', 'admin');
38
39
```

- Action Output:**

| Time        | Action  | Message           | Duration / Fetch |
|-------------|---|-------------------|------------------|
| 5 12:31:09  | CREATE TABLE tollgates ( tollgate_id INT AUT...   | 0 row(s) affected | 0.005 sec        |
| 6 12:31:09  | CREATE TABLE transactions ( transaction_id IN...  | 0 row(s) affected | 0.031 sec        |
| 7 12:31:09  | USE tollgate_db                                   | 0 row(s) affected | 0.000 sec        |
| 8 12:31:09  | INSERT INTO tollgates (tollgate_name, lane_count) | 1 row(s) affected | 0.016 sec        |
| 9 12:31:09  | CREATE TABLE users ( user_id INT AUTO_INCR...     | 0 row(s) affected | 0.000 sec        |
| 10 12:31:09 | INSERT INTO users (username, password) VALUE...   | 1 row(s) affected | 0.000 sec        |

Fig:5.2

## OUTPUT

The screenshot shows a database interface with a toolbar at the top containing various icons. A query is entered in the top bar: `select * from vehicles;`. Below the toolbar is a grid titled "Result Grid" with the following data:

|   | vehicle_id | vehicle_number | type | entry_time          |
|---|------------|----------------|------|---------------------|
| ▶ | 1          | TN-09-K-2789   | Car  | 2025-11-01 12:32:39 |
|   | 2          | TN-18-K-2778   | Car  | 2025-11-01 12:33:16 |

The interface includes a sidebar on the right with icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats". At the bottom right is a "Read Only" button.

Fig:5.3

## GUI VEHICLE ENTRY

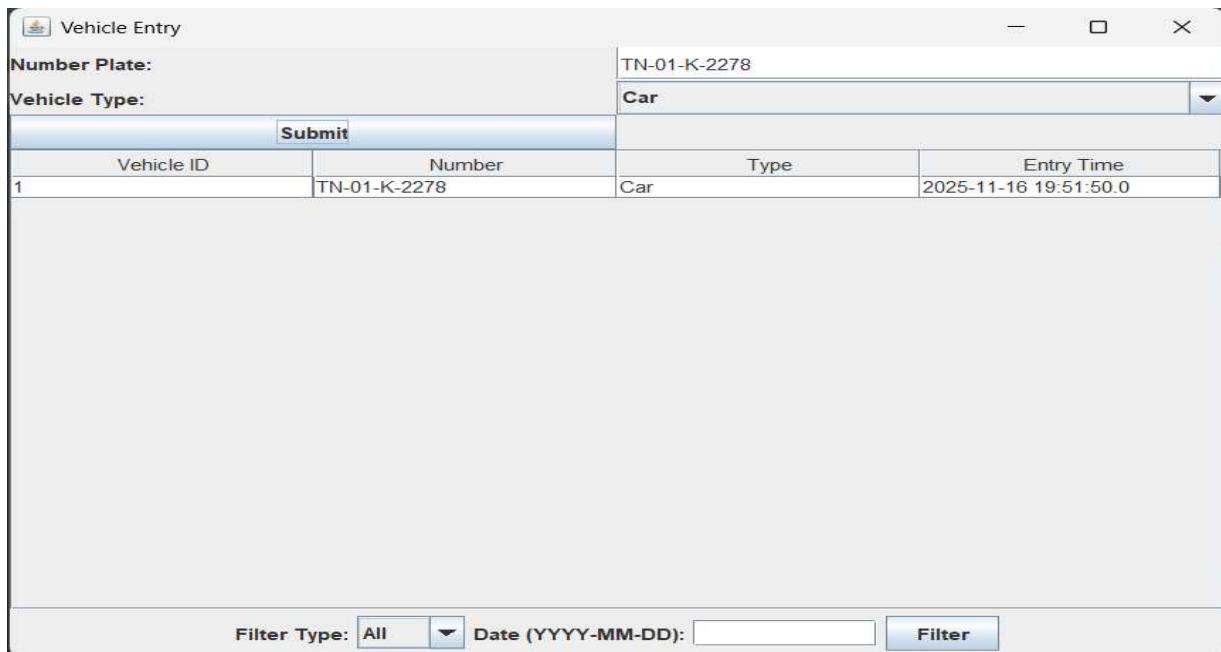


Fig:5.4

## RECIEPT PRINT

Tollgate Receipt

-----

Vehicle Number: TN-01-K-7777

Vehicle Type: Bike

Toll Amount: ₹30.0

Entry Time: 2025-11-05 09:06:39.237

Tollgate ID: 1

-----

Thank you for using our service!

Fig:5.5

## **CHAPTER 6**

### **CONCLUSION AND FUTURE ENHANCEMENT**

In such a way, with the help of our Tollgate Management System, toll operators will be able to easily view the list of vehicle transactions and maintain accurate records of toll collections. The system clearly displays the available data of vehicle entries and exits, along with the respective toll charges, thus making the overall toll management process simpler and more efficient. It also minimizes manual errors and ensures transparency in financial calculations.

In future, this system can be enhanced to support more advanced digital tolling methods, such as complete FASTag automation, ANPR (Automatic Number Plate Recognition) based toll collection, and fully AI-based traffic analysis. The system can also be integrated with national highway data, cloud storage, and real-time traffic monitoring to improve overall road safety and congestion management. Hence, this project provides benefits to both toll administrators and vehicle users in all possible ways and can be expanded into a more intelligent, automated tolling network in the future.

## REFERENCES

1. <https://morth.nic.in/> — MINISTRY OF ROAD TRANSPORT & HIGHWAYS, GOVT. OF INDIA (TOLLING POLICY & FASTAG INFO)
2. <https://www.nhai.gov.in/> — NATIONAL HIGHWAYS AUTHORITY OF INDIA – TOLL OPERATION GUIDELINES
3. <https://www.npci.org.in/what-we-do/netc-fastag> — NPCI NETC FASTAG SYSTEM
4. <https://www.ibm.com/topics/database-management> —DATABASE MANAGEMENT CONCEPTS
5. <https://www.geeksforgeeks.org/dbms/> —DBMS CONCEPTS (NORMALIZATION & ER DIAGRAMS)