VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", MACHHE, BELAGAVI – 590018

Mini Project Report
on

TOLL PLAZA MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the V semester
Bachelor of Engineering
in
Computer Science and Engineering
of
Visvesvaraya Technological University, Belagavi
by

Mr. Nayab Sahil (1CD20CS105)
Mr. Akshay Kumar Singh (1CD20CS009)

Under the Guidance of

Ms. Priyadarshini M
Assistant Professor,
Dept of CSE, CITech

Ms.Bhavana P
Assistant Professor,
Dept of CSE, CITech

Department of Computer Science and Engineering CAMBRIDGE INSTITUTE OF TECHNOLOGY, BENGALURU – 560036
2022-2023

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K.R. Puram, Bengaluru – 560036

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that **Mr. Nayab Sahil** and **Mr. Akshay Kumar Singh** bearing USN **1CD20CS105, 1CD20CS009,** respectively are bonafide students of **Cambridge Institute of Technology,** has successfully completed Mini Project entitled "**Toll Plaza Management System**" in partial fulfillment of the requirements for V semester **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2022 - 2023. It is certified that all Corrections / Suggestions indicated for Internal Assessment have been incorporated in the report. The Database Project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said semester.

Internal Guides: **Head of the Dept.**

1) Ms.Bhavana P
Dept. of CSE., CITech.

Dr. Shashikumar D. R.
Dept. of CSE., CITech.

2) Ms. Priyadarshini M
Dept. of CSE., CITech.

Examiners:

1) **2)**

ACKNOWLEDGEMENT

We are extremely thankful to **Dr. G.Indumathi,** Principal, CITech., Bengaluru, for providing us the academic ambience and everlasting motivation to carry out this work and shaping our careers.

We express our sincere gratitude to **Dr. Shashikumar D. R.,** HOD, Dept. of Computer Science and Engineering, CITech., Bengaluru, for his stimulating guidance, continuous encouragement and motivation throughout the course of present work.

We also wish to extend our thanks to **Ms.Bhavana P,** Assistant Professor, Dept. of Computer Science and Engineering, CITech., Bengaluru and **Ms.Priyadarshini M,** Assistant Professor, Dept. of Computer Science and Engineering, CITech., Bengaluru, for their expert guidance and constructive suggestions to improve the quality of this work.

We would also like to thank all other teaching and technical staffs of Department of Computer Science and Engineering, who have directly or indirectly helped us in the completion of this Project Work.

And lastly we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in the successful completion of this project.

Nayab Sahil (1CD20CS105)
Akshay Kumar Singh (1CD20CS009)

ABSTRACT

The toll plaze management system is made to automatically record the time, the location, and information about the owner of the car, such as name, date of registration, and vehicle model. This system is excellent for automating toll gates, managing time, and automatically tracking vehicles. Online toll gate management solutions have been very helpful in reducing the excessive traffic that is now a feature of urban cities. It is one of the straightforward approaches of controlling the heavy flow of traffic. The passengers using this form of transportation are transported by their means of transportation, which enables them to be aware of the account of money that has been paid and the money remaining in the tag. It frees the passenger from having to wait in line to pay the toll, which reduces fuel usage and also allows them to avoid carrying cash. Such issues are avoided by our system. User doesn't need to wait at tollgate because they can get a gate pass online.

CONTENTS

CHAPTER 1

INTRODUCTION

Toll Plaza Management System is a web-based application that can provide all the information related to toll plazas and the passenger checks in either online or on a mobile device and pays the amount, then he/she will be provided by a receipt. The travelers passing through this mode of transport, carried by their transport that allows them to be aware of the account of money that has been paid and the money left in the tag. It relieves the traveler of the burden of waiting in the queue to make the toll payment, which decreases the fuel-consumption and also taking cash with them can be avoided. Our system avoids this type of problems. user get gate pass from online so user don’t need to wait in tollgate. There are two ways of collecting toll tax being in practice at present. First is the traditional manual method where one person collects the money and issues a receipt. The other one is Smart Card system where the person needs to show the smart card to the system installed at the toll tax depot to open the barrier.

CHAPTER 2

REQUIREMENTS

2.1 Software Requirement Specifications

- Operating System : Windows 11
- Frontend Software: IntelliJ IDEA Community Edition 2022.2.1
- Backend Software: MySQL Shell 8.0.31

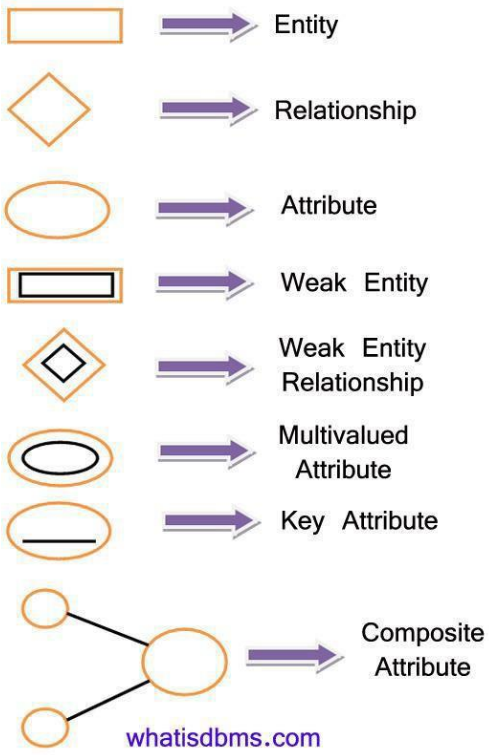
2.2 Hardware Requirement Specifications

- Computer Processor Core i5
- Processor Speed 2.3 GHz Processor
- Hard Disk INSTALLED
- RAM 8GB

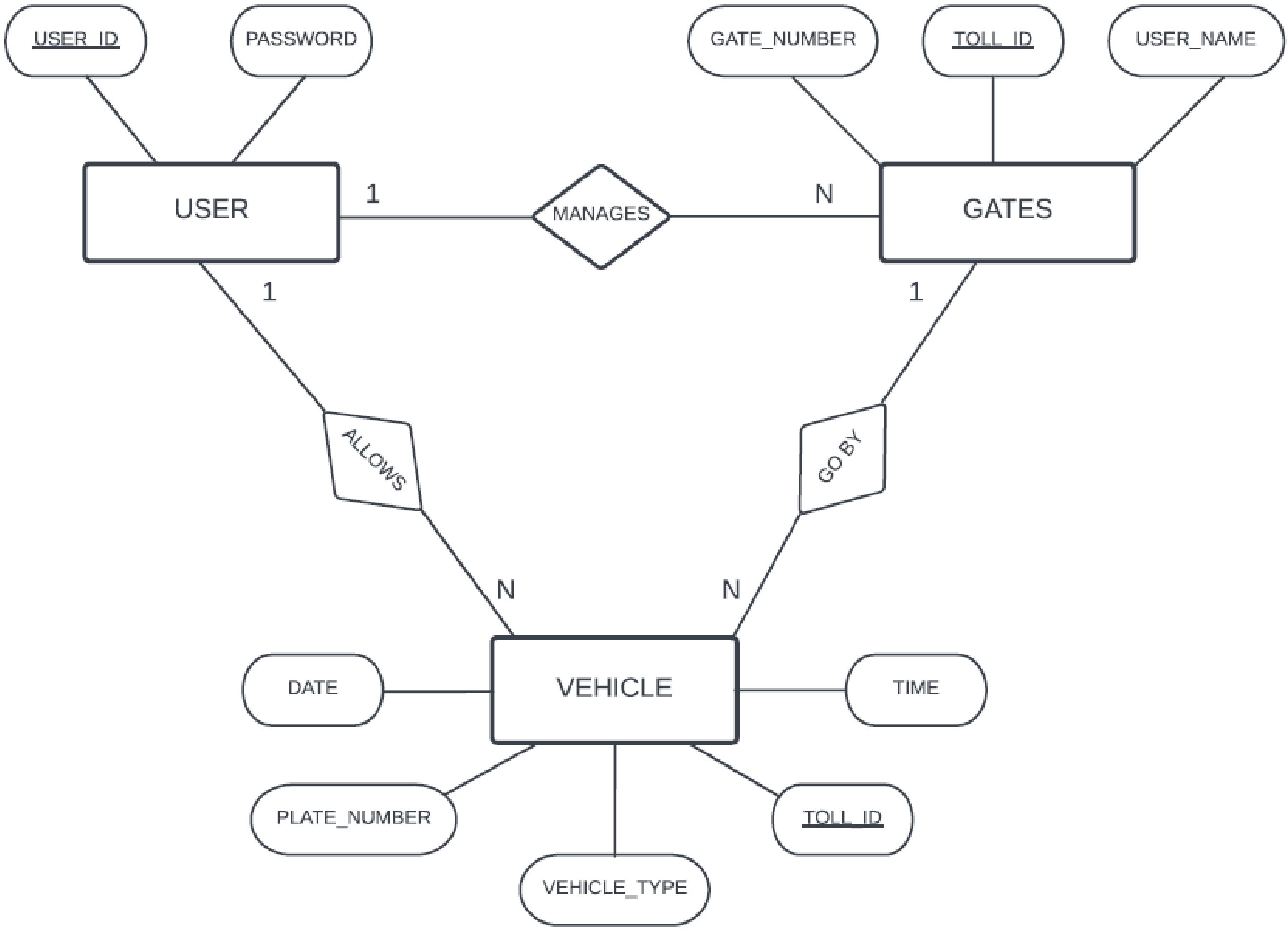
CHAPTER 3

ENTITY RELATIONSHIP DIAGRAM

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. Although data modelling has become a necessity around 1970's there was no standard way to model databases or business processes. Although many solutions were proposed and discussed none were widely adopted. Peter Chen is credited with introducing the widely adopted ER model in his paper "The Entity Relationship Model-Toward a Unified View of Data". The focus was on entities and relationships and he introduced a diagramming representation for database design as well. They are widely used to design relational databases. The entities in the ER schema become tables, attributes and converted the database schema. Since they can be used to visualize database tables and their relationships it's commonly used for database troubleshooting as well. Entity relationship diagrams are used in software engineering during the planning stages of the software project. They help to identify different system elements and their relationships with each other. It is often used as the basis for data flow diagrams or DFD's as they are commonly known.



ER DIAGRAM


USER_ID
PASSWORD
GATE_NUMBER
TOLL_ID
USER_NAME
USER
1
MANAGES
N
GATES
1
1
ALLOWS
GO BY
N
N
DATE
VEHICLE
TIME
PLATE_NUMBER
TOLL_ID
VEHICLE_TYPE

CHAPTER 4

SCHEMA DIAGRAM

4.1 SCHEMA DIAGRAM

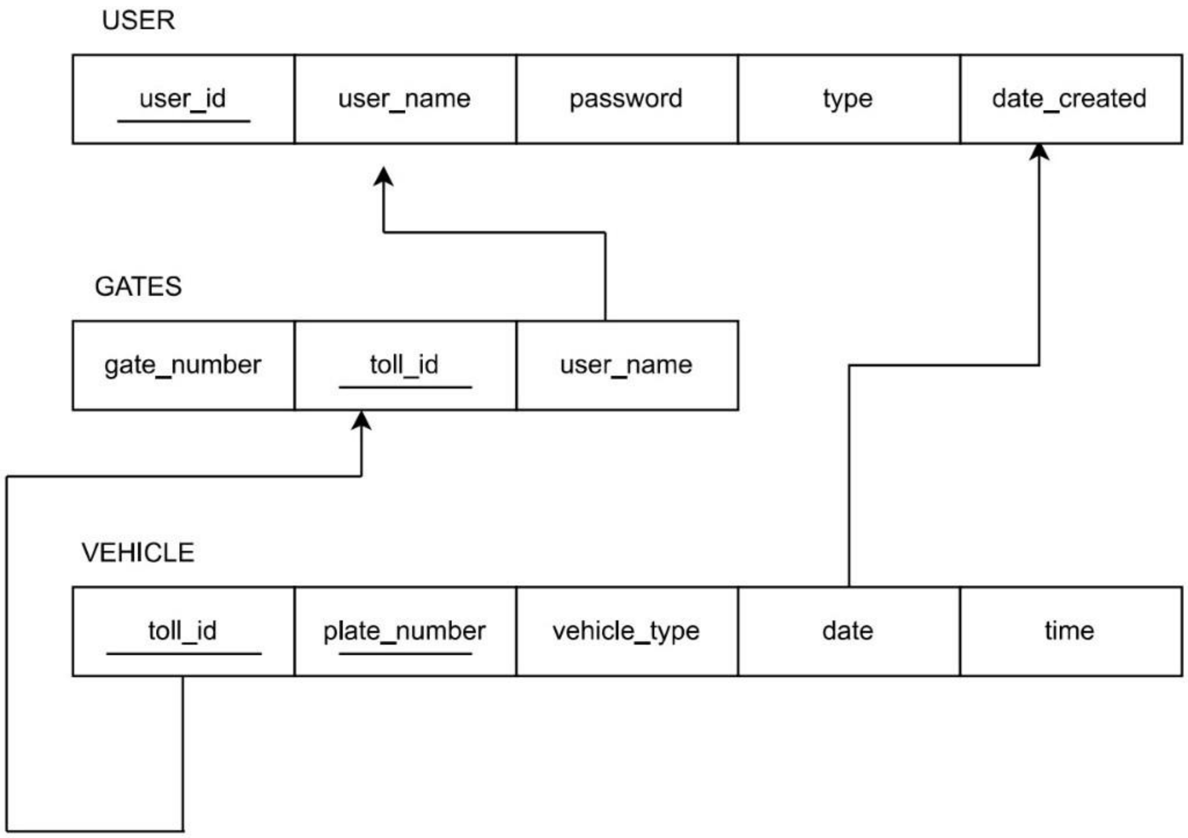
A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It’s the database designers who design the schema to help programmers understand the database and make it useful. It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information.

A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed.

A database schema can be divided broadly into two categories –

- Physical Database Schema − This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- Logical Database Schema − This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

SCHEMA DIAGRAM



CHAPTER 5

IMPLEMENTATION

5.1 Backend Implementation (database)

VEHICLE TABLE:

CREATE TABLE VEHICLE(

TOLLID VARCHAR(20),

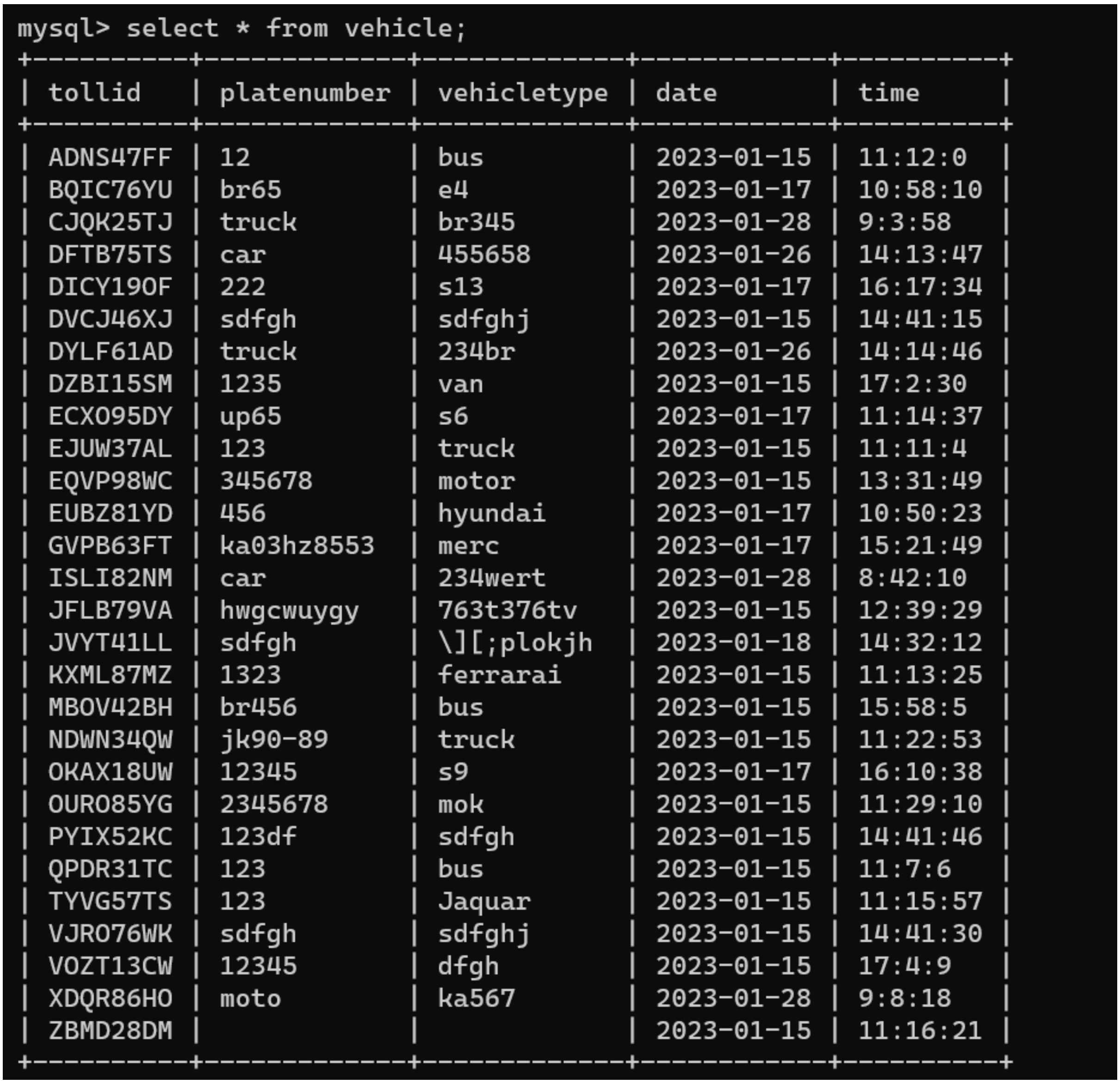
PLATENUMBER VARCHAR(20),

VEHICLETYPE,

DATE DATE,

TIMEVARCHAR(20));

INSERT INTO VEHICLE VALUES(‘ADNS47FF’, ‘2’, ‘BUS’, ‘2023-01-15’, ‘11:12:0’);



```
mysql> select * from vehicle;
+----------+-------------+-------------+------------+----------+
| tollid   | platenumber | vehicletype | date       | time     |
+----------+-------------+-------------+------------+----------+
| ADNS47FF | 12          | bus         | 2023-01-15 | 11:12:0  |
| BQIC76YU | br65        | e4          | 2023-01-17 | 10:58:10 |
| CJQK25TJ | truck       | br345       | 2023-01-28 | 9:3:58   |
| DFTB75TS | car         | 455658      | 2023-01-26 | 14:13:47 |
| DICY19OF | 222         | s13         | 2023-01-17 | 16:17:34 |
| DVCJ46XJ | sdfgh       | sdfghj      | 2023-01-15 | 14:41:15 |
| DYLF61AD | truck       | 234br       | 2023-01-26 | 14:14:46 |
| DZBI15SM | 1235        | van         | 2023-01-15 | 17:2:30  |
| ECXO95DY | up65        | s6          | 2023-01-17 | 11:14:37 |
| EJUW37AL | 123         | truck       | 2023-01-15 | 11:11:4  |
| EQVP98WC | 345678      | motor       | 2023-01-15 | 13:31:49 |
| EUBZ81YD | 456         | hyundai     | 2023-01-17 | 10:50:23 |
| GVPB63FT | ka03hz8553  | merc        | 2023-01-17 | 15:21:49 |
| ISLI82NM | car         | 234wert     | 2023-01-28 | 8:42:10  |
| JFLB79VA | hwgcwuygy   | 763t376tv   | 2023-01-15 | 12:39:29 |
| JVYT41LL | sdfgh       | \][;plokjh  | 2023-01-18 | 14:32:12 |
| KXML87MZ | 1323        | ferrarai    | 2023-01-15 | 11:13:25 |
| MBOV42BH | br456       | bus         | 2023-01-15 | 15:58:5  |
| NDWN34QW | jk90-89     | truck       | 2023-01-15 | 11:22:53 |
| OKAX18UW | 12345       | s9          | 2023-01-17 | 16:10:38 |
| OURO85YG | 2345678     | mok         | 2023-01-15 | 11:29:10 |
| PYIX52KC | 123df       | sdfgh       | 2023-01-15 | 14:41:46 |
| QPDR31TC | 123         | bus         | 2023-01-15 | 11:7:6   |
| TYVG57TS | 123         | Jaquar      | 2023-01-15 | 11:15:57 |
| VJRO76WK | sdfgh       | sdfghj      | 2023-01-15 | 14:41:30 |
| VOZT13CW | 12345       | dfgh        | 2023-01-15 | 17:4:9   |
| XDQR86HO | moto        | ka567       | 2023-01-28 | 9:8:18   |
| ZBMD28DM |             |             | 2023-01-15 | 11:16:21 |
+----------+-------------+-------------+------------+----------+
```

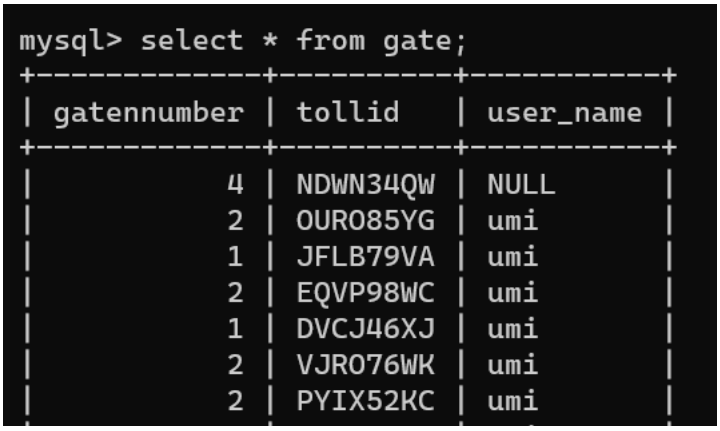
GATE TABLE:

CREATE TABLE GATE(

GATENUMBER INT,

TOLLID VARCHAR(20),USER_NAME VARCHAR(20));

INSERT INTO GATE VALUES(‘4’, ‘ADNS47FF’, ‘UMI’);



```
mysql> select * from gate;
+-------------+----------+-----------+
| gatennumber | tollid   | user_name |
+-------------+----------+-----------+
|           4 | NDWN34QW | NULL      |
|           2 | OURO85YG | umi       |
|           1 | JFLB79VA | umi       |
|           2 | EQVP98WC | umi       |
|           1 | DVCJ46XJ | umi       |
|           2 | VJRO76WK | umi       |
|           2 | PYIX52KC | umi       |
```

USERS TABLE:

CREATE TABLE USERS(

ID VARCHAR(20),

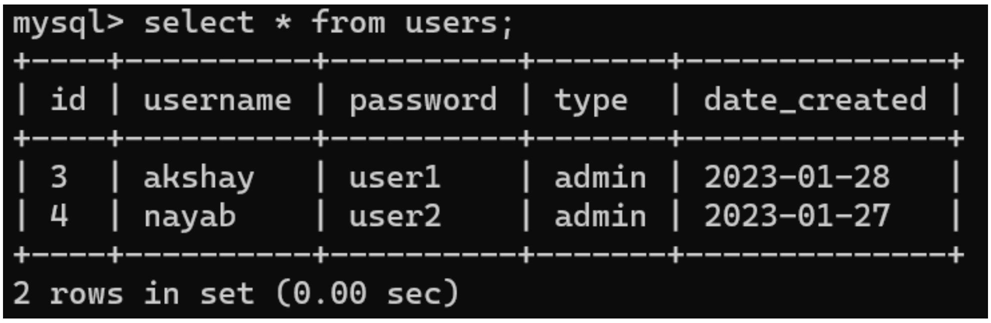
USERNAME VARCHAR(20),

PASSWORD VARCHAR(20),

TYPE VARCHAR(20),

DATE_CREATED DATE));

INSERT INTO USERS VALUES(‘3’, ‘AKSHAY’, ‘USER1’, ‘ADMIN’, ‘2023-01-28’);



```
mysql> select * from users;
+----+----------+----------+-------+--------------+
| id | username | password | type  | date_created |
+----+----------+----------+-------+--------------+
| 3  | akshay   | user1    | admin | 2023-01-28   |
| 4  | nayab    | user2    | admin | 2023-01-27   |
+----+----------+----------+-------+--------------+
2 rows in set (0.00 sec)
```

5.2 Frontend Implementation

The backend and frontend is connected through JDBC. The frontend entry is done using Swings. Add, Delete, Update, Next and Done Buttons are inserted using drag and drop method. Once the records are added, deleted or updated the corresponding operation message is displayed using showMessageDialog() function of JOptionPane class. One JFrame can be moved to another using Next Button

CODE:

CONNECT:

```
package Project;
import java.sql.*;
public class Connect {
    private static final String Database="jdbc:mysql://localhost:3306/toll_plaza";
    private static final String userName ="root";
    private static final String passwd="akshay7061@45";
    Statement sql;
    Connection con;
    Connect() throws  Exception
    {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("Trying to connect");
            con=DriverManager.getConnection(Database,userName,passwd);
            System.out.println("Connected");
            sql= con.createStatement();

    }
    public static void main(String[] args) throws Exception {
        new Connect();
    }
}
```

LOGIN PAGE:

```
package Project;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class first extends JFrame implements ActionListener {
    public JTextField t1;
    public JPasswordField p1;
    Connect con=null;
   static first App;
    first()
    {
        super("login");
        setSize(300, 300);
        setResizable(true);
        Container c = getContentPane();
        c.setBackground(Color.black);
        c.setLayout(new GridLayout(4, 1));
        JPanel tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l = new JLabel("LOGIN");
        l.setFont(new Font("Impact", Font.BOLD, 32));
        l.setForeground(Color.black);
        l.setFont(new Font("Impact", Font.BOLD, 32));
        tp.add(l);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l1 = new JLabel("USERID");
        l1.setForeground(Color.black);
        t1 = new JTextField(20);
        tp.add(l1);
        tp.add(t1);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l2 = new JLabel("PASSWORD");
        l2.setForeground(Color.black);
        p1 = new JPasswordField(20);
        tp.add(l2);
        tp.add(p1);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JButton b1 = new JButton("SIGN IN");
        b1.setForeground(Color.black);
        b1.addActionListener(this);
        tp.add(b1);
        add(tp);
        setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e)
    {
        String userid = t1.getText();
        String password = String.valueOf(p1.getPassword());
        System.out.println(userid + " " + password);
        if (userid.length() < 3 || password.length() < 3)
        {
            JOptionPane.showMessageDialog(this, "Please enter user name and password");
            return;
        }
        try {
            if(con==null)
                con = new Connect();
            ResultSet rs=con.sql.executeQuery(" select * from users where
username='"+userid+"' and password='"+password+"'");
            if(rs.next())
                new second(userid);
            else
                JOptionPane.showMessageDialog(this, "invalid user");
        }
        catch (Exception ee) {
            con=null;
            System.out.println(ee);
            JOptionPane.showMessageDialog(this, "server down");
        }
    }
    public static void main(String[] args)
    {
        App=new first();
    }
}
```

GATES PAGE:

```
package Project;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class second extends JFrame implements ActionListener {
    private JButton b1,b2,b3,b4,b5;
    String userid;
    second( String userid)
    {
        super("gates");
        this.userid=userid;
        setSize(400, 400);
        setResizable(false);
        Container c = getContentPane();
        c.setBackground(Color.black);
        c.setLayout(new GridLayout(4,1));
        JPanel tp=new JPanel();
        tp.setBackground(Color.white);
        JLabel ll= new JLabel("gates");
        ll.setForeground(Color.black);
        ll.setFont(new Font("Impact",Font.BOLD,32));
        tp.add(ll);
        add(tp);
        tp=new JPanel();
        tp.setBackground(Color.white);
        b1=new JButton("gate1");
        b1.addActionListener( this);
        b2=new JButton("gate2");
        b2.addActionListener(this);
        tp.add(b1);
        tp.add(b2);
        add(tp);
        tp=new JPanel();
        tp.setBackground(Color.white);
        b3=new JButton("gate3");
        b3.addActionListener(this);
        b4=new JButton("gate4");
        b4.addActionListener(this);
        tp.add(b3);
        tp.add(b4);
        add(tp);
        tp=new JPanel();
        tp.setBackground(Color.white);
        b5=new JButton("search");
        b5.addActionListener(this);
        tp.add(b5);
        add(tp);
        setVisible(true);
    }
    public static void main(String[] args)
    {
        //new second();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
       if(e.getSource()==b1)
       {
           new third("1",userid);
       }
       if(e.getSource()==b2)
        {
           new third("2",userid);
        }
       if(e.getSource()==b3)
        {
           new third("3",userid);
        }
        if(e.getSource()==b4)
        {
           new third("4",userid);
        }
        if(e.getSource()==b5)
        {
           new fourth();
        }
    }
}
```

ADD:

```
package Project;
import javax.swing.*;
import java .awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.*;
import java.time.LocalDate;
import java.time.LocalTime;
import java.util.Locale;
import java.util.Random;
public class third extends JFrame implements ActionListener {
    public static JTextField t1, t2, t3,d1,ti;
    String gate;
    String userid;
    third(String gate, String userid) {
        super("REGISTRATION");
        this.gate=gate;
        this.userid=userid;
        setSize(500, 800);
        setResizable(true);
        Container c = getContentPane();
        c.setBackground(Color.WHITE);
        c.setLayout(new GridLayout(7, 2));
        JPanel tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l1 = new JLabel("CAR REGISTRATION");
        l1.setForeground(Color.black);
        l1.setFont(new Font("Impact", Font.BOLD, 32));
        tp.add(l1);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l2 = new JLabel("VEHICLE TYPE");
        l2.setForeground(Color.black);
        t1 = new JTextField("", 20);
        tp.add(l2);
        tp.add(t1);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l3 = new JLabel("PLATE NUMBER");
        l3.setForeground(Color.black);
        t2 = new JTextField("", 20);
        tp.add(l3);
        tp.add(t2);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l5 = new JLabel("TOLL ID");
        l2.setForeground(Color.black);
        t3 = new JTextField(tollGen(), 20);
        tp.add(l5);
        tp.add(t3);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l4 = new JLabel("DATE");
        l4.setForeground(Color.black);
        LocalDate dat=LocalDate.now();
        String date=dat.getYear()+"-"+dat.getMonthValue()+"-"+dat.getDayOfMonth();
        d1 = new JTextField(date, 20);
        tp.add(l4);
        tp.add(d1);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l6 = new JLabel("TIME");
        l6.setForeground(Color.black);
        LocalTime tim=LocalTime.now();
        String time=tim.getHour()+":"+tim.getMinute()+":"+tim.getSecond();
        ti = new JTextField(time, 20);
        tp.add(l6);
        tp.add(ti);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JButton b1 = new JButton("ADD");
        b1.addActionListener(this);
        tp.add(b1);
        add(tp);
        setVisible(true);
    }
    public String tollGen() {
        Random r = new Random();
        String x = "";
        for (int i = 0; i <= 3; i++)
        {
            int k = r.nextInt(26) + 65;
            x += (char) k;

        }
        x += (r.nextInt(9) + 1);

        x += (r.nextInt(9) + 1);
        for (int i = 0; i < 2; i++) {

            int k = r.nextInt(26) + 65;
```

```
          x += (char) k;
       }
       return x;
   }
   public static void main(String[] args) {
       new third("9","xyx");
   }
   @Override
   public void actionPerformed(ActionEvent e) {
       String plat=t1.getText();
       String type=t2.getText();
       String tollid=t3.getText();
       String date=d1.getText();
       String time=ti.getText();
       String sql="INSERT INTO VEHICLE VALUES(?,?,?,?,?) ";
       PreparedStatement stm=null;
       try {
           stm=  first.App.con.con.prepareStatement(sql);
           stm.setString(1,tollid);
           stm.setString(2,plat);
           stm.setString(3,type);
           stm.setString(4,date);
           stm.setString(5,time);
          if( stm.executeUpdate()==0)
          {
              JOptionPane.showMessageDialog(this,"");
              return;
          }
           JOptionPane.showMessageDialog(this,"inserted");
           sql="INSERT INTO GATE VALUES(?,?,?) ";
           stm=null;
           stm=  first.App.con.con.prepareStatement(sql);
           stm.setString(2,tollid);
           stm.setString(1,gate);
           stm.setString(3,userid);
           stm.executeUpdate();
              this.dispose();
       }
       catch (SQLException ex)
       {
           JOptionPane.showMessageDialog(this,"Please try again");
       }
   }
}
```

SEARCH:

```
package Project;
import javax.swing.*;
import java .awt.*;
import java.awt.event.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
public class fourth extends JFrame implements ActionListener {
    JLabel l1;
    JTextField t1;
    JButton b1;
    fourth()
    {
        super("SEARCH");
        setSize(300, 300);
        setResizable(true);
        Container c = getContentPane();
        c.setBackground(Color.black);
        c.setLayout(new GridLayout(3, 1));
        JPanel tp = new JPanel();
        l1 = new JLabel("SEARCH");
        l1.setFont(new Font("Impact", Font.BOLD, 32));
        tp.add(l1);
        add(tp);
        tp = new JPanel();
        l1= new JLabel("PLATE NUMBER");
        t1 = new JTextField("", 20);
        tp.add(l1);
        tp.add(t1);
        add(tp);
        tp = new JPanel();
        JButton b1 = new JButton("FIND");
        b1.addActionListener(this);
        tp.add(b1);
        add(tp);
        setVisible(true);
    }
    @Override
    public void actionPerformed (ActionEvent e)
    {
      Connect con=null;
      PreparedStatement p=null;
      ResultSet rs=null;
        String sql="select * from vehicle where platenumber='"+(t1.getText())+"'";
        try
        {
           rs=  first.App.con.sql.executeQuery(sql);
            if(rs.next())
            {
                new
fifth(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5));
            }
            else
            {
```

```
            JOptionPane.showMessageDialog(this,"No result found");
          }
        }
        catch (Exception ex)
        {
          throw new RuntimeException(ex);
        }
    }
    public static void main(String[] args)
    {
        new fourth();
    }
}
```

UPDATE :

```
package Project;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class fifth extends  JFrame implements ActionListener
{
    JTextField t1, t2, t3,t4,t5;
    String tollId;
    fifth(String tollid,String platenum,String type,String date,String time)
    {
        super("SEARCH DETAILS");
        tollId=tollid;
        setSize(400, 600);
        setResizable(false);
        Container c = getContentPane();
        setLayout(new GridLayout(7, 2));
        setBackground(Color.BLACK);
        JPanel tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l = new JLabel("SEARCH DETAILS");
        l.setFont(new Font("Impact", Font.BOLD, 32));
        l.setForeground(Color.black);
        tp.add(l);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l1 = new JLabel("TOLL ID");
        l1.setForeground(Color.black);
        t1 = new JTextField(tollid, 20);
        t1.setEditable(false);
        tp.add(l1);
        tp.add(t1);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l2 = new JLabel("CAR ");
        l1.setForeground(Color.black);
        t2 = new JTextField(platenum, 20);
        tp.add(l2);
        tp.add(t2);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l3 = new JLabel("MODEL");
        l3.setForeground(Color.black);
        t3 = new JTextField(type, 20);
        tp.add(l3);
        tp.add(t3);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l4 = new JLabel("DATE");
        l4.setForeground(Color.black);
        t4 = new JTextField(date, 20);
        t4.setEditable(false);
        tp.add(l4);
        tp.add(t4);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JLabel l5 = new JLabel("TIME");
        l5.setForeground(Color.black);
        t5 = new JTextField(time, 20);
        tp.add(l5);
        tp.add(t5);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);        JButton b1 = new JButton("MODIFY");
        b1.setName("modify");
        b1.addActionListener(this);
        tp.add(b1);
        add(tp);
        setVisible(true);
    }

    public static void main(String[] args)
    {
        // new fifth();
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        JButton btn =(JButton) e.getSource();
        if(btn.getName().equals("modify"))
        {
            PreparedStatement stm=null;
            System.out.println("hello");
            String sql ="update vehicle set platenumber= ?,vehicletype=? ,time=? where
tollid=?";
            try
            {
                stm = first.App.con.con.prepareStatement(sql);
                stm.setString(1, t2.getText());
                stm.setString(2, t3.getText());
                stm.setString(3, t5.getText());
                stm.setString(4, tollId);
                if(stm.executeUpdate()==0)
                {
                    JOptionPane.showMessageDialog(this,"Please try again");
                    return;
                }
                JOptionPane.showMessageDialog(this,"Details updated");
                this.dispose();
            }
            catch (Exception ee)
            {
                System.out.println(ee);
            }

            return;
        }
    }
}
```

DELETE:

```
package Project;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class fifth extends  JFrame implements ActionListener
{
    JTextField t1, t2, t3,t4,t5;
    String tollId;
    fifth(String tollid,String platenum,String type,String date,String time)
    {
        super("SEARCH DETAILS");
        tollId=tollid;
```

```
setSize(400, 600);
setResizable(false);
Container c = getContentPane();
setLayout(new GridLayout(7, 2));
setBackground(Color.BLACK);
JPanel tp = new JPanel();
tp.setBackground(Color.white);
JLabel l = new JLabel("SEARCH DETAILS");
l.setFont(new Font("Impact", Font.BOLD, 32));
l.setForeground(Color.black);
tp.add(l);
add(tp);
tp = new JPanel();
tp.setBackground(Color.white);
JLabel l1 = new JLabel("TOLL ID");
l1.setForeground(Color.black);
t1 = new JTextField(tollid, 20);
t1.setEditable(false);
tp.add(l1);
tp.add(t1);
add(tp);
tp = new JPanel();
tp.setBackground(Color.white);
JLabel l2 = new JLabel("CAR ");
l1.setForeground(Color.black);
t2 = new JTextField(platenum, 20);
tp.add(l2);
tp.add(t2);
add(tp);
tp = new JPanel();
tp.setBackground(Color.white);
JLabel l3 = new JLabel("MODEL");
l3.setForeground(Color.black);
t3 = new JTextField(type, 20);
tp.add(l3);
tp.add(t3);
add(tp);
tp = new JPanel();
tp.setBackground(Color.white);
JLabel l4 = new JLabel("DATE");
l4.setForeground(Color.black);
t4 = new JTextField(date, 20);
t4.setEditable(false);
tp.add(l4);
tp.add(t4);
add(tp);
tp = new JPanel();
tp.setBackground(Color.white);
JLabel l5 = new JLabel("TIME");
l5.setForeground(Color.black);
t5 = new JTextField(time, 20);
```

```
        tp.add(l5);
        tp.add(t5);
        add(tp);
        tp = new JPanel();
        tp.setBackground(Color.white);
        JButton b3 = new JButton("DELETE");
        b3.setName("delete");
        b3.addActionListener(this);
        tp.add(b3);
        add(tp);
        setVisible(true);
    }

    public static void main(String[] args)
    {
       // new fifth();
    }
        try
        {

            if(first.App.con.sql.executeUpdate("delete from vehicle where tollid='"+tollId+"'")>0)
            {
                first.App.con.sql.executeUpdate("delete from gate where tollid='"+tollId+"'");
                JOptionPane.showMessageDialog(this,"Deleted");
                this.dispose();
                return;
            }
            JOptionPane.showMessageDialog(this,"please try again");

        } catch (SQLException ex) {
            System.out.println(ex);
        }
    }
}
```

CHAPTER 6

SNAPSHOTS

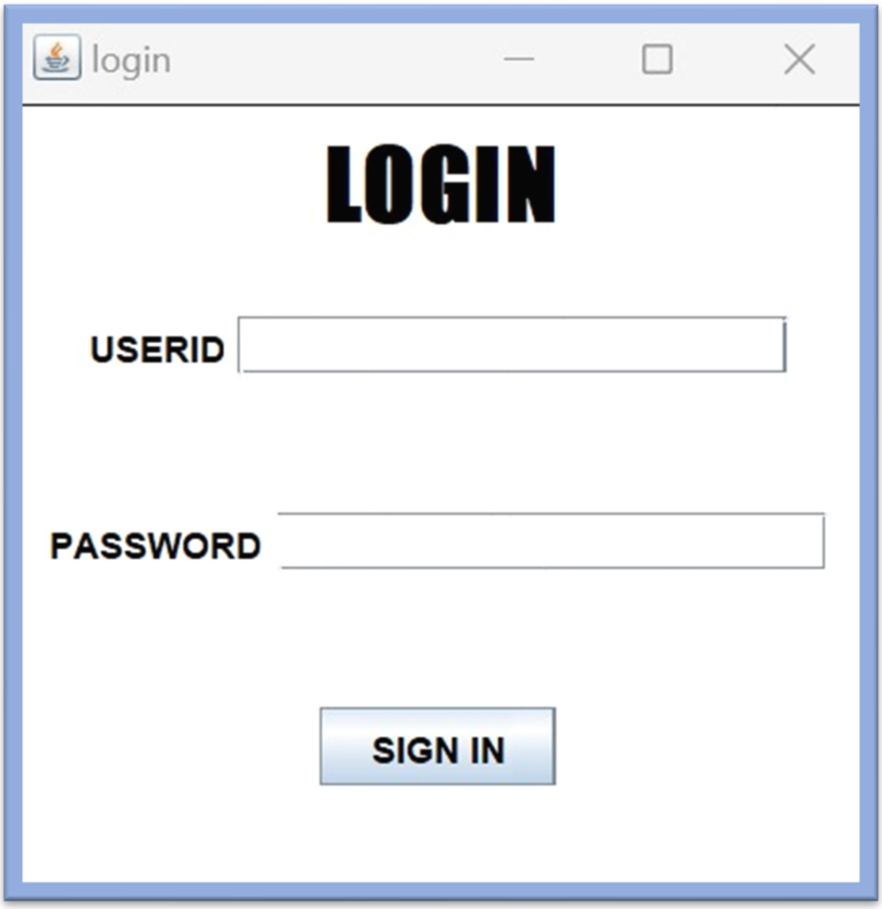


Figure 6.1: Login

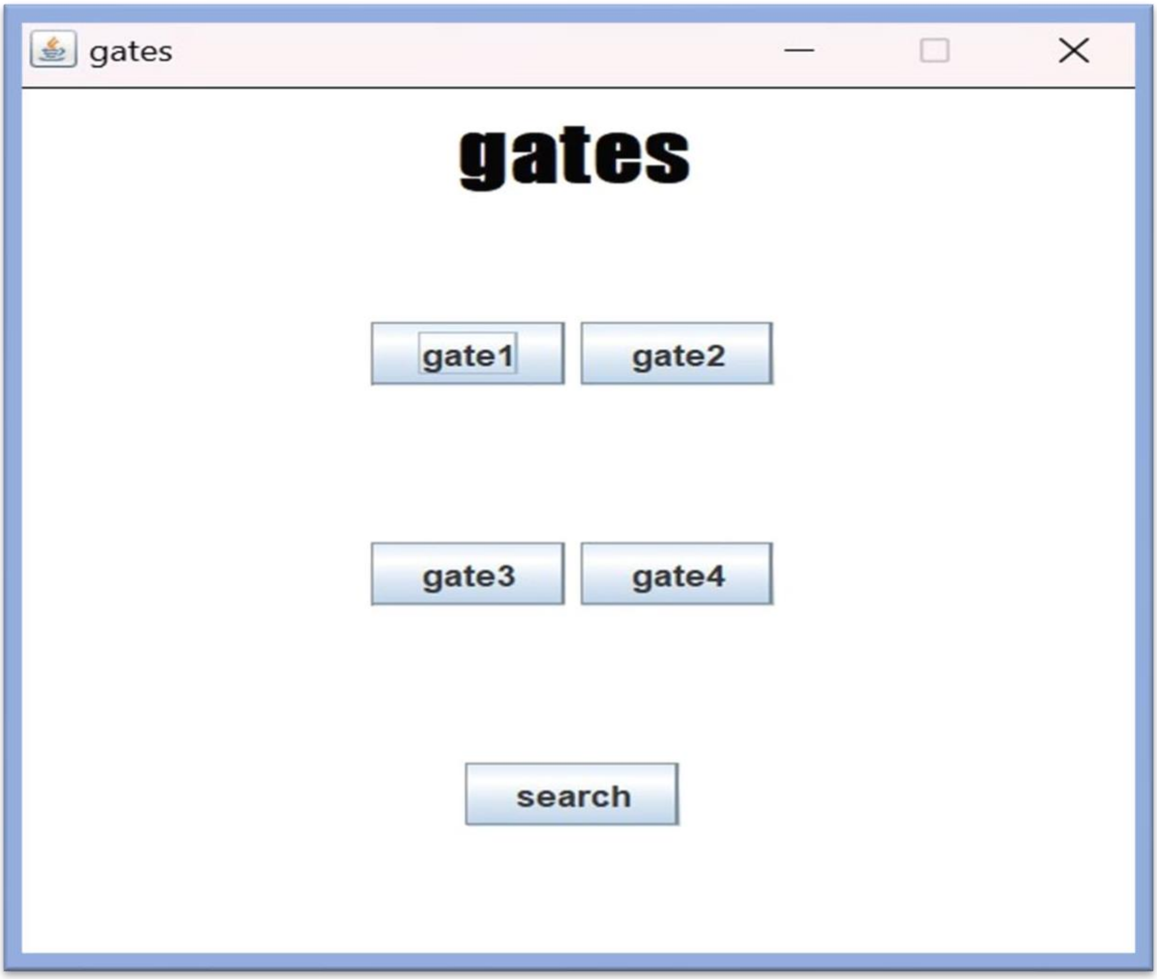
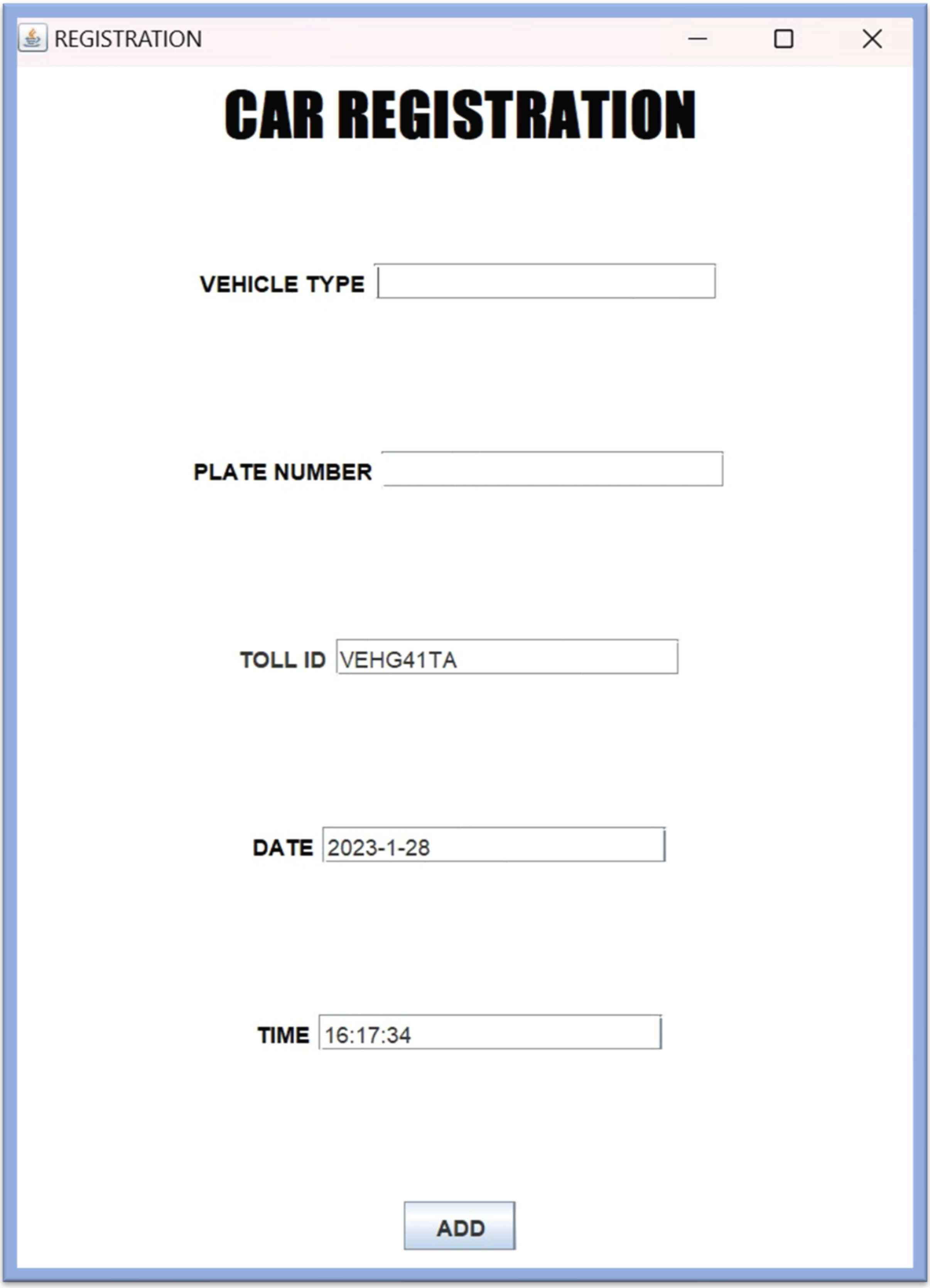


Figure 6.2: Gates



REGISTRATION

CAR REGISTRATION

VEHICLE TYPE

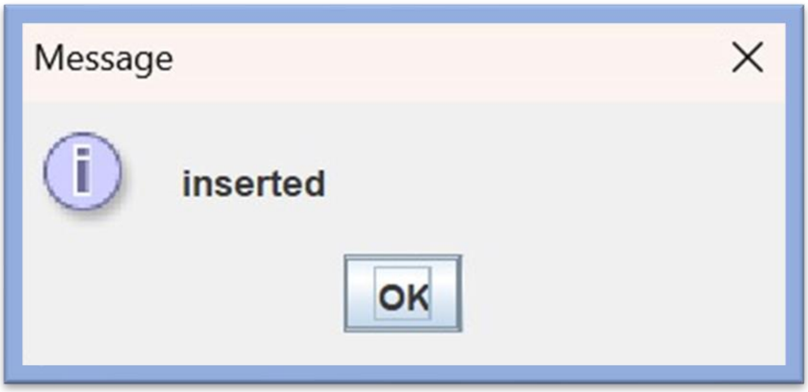
PLATE NUMBER

TOLL ID VEHG41TA

DATE 2023-1-28

TIME 16:17:34

Figure 6.3: Car Registration



Message

inserted

Figure 6.4: Inserted

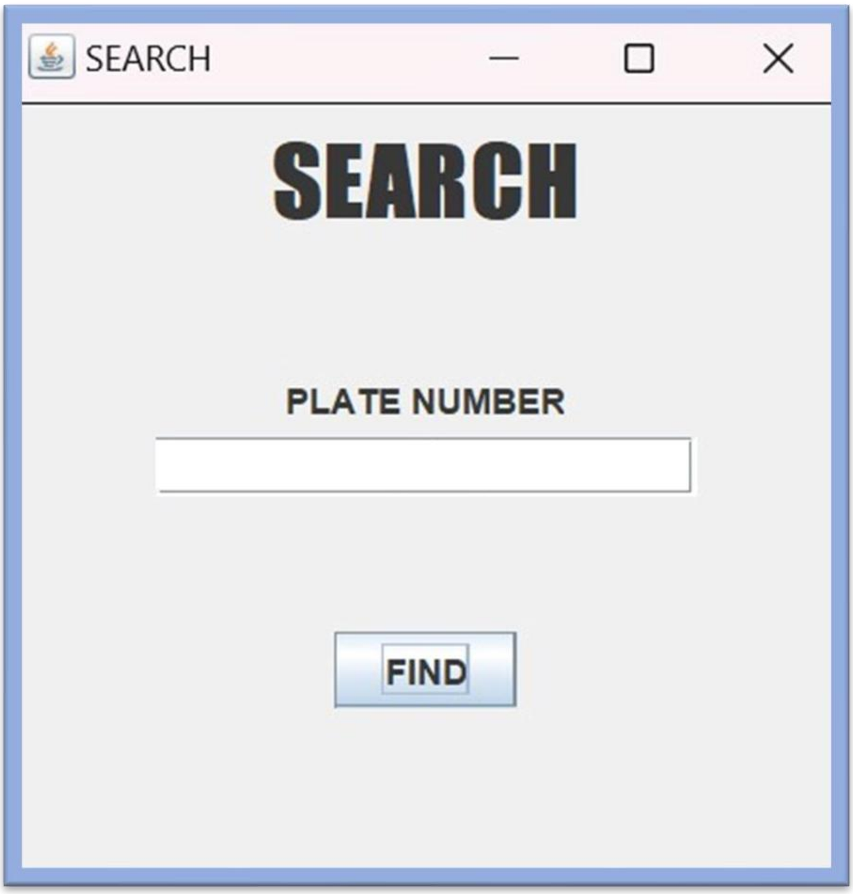
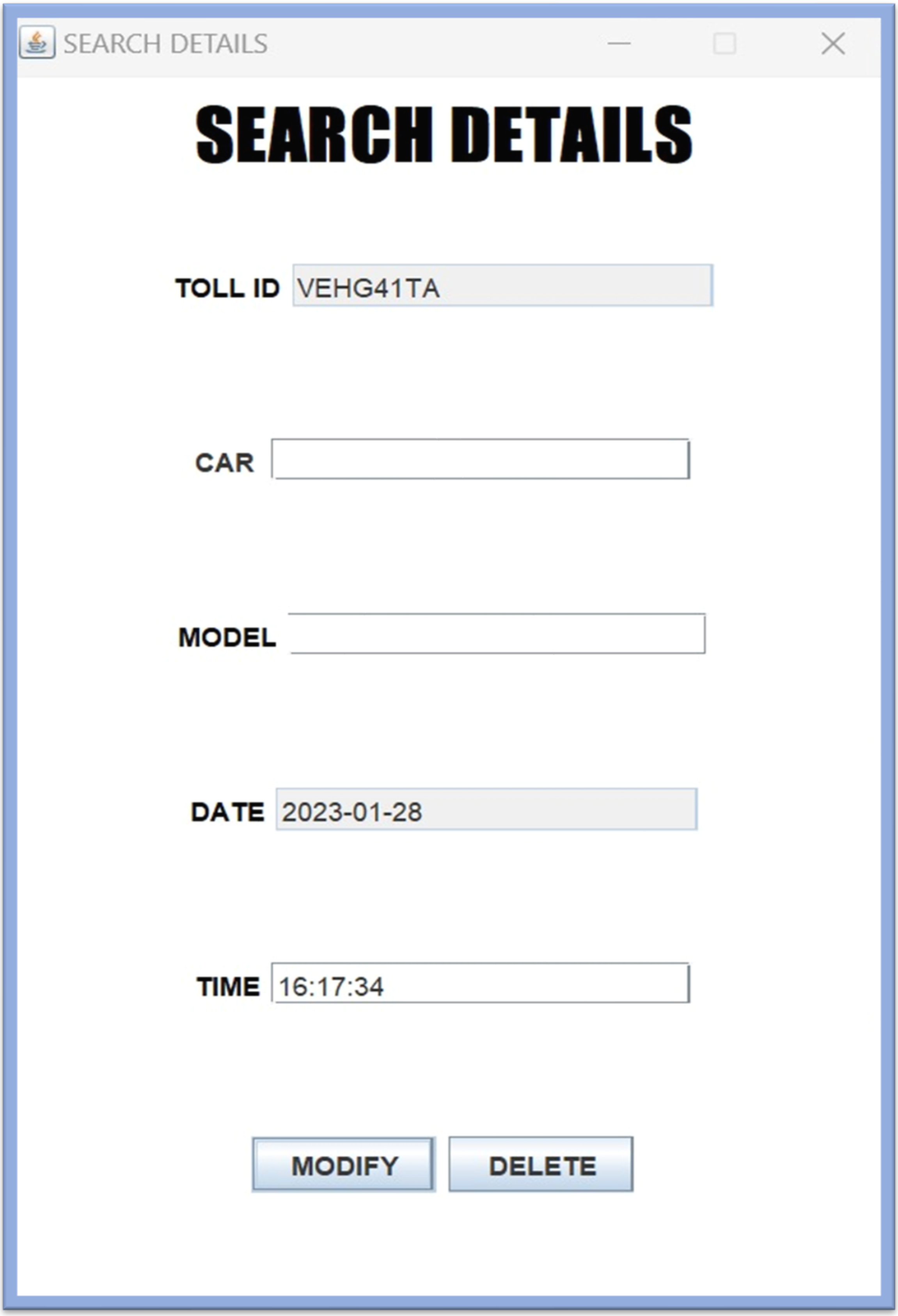


Figure 6.5: Search



SEARCH DETAILS

SEARCH DETAILS

TOLL ID: VEHG41TA

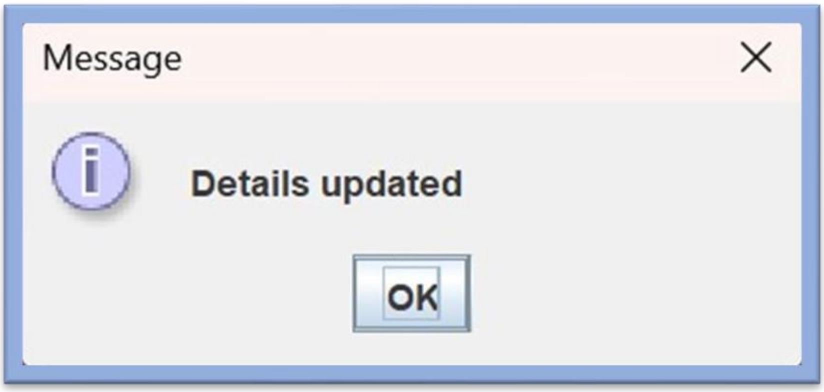
CAR:

MODEL:

DATE: 2023-01-28

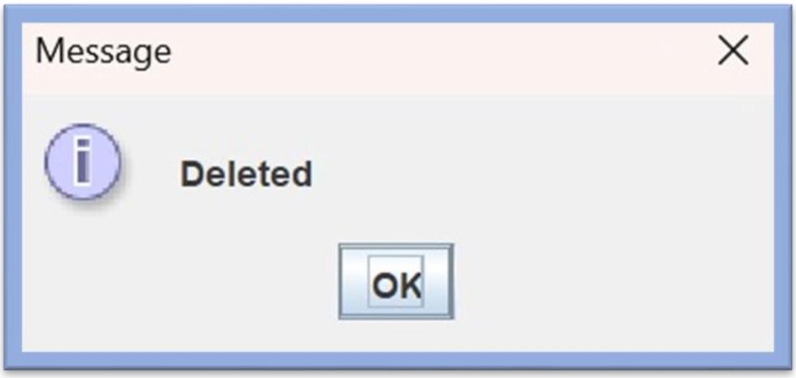
TIME: 16:17:34

Figure 6.6: Search Details



Details updated

Figure 6.7: Update



Deleted

Figure 6.8: Delete

CONCLUSION

Toll plaza management is the process of collecting, monitoring, and managing tolls from vehicles passing through a toll plaza. A toll plaza is a physical structure located on a road or highway that collects tolls from vehicles passing through it.

Toll plazas are typically owned and operated by a government entity or a private company contracted by the government. In some cases, the toll plaza may be owned and operated by the road or highway owner.

Toll plaza management typically includes the following activities:

- Collecting tolls from vehicles
- Monitoring traffic at the toll plaza
- Managing the toll plaza equipment
- Managing the toll plaza staff
- Ensuring the safety of motorists and pedestrians at the toll plaza

REFERENCES

[1] Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.

[2] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

[3] https://dev.mysql.com/doc/refman/8.0/en/

[4] https://www.geeksforgeeks.org/how-to-insert-records-to-a-table-using-jdbc-connection/

[5] https://www.youtube.com/watch?v=CdNno3WuDkI

[6] https://www.youtube.com/watch?v=e7Izp5l6VCg&list=WL&index=13

Vision

To become a premier institute transforming our students to be global professionals.

Mission

M1: Develop competent Human Resources, and create state-of-the-art infrastructure to impart quality education and to support research.

M2: Adopt tertiary approach in teaching – learning pedagogy that transforms students to become professionally competent technocrats and entrepreneurs.

M3: Nurture and train students to develop the qualities of global professionals.

Department of Computer Science and Engineering

Vision

To impart quality education in the field of Computer Science and Engineering with emphasis on innovative thinking, communication and leadership skills to meet the global challenges in IT paradigm.

Mission

M1: Focus on student centric approach through experiential learning and necessary infrastructure.

M2: Develop innovative thinking, communication and leadership skills by creating conducive environment and relevant training.

M3: Enrich students by developing the traits of global professionals.

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K. R. PURAM, BENGALURU - 560036