VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

A PROJECT REPORT
ON

BUS PASS MANAGEMENT SYSTEM

BY

RANITH N
4SF20CS100

SHREYITH J AMIN
4SF20CS139

In the partial fulfillment of the requirement for V Sem. B. E. (CSE)

DBMS LABORATORY WITH MINI PROJECT

Under the guidance of

Ms. Vidya V V
Asst. Professor, Dept. of CS&E

Department of Computer Science & Engineering
SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
Adyar, Mangaluru-575007
2022-23

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT

An Autonomous Institution
Adyar, Mangaluru – 575007

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**BUS PASS MANAGEMENT SYSTEM"** is submitted in partial fulfillment for the requirement of V sem. B. E. (Computer Science & Engineering), "DBMS LABORATORY WITH MINI PROJECT" during the year 2022 – 23 is a result of bonafide work carried out by

RANITH N	4SF20CS100
SHREYITH J AMIN	4SF20CS139

…………………………
Ms. Vidya V V
Asst. Prof. Dept. of CS&E
SCEM, Mangaluru

…………………………
Dr. Nagesh H R
HOD, Dept. of CS&E
SCEM, Mangaluru

Signature of the Examiners

1. ………………………….

2. ………………………….

ABSTRACT

A database management system (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool used to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply away to store up and retrieve database information that is both convenient and efficient. Bus Pass Management System is developed for managing and storing all the data related to Student Buss Pass. The purpose of developing this database is to computerize the traditional way of storing or entering the data. This software manages related problems of the Bus Pass System and avoids problems that might occur when data carried out manually. The purpose of this project is to make a system to carry out various operations that is needed to be performed on the table. The database is designed in such a way that updating of various information become easy and faster access.

ACKNOWLEDGEMENT

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **“Project title”** We have completed it as a part of the V semester DBMS Laboratory with Mini Project (18CSL58) of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi.

We are profoundly indebted to our guides, **Ms.Vidya V V,** Assistant Professor, Department of Computer Science & Engineering for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Nagesh H R**, Head & Professor, Department of Computer Science & Engineering for her invaluable support and guidance.

We sincerely thank **Dr. Rajesha. S**, Principal, Sahyadri College of Engineering & Management and **Dr. D. L. Prabhakara**, Director, Sahyadri Educational Institutions, who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

RANITH N
4SF20CS100
V Sem, B.E., CSE
SCEM,Mangaluru

SHREYITH J AMIN
4SF20CS139
V Sem, B.E., CSE
SCEM,Mangaluru

PAGE INDEX

CHAPTER 1

1. INTRODUCTION

1.1 INTRODUCTION TO DBMS

DBMS Stands for "Database Management System." In short, a DBMS is a database program. Technically speaking, it is a software system that uses a standard method of cataloging, retrieving, and running queries on data. The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data (storage and hardware).

As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won't have to modify programs just because changes have been made to the database. With relational DBMSs (RDBMSs), this API is SQL, a standard programming.

1.2 BACKGROUND OF THE PROJECT

Bus Pass Management System is developed for managing and storing all the data related to Student Buss Pass. The purpose of developing this database is to computerize the traditional way of storing or entering the data. This software manages related problems of the Bus Pass System and avoids problems that might occur when data carried out manually. The purpose of this project is to make a system to carry out various operations that is needed to be performed on the table. The database is designed in such a way that updating of various information become easy and faster access.

1.3 NECESSITY OF PROJECT

Bus Pass Management System used to store all the data of the student who are enrolling for bus pass. It has made work faster and easier for the university, that the staff of the college working in this department can get the details of the student who are enrolling for college bus pass in very efficient manner. Since it is in online mode the problem of loss of data will be solved. Whenever an administrator enters the Bus Pass system it shows the basic information like Drivers Records, Vehicles Records, boarding point records, students who have registered records.

1.4 APPLICATIONS AND ADVANTAGES

The software is used to store the data most securely and in proper arrangement way. This software make work easier and faster. This is even help to store all the relevant data about students boarding point details and it generates a bus pass card for the student automatically. The administrator will manage all the drivers details, vehicle details. Whenever an administrator enters the Bus Pass system it shows the basic information like the student who have registered for bus pass system and the boarding point of the student etc. It minimize the use of hard copies

1.5 IMPLEMENTATION

The Panchayat Tax Management System is implemented using home page module, registration/login module, store module, user access module, with insert, update, delete operations. MySQL which holds the database and java for the front end which displays the provided modules.

1.6 ORACLE

Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is a multi-model database management system produced and marketed by Oracle Corporation. It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW), etc. Larry Ellison and his two friends and former co-workers, Bob Miner and Ed Oates started a consultancy called Software Development and Laboratories (SDL) in 1977. SDL developed the original version of oracle.

1.7 MY SQL

The software is used to store the data most securely and in proper arrangement way. This software make work easier and faster. This is even help to store all the relevant data about students boarding point details and it generates a bus pass card for the student automatically. MySQL server is used for data operations like querying, sorting, filtering, grouping, modifying and joining the tables. Before learning the commonly used queries, let us look into some of the advantages of MySQL.

1.8 JAVA

Java is a popular programming language. It is owned by Oracle, and more devices run Java. Java works on different platforms (Windows, Mac, Linux). It is open-source secure, fast, free and powerful. As Java is close to C++ and C#, it makes it easy for programmers to switch to Java. Java was developed in the mid-1990s by James A. Gosling, a former computer scientist with Sun Microsystems.

Java is an Object-Oriented programming language developed by James Gosling in the early 1990s. The team initiated this project to develop a language for digital devices such as set-top boxes, television, etc. Java is used in all kinds of applications like Mobile Applications (Android is Java-based), desktop applications, web applications, client-server applications, enterprise applications, and many more.

1.9 NETBEANS

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules. The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications. The underlying NetBeans platform supports creation of new applications and further development of existing applications using modular software components. As an application running on the NetBeans Platform, the NetBeans IDE itself is extensible and can be extended to support new

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1Hardware Requirements

- Processor: Any processor above 500 MHz
- RAM: Min 4GB
- Hard Disk: Min 256GB

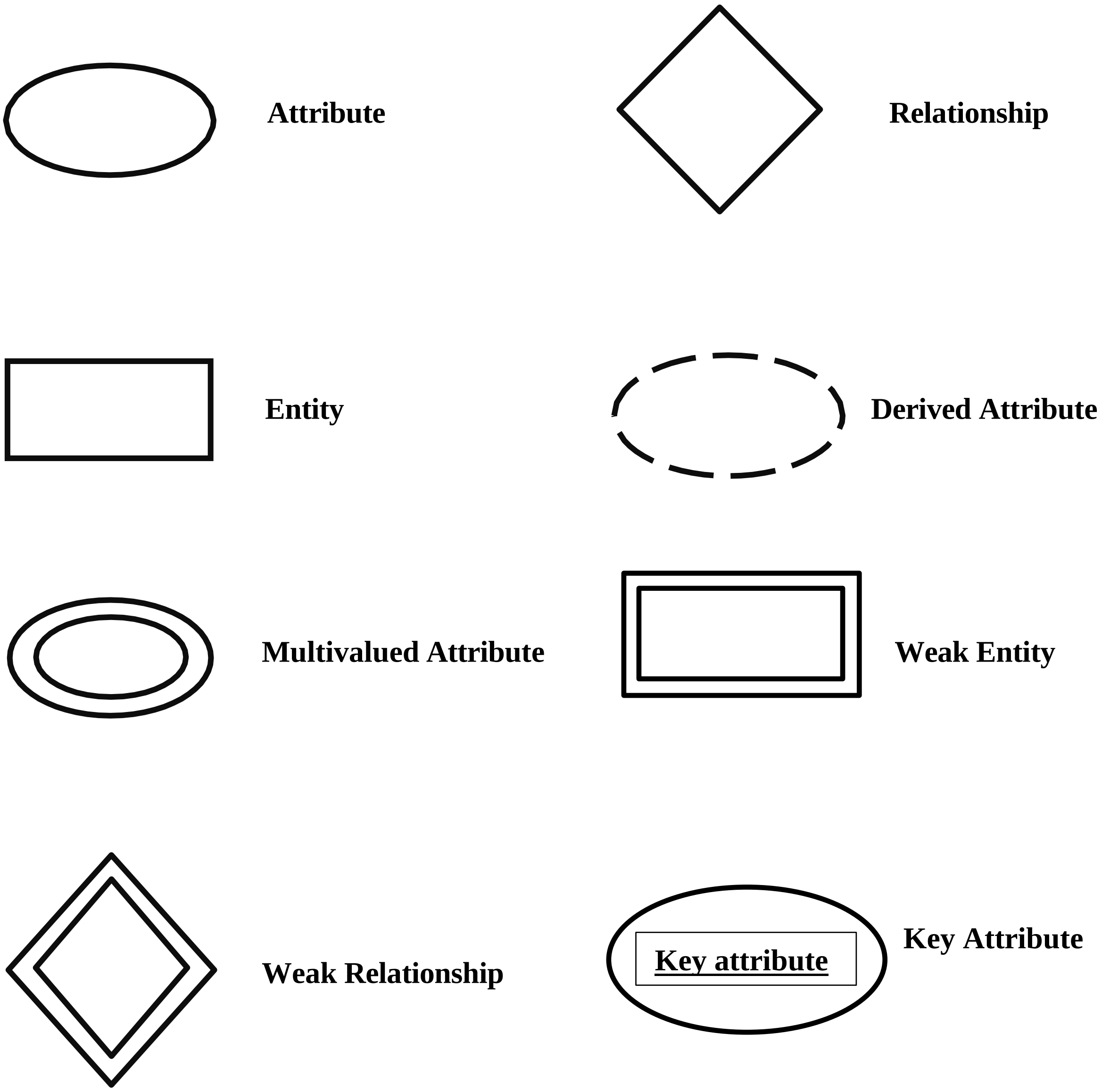
2.2 Software Requirements

- Backend: MYSQL
- Programming Language: Java /Python
- IDE: NetBeans 12.6

CHAPTER 3

3.1 ER-diagram

An entity–relationship model describes interrelated things of interest in a specific domain knowledge. The ER Diagram of our project is shown in the **figure:3.1.1**



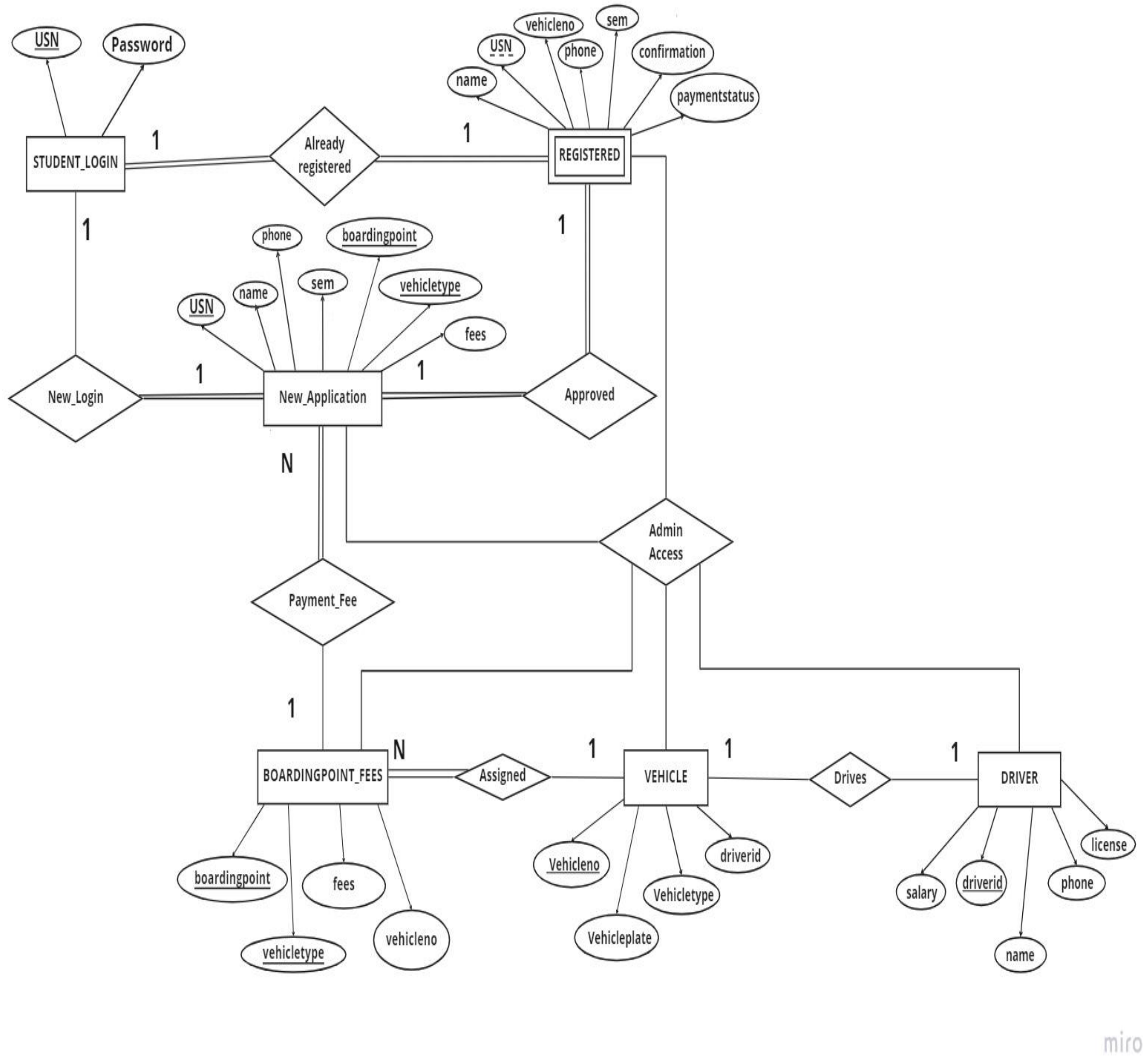


Figure 3.1.1 ER Diagram of Bus Pass Management System

3.2. RELATIONAL SCHEMA

3.2.1 Mapping From ER Diagram to Schema Diagram

1. Mapping of regular entities:-This step involves mapping all the regular entity types to tabular format by identifying their primary keys.
2. Mapping of Weak Entity :-When mapping weak entity types along with other attributes the partial key and primary key of parent entity together will form their primary key of the new relation.
3. Mapping of 1:1 Relation:-In this step foreign keys are assigned using foreign key approach. The primary key of the participating relation R or S is added as primary key to second entity types by looking at the participating constraints.
4. Mapping of 1:N Relation:-Foreign key approach is used to add one sided primary key to the n sided entity at foreign key.
5. Mapping of M:N Relation :-Here we use the cross reference approach where the relationship is converted to a new relation within attributes on primary keys of both participating relation.
6. Mapping of N-ary Relation:-For mapping N ary relationship we create a new relation with a relationship name in its attribute and primary keys of all participating entity types.
7. Mapping of Multivalued Relation :-For multivalued attributes a separate relation has to be created along with primary key of parent relation. A relational schema for a database is an outline of how data is organized.

STEP 1: Mapping of regular entity types.

The regular entity types of our project are shown in figure

STUDENT_LOGIN

USN	Password

NEW_APPLICATION

Name	Phone	Sem

BOARDINGPOINT_FEES

Boardingpoint	Vehicletype	Fees

VEHICLE

Vehicleno	**Vehicletype**	**Vehicleplate**

DRIVER

DriverID	**Name**	**Phone**	**License**	**salary**

Figure 3.2.1

STEP2: Mapping of weak entity types

The weak entity types of our project are shown in figure

USN	**Name**	**Phone**	**Sem**	**Vehicleno**	**Payment**	**Confirmation**

STEP 3: Mapping of binary 1:1 relation types

NEW APPLICATION

USN	**Name**	**Phone**	**Sem**

VEHICLE

Vehicleno	**vehicleplate**	**vehicletype**	**driverID**

Figure 3.2.2 1:1 Mapping

STEP 4: Mapping of 1: N relation types

For every 1:N relation types identify the entity which is in the N-side. Make primary key of entity which is participating in 1 side as foreign key of entity which is N-side. If there are any attributes for the relationship add to the N-side.

NEW APPLICATION

USN	**Name**	**Phone**	**Sem**	**Boardingpoint**	**Vehicletype**	**Fees**

BOARDINGPOINT_FEES

Boardingpoint	**Vehicletype**	**fees**	**Vehicleno**

Figure 3.2.3 1:N Mapping

STEP 5: Mapping of M: N relation types

The ERD of our project doesn’t contain M:N relationship type

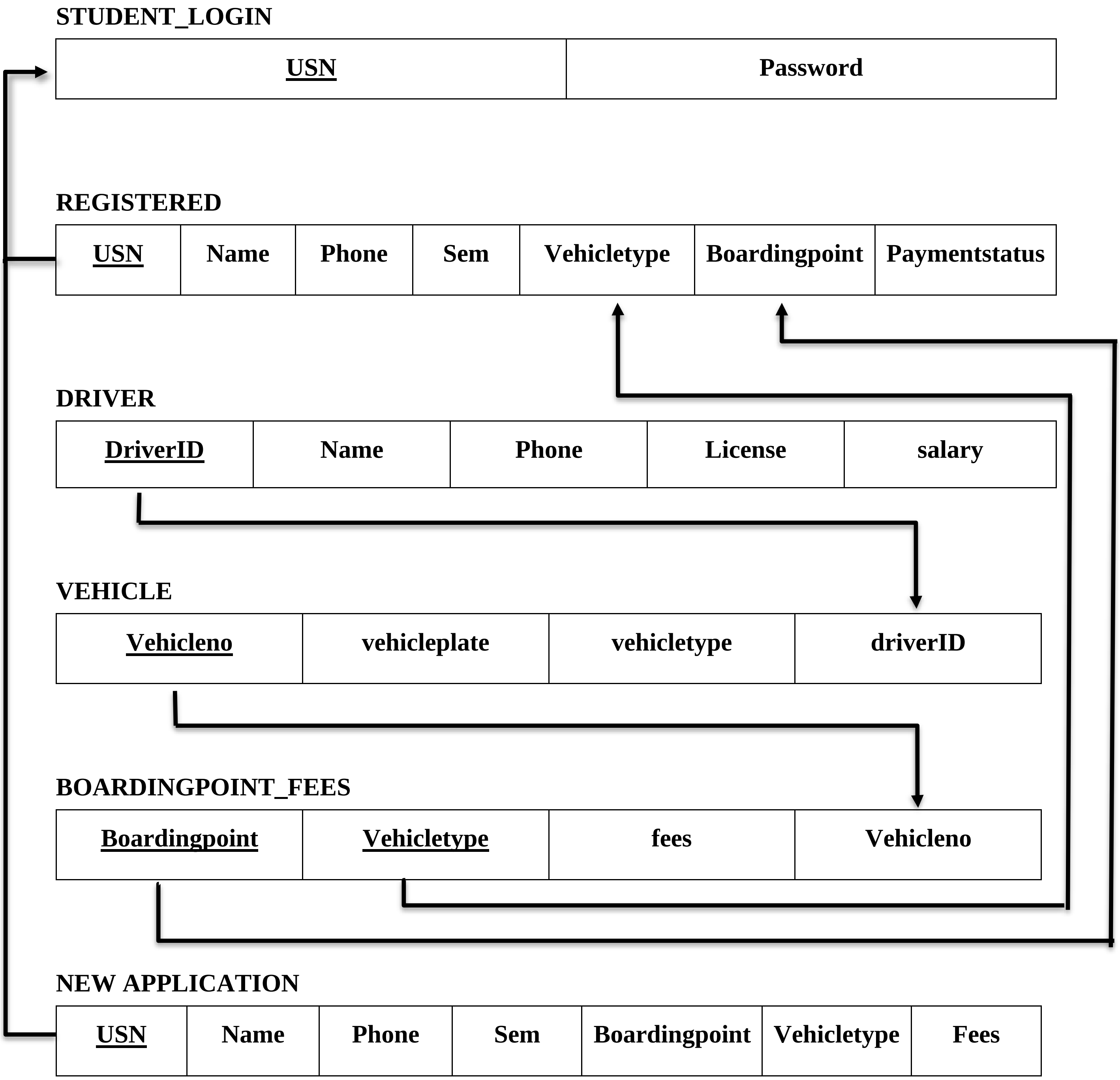
STEP 6: Mapping of multivalued attributes

The ERD of our project doesn’t contain Multivalued Attribute

STEP 7: Mapping of n-ary relation types

The ERD of our project does not contain any n-ary relations

3.3 SCHEMA DIAGRAM



STUDENT_LOGIN

USN	Password

REGISTERED

USN	Name	Phone	Sem	Vehicletype	Boardingpoint	Paymentstatus

DRIVER

DriverID	Name	Phone	License	salary

VEHICLE

Vehicleno	vehicleplate	vehicletype	driverID

BOARDINGPOINT_FEES

Boardingpoint	Vehicletype	fees	Vehicleno

NEW APPLICATION

USN	Name	Phone	Sem	Boardingpoint	Vehicletype	Fees

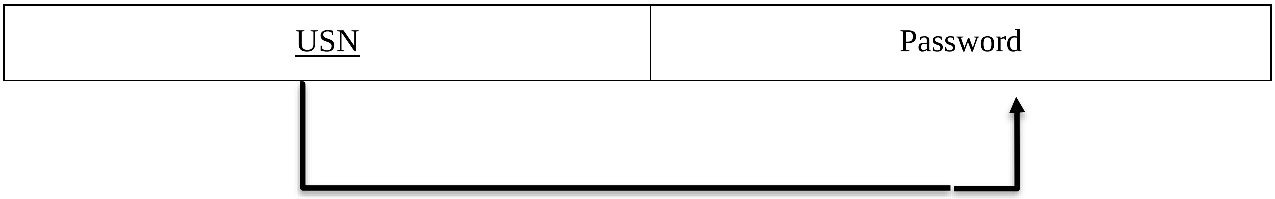
Figure 3.3.1 Schema Diagram

CHAPTER 4

NORMALIZATION

1NF:

STUDENT_LOGIN

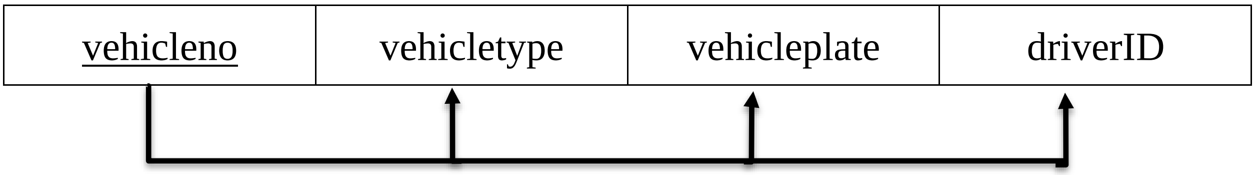


USN	Password

FD1: USN-> {PASSWORD }

1NF: It is 1NF because there is no multivalued attributes in the relational schema

VEHICLE

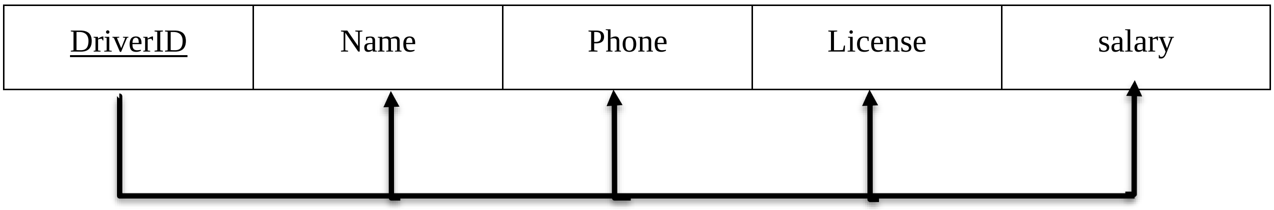


vehicleno	vehicletype	vehicleplate	driverID

VEHICLENO-> {VEHICLETYPE,VEHICLEPLATE,DRIVERID}

1NF: It is 1NF because there is no multivalued attributes in the relational schema

DRIVER



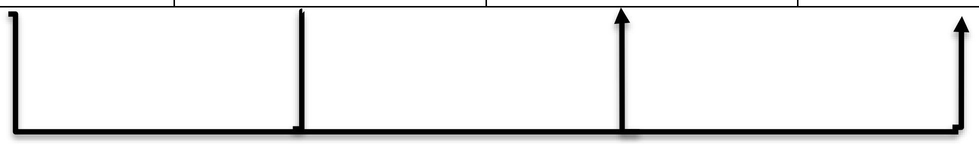
DriverID	Name	Phone	License	salary

DRIVERID-> {NAME,PHONE,LICENSE,SALARY }

1NF: It is 1NF because there is no multivalued attributes in the relational schema

BOARDINGPOINT_FEES

Boardingpoint	Vehicletype	fees	Vehicleno



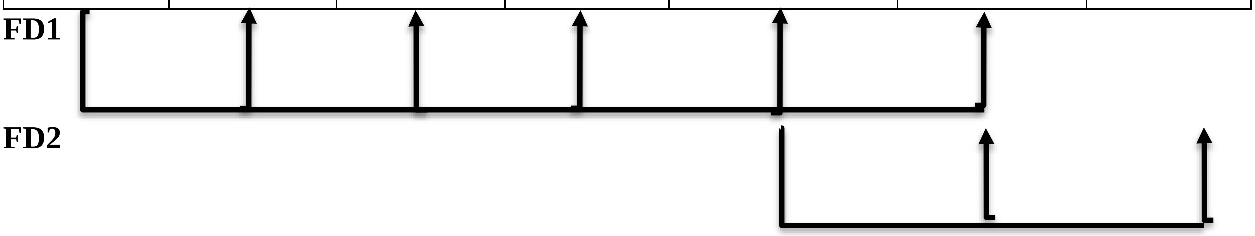
BOARDINGPOINT_FEES,VEHICLETYPE-> {FEES,VEHICLENO}

1NF: It is 1NF because there is no multivalued attributes in the relational schema

2NF:

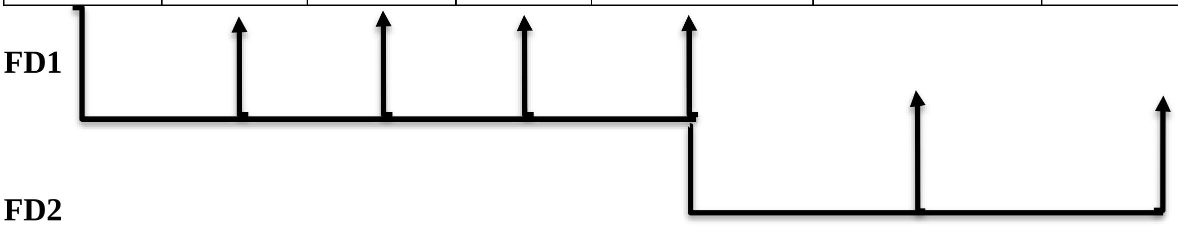
NEW APPLICATION

USN	Name	Phone	Sem	Boardingpoint	Vehicletype	Fees



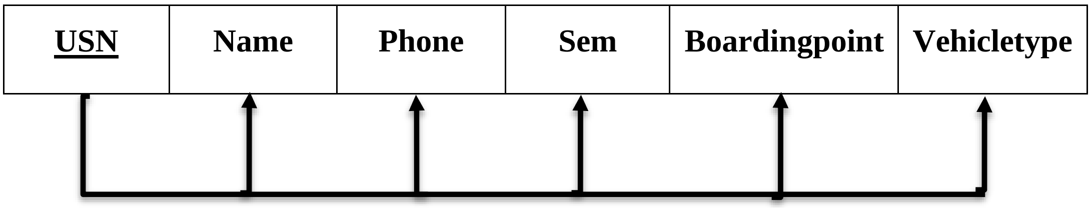
REGISTERED

USN	Name	Phone	Sem	Vehicletype	Boardingpoint	Paymentstatus



3NF:

NEW APPLICATION

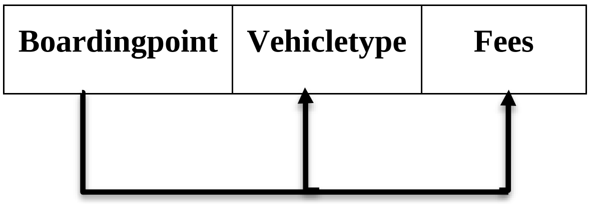


USN	Name	Phone	Sem	Boardingpoint	Vehicletype

USN-> {NAME,PHONE,SEM,BOARDINGPOINT,VEHICLETYPE}

REGISTERED

FD1

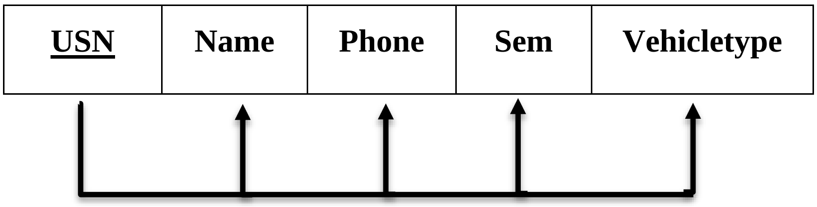


Boardingpoint	Vehicletype	Fees

BOARDINGPOINT -> {FEES,VEHICLETYPE}

REGISTERED

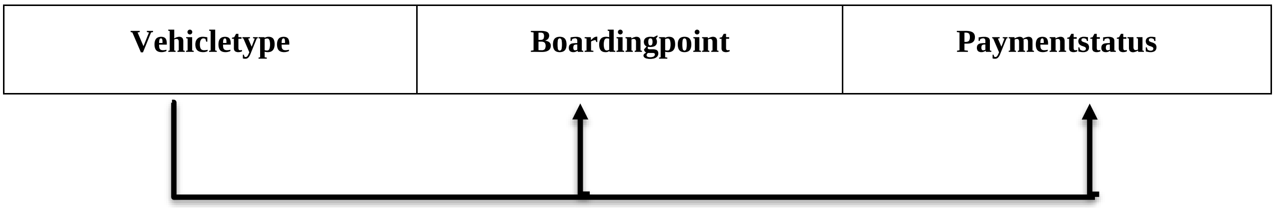
FD1



USN	Name	Phone	Sem	Vehicletype

USN-> {NAME,PHONE,SEM,VEHICLETYPE}

FD2



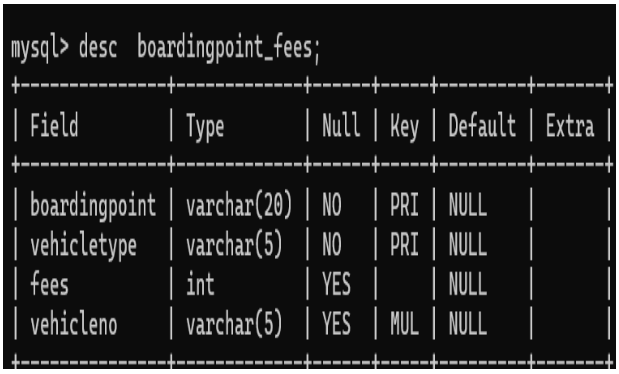
Vehicletype	Boardingpoint	Paymentstatus

VEHICLETYPE,BOARDINGPOINT-> {PAYMENTSTATUS}

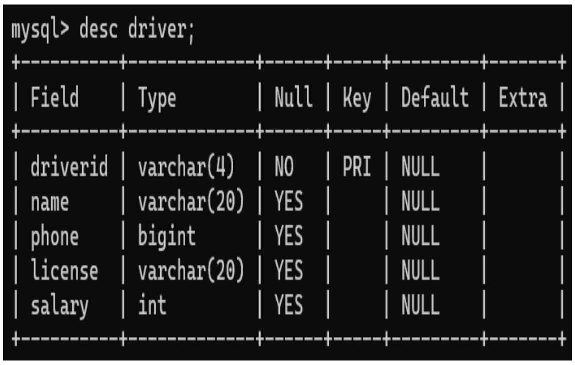
CHAPTER 5

Implementation

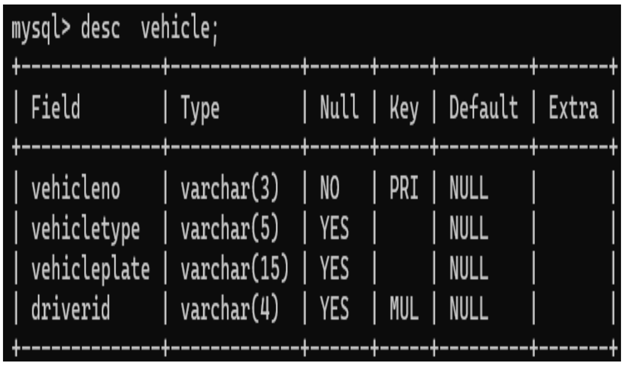
5.1 Create Table <table name> <description>



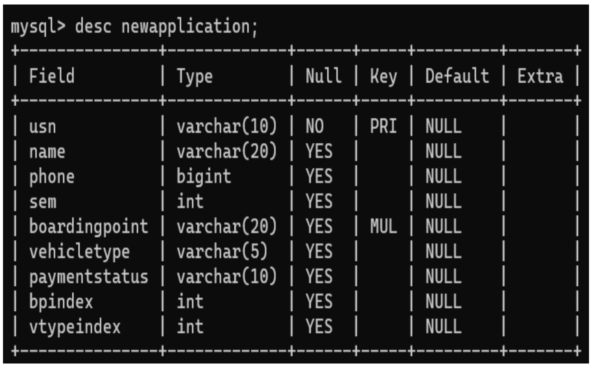
```
mysql> desc  boardingpoint_fees;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| boardingpoint | varchar(20) | NO   | PRI | NULL    |       |
| vehicletype   | varchar(5)  | NO   | PRI | NULL    |       |
| fees          | int         | YES  |     | NULL    |       |
| vehicleno     | varchar(5)  | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```



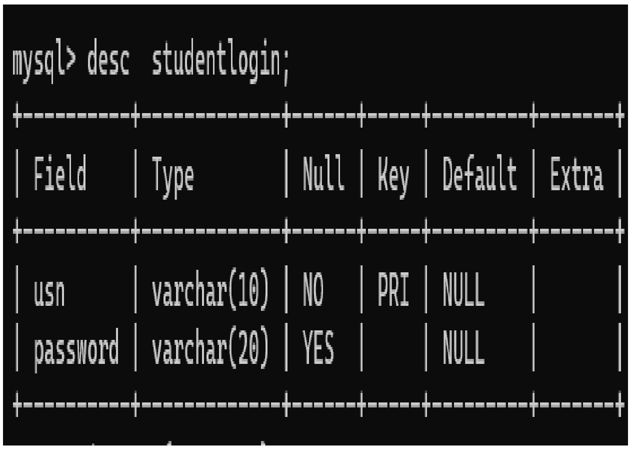
```
mysql> desc driver;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| driverid | varchar(4)  | NO   | PRI | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| phone    | bigint      | YES  |     | NULL    |       |
| license  | varchar(20) | YES  |     | NULL    |       |
| salary   | int         | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```



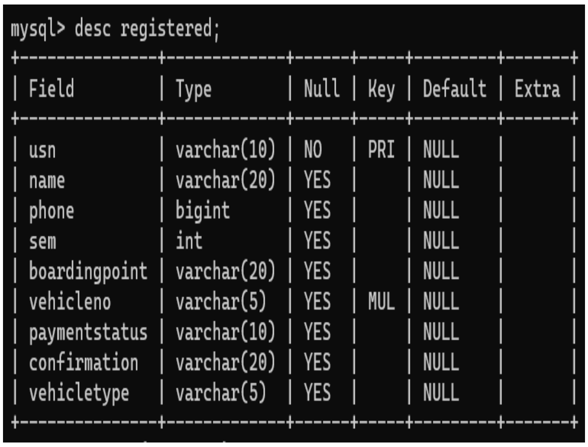
```
mysql> desc  vehicle;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| vehicleno    | varchar(3)  | NO   | PRI | NULL    |       |
| vehicletype  | varchar(5)  | YES  |     | NULL    |       |
| vehicleplate | varchar(15) | YES  |     | NULL    |       |
| driverid     | varchar(4)  | YES  | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```



```
mysql> desc newapplication;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| usn           | varchar(10) | NO   | PRI | NULL    |       |
| name          | varchar(20) | YES  |     | NULL    |       |
| phone         | bigint      | YES  |     | NULL    |       |
| sem           | int         | YES  |     | NULL    |       |
| boardingpoint | varchar(20) | YES  | MUL | NULL    |       |
| vehicletype   | varchar(5)  | YES  |     | NULL    |       |
| paymentstatus | varchar(10) | YES  |     | NULL    |       |
| bpindex       | int         | YES  |     | NULL    |       |
| vtypeindex    | int         | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```



```
mysql> desc  studentlogin;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| usn      | varchar(10) | NO   | PRI | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```



```
mysql> desc registered;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| usn           | varchar(10) | NO   | PRI | NULL    |       |
| name          | varchar(20) | YES  |     | NULL    |       |
| phone         | bigint      | YES  |     | NULL    |       |
| sem           | int         | YES  |     | NULL    |       |
| boardingpoint | varchar(20) | YES  |     | NULL    |       |
| vehicleno     | varchar(5)  | YES  | MUL | NULL    |       |
| paymentstatus | varchar(10) | YES  |     | NULL    |       |
| confirmation  | varchar(20) | YES  |     | NULL    |       |
| vehicletype   | varchar(5)  | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

5.2 JDBC DRIVER IS USED TO CONNECT JAVA AND MY SQL

CODE:

```
package checkmysqlconnector;
import java.util.logging.Level;
import java.util.logging.Logger;
public class
CheckMysqlConnector {
public static void main(String[] args) {

 // TODO code application logic here
 String className ="com.mysql.cj.jdbc.Driver";
 try {
 Class.forName(className);
 System.out.println("driver loaded succ");
 }
 catch (ClassNotFoundException ex){
 System.out.println("driver nots loaded succ");
 System.out.println(ex.getMessage());
        }
    }
}
```

5.3 Insert/Update/Delete has been implemented as follows:

FOR INSERT:

```
try {
        string driverid,name,phone,license,salary;
        driverid=driverid.gettext();
        name=name.gettext();
        phone=phone.gettext();
        license=license.gettext();
        salary=salary.gettext();

        pst=con.preparestatement("insert into driver(driverid,name,phone,license,salary)
        values(?,?,?,?,?)");
        pst.setstring(1, driverid);
        pst.setstring(2, name);
        pst.setstring(3, phone);
        pst.setstring(4, license);
        pst.setstring(5, salary);
        if(driverid.isempty() || name.isempty() || phone.isempty() || license.isempty() ||
        salary.isempty())
        {
        joptionpane.showmessagedialog(this, "field is empty");
        }
        else if(phone.length()!=10)
        {
        joptionpane.showmessagedialog(this, "invalid phone number");
        }
        else
        {
        pst.executeupdate();
        joptionpane.showmessagedialog(this, "record saved");
        table_update();

        driverid.settext("");
        name.settext("");
        phone.settext("");
        license.settext("");
        salary.settext("");
        driverid.requestfocus();}}
         catch (sqlexception ex) {
        logger.getlogger(driverrecords.class.getname()).log(level.severe, null, ex);
        joptionpane.showmessagedialog(this, ex);
        }
}
```

FOR DELETE:

```
try {
            String driverid,name,phone,license;
            driverid=DriverID.getText();

            pst=con.prepareStatement("delete from driver where driverid=?");
            pst.setString(1, driverid);


            if(!driverid.isEmpty())
            {
                pst.executeUpdate();
                JOptionPane.showMessageDialog(this, "Record deleted");
                table_update();
            }
            else
            {
                JOptionPane.showMessageDialog(this, "Select row to Delete");
            }



            DriverID.setText("");
            Name.setText("");
            Phone.setText("");
            License.setText("");
             Salary.setText("");

            DriverID.requestFocus();

        } catch (SQLException ex) {
            Logger.getLogger(DriverRecords.class.getName()).log(Level.SEVERE, null, ex);
        }
}
```

FOR UPDATE:

```
try {
            DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
            int selectedIndex = jTable1.getSelectedRow();

        String driverid,name,phone,license,salary;
        driverid=DriverID.getText();
        name=Name.getText();
        phone=Phone.getText();
        license=License.getText();
        salary=Salary.getText();

        pst=con.prepareStatement("update driver set name=?,phone=?,license=?,salary=? Where
        driverid=?");

        pst.setString(1, name);
        pst.setString(2, phone);
        pst.setString(3, license);
        pst.setString(4, salary);
        pst.setString(5, driverid);

        if(!driverid.isEmpty())
        {
            pst.executeUpdate();
            JOptionPane.showMessageDialog(this, "Record updated");
            table_update();
        }
        else
        {
            JOptionPane.showMessageDialog(this, "Select row to Update");
        }

        DriverID.setText("");
        Name.setText("");
        Phone.setText("");
        License.setText("");
        Salary.setText("");
        DriverID.requestFocus();

    } catch (SQLException ex) {
        Logger.getLogger(DriverRecords.class.getName()).log(Level.SEVERE, null, ex);
    }
    // TODO add your handling code here:
 }
```

5.4 Triggers

Trigger:

We have used the below trigger to convert the owner name in all the registration page from lower case to upper case.

```
CREATE DEFINER=`root`@`localhost`

TRIGGER `bd_reg_BEFORE_INSERT` BEFORE INSERT ON

`bd_reg` FOR EACH ROW BEGIN

set new.owner_name=upper(NEW.owner_name);

END
```

CHAPTER 6

RESULTS AND DISSCUSSION

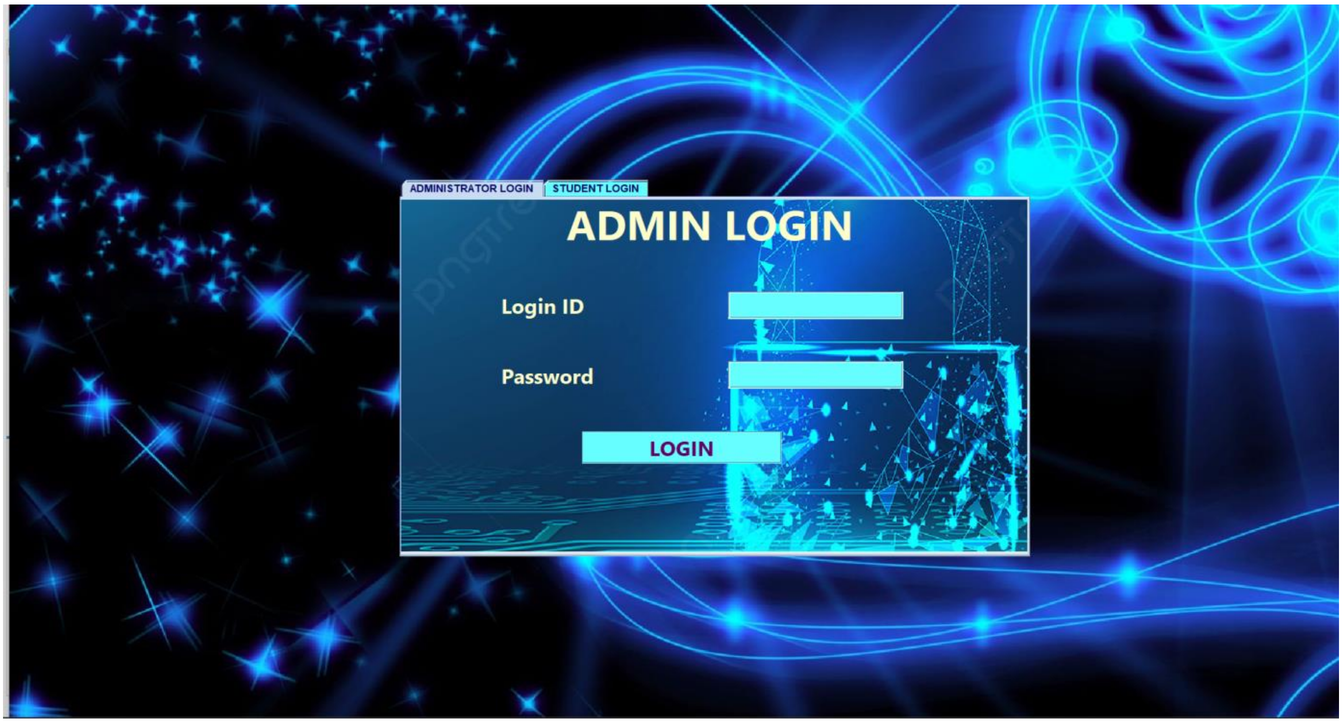


Figure 6.1 Admin Login

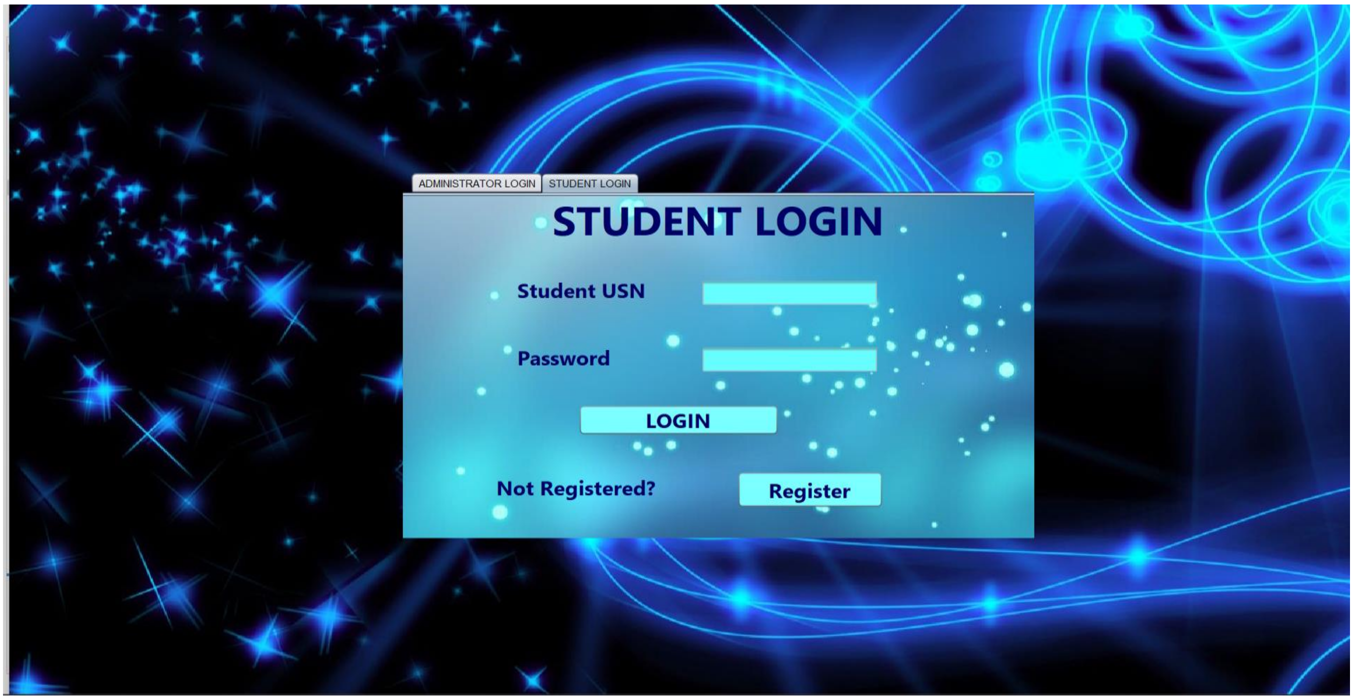


Figure 6.2 Student Login

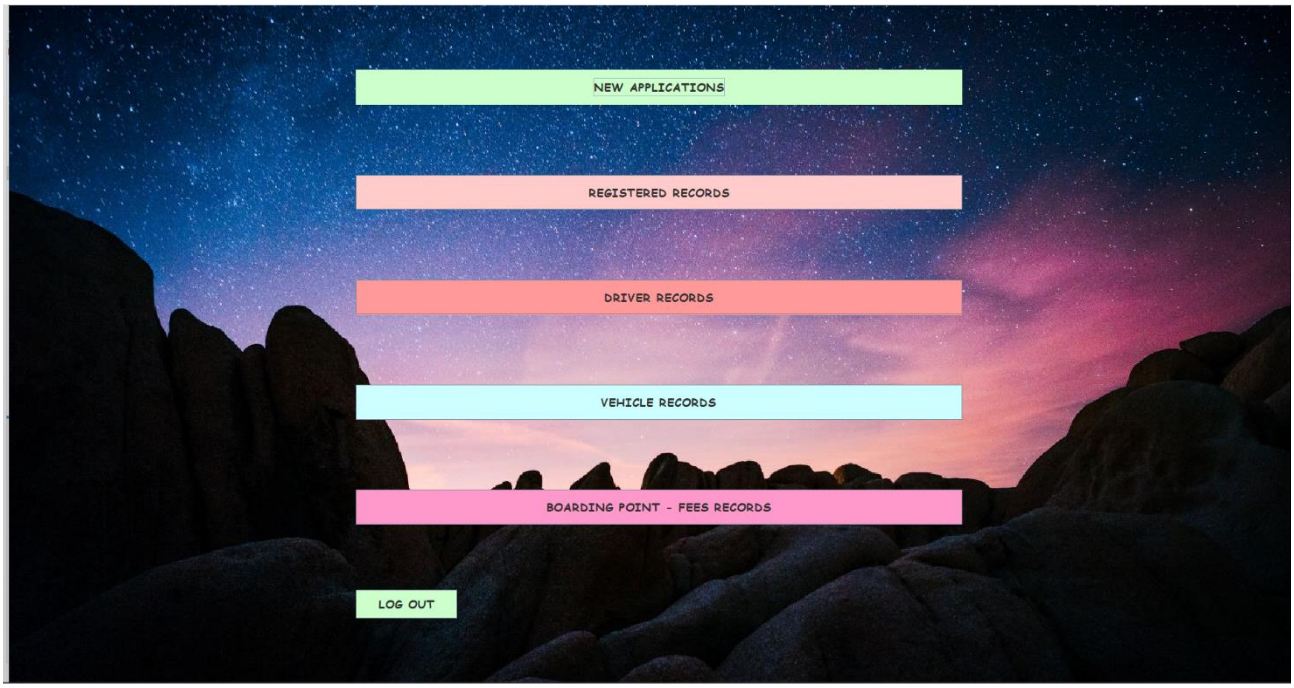
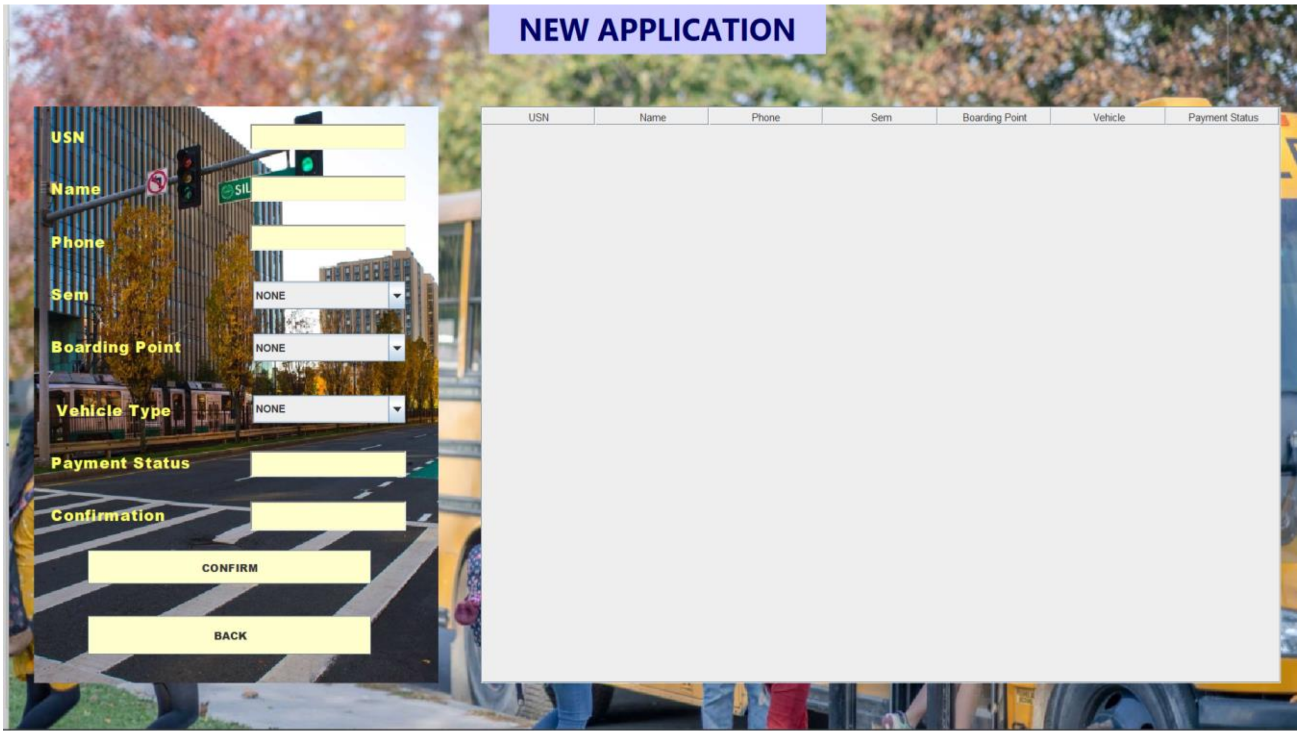


Figure 6.3 Admin Page



USN

Name

Phone

Sem NONE

Boarding Point NONE

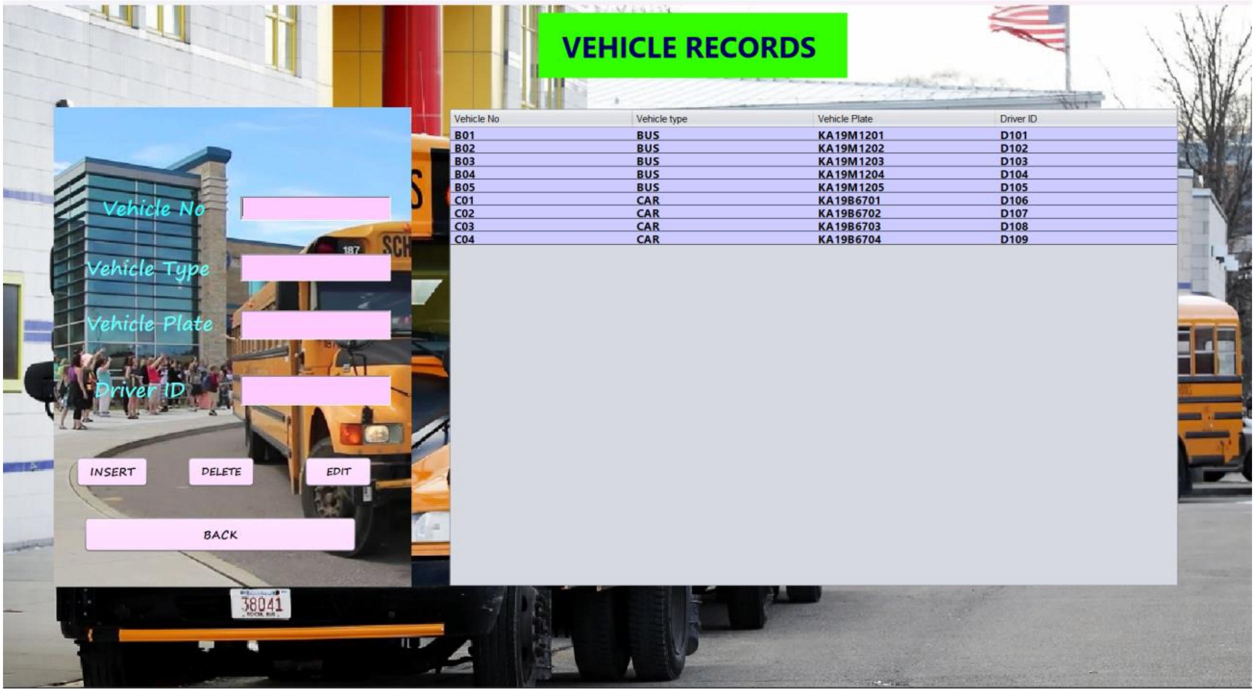
Vehicle Type NONE

Payment Status

Confirmation

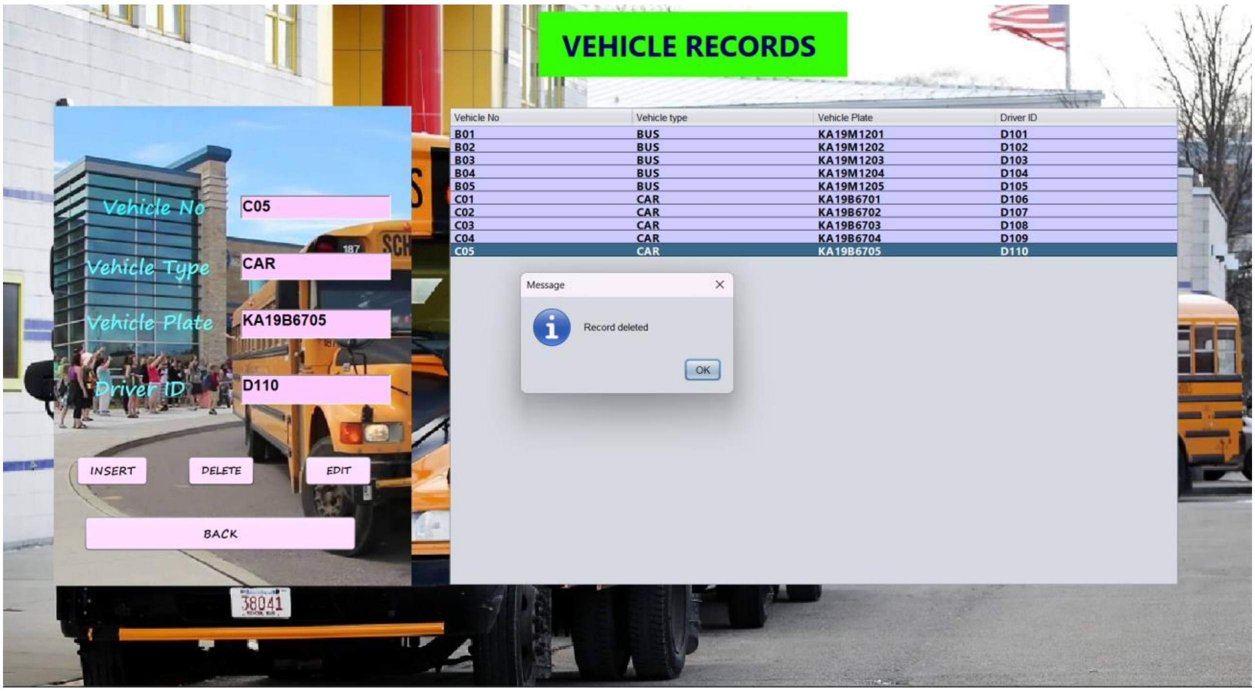
USN	Name	Phone	Sem	Boarding Point	Vehicle	Payment Status

Figure 6.4 New Application page



Vehicle No	Vehicle type	Vehicle Plate	Driver ID
B01	BUS	KA19M1201	D101
B02	BUS	KA19M1202	D102
B03	BUS	KA19M1203	D103
B04	BUS	KA19M1204	D104
B05	BUS	KA19M1205	D105
C01	CAR	KA19B6701	D106
C02	CAR	KA19B6702	D107
C03	CAR	KA19B6703	D108
C04	CAR	KA19B6704	D109

Figure 6.5 Vehicle Record



Vehicle No C05

Vehicle Type CAR

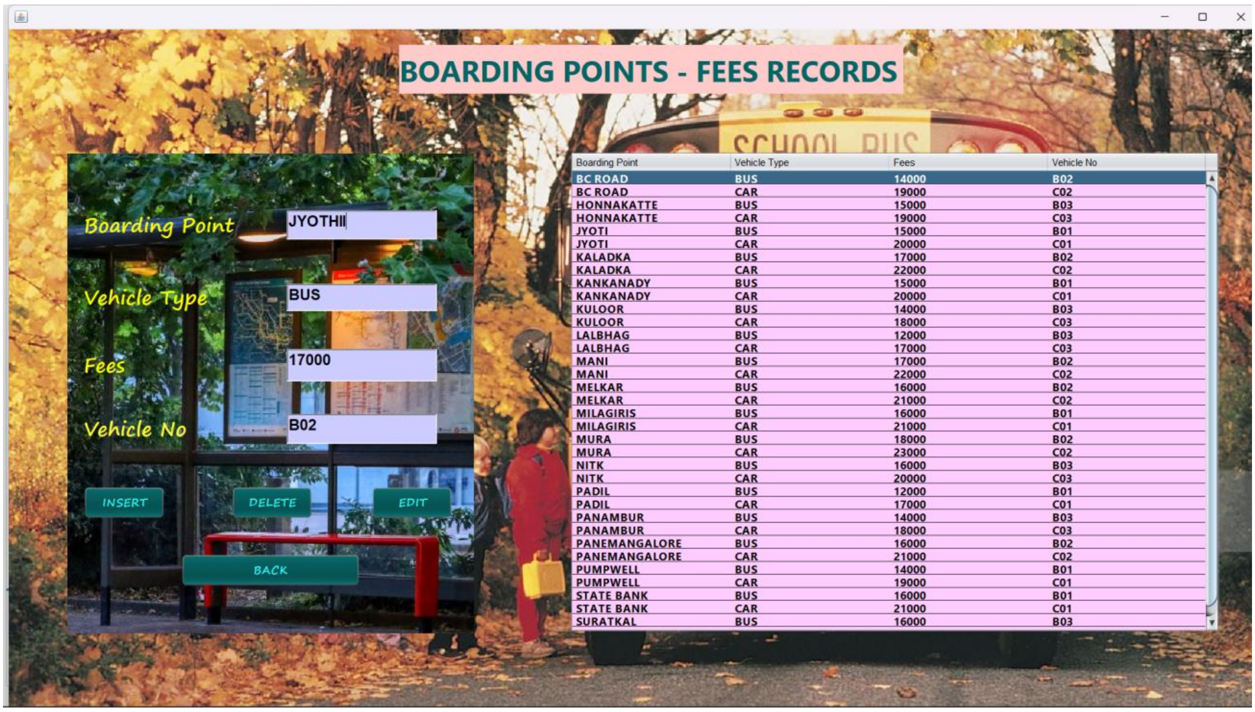
Vehicle Plate KA19B6705

Driver ID D110

Vehicle No	Vehicle type	Vehicle Plate	Driver ID
B01	BUS	KA19M1201	D101
B02	BUS	KA19M1202	D102
B03	BUS	KA19M1203	D103
B04	BUS	KA19M1204	D104
B05	BUS	KA19M1205	D105
C01	CAR	KA19B6701	D106
C02	CAR	KA19B6702	D107
C03	CAR	KA19B6703	D108
C04	CAR	KA19B6704	D109
C05	CAR	KA19B6705	D110

Record deleted

Figure 6.6 Delete Record



BOARDING POINTS - FEES RECORDS

Boarding Point: JYOTHII

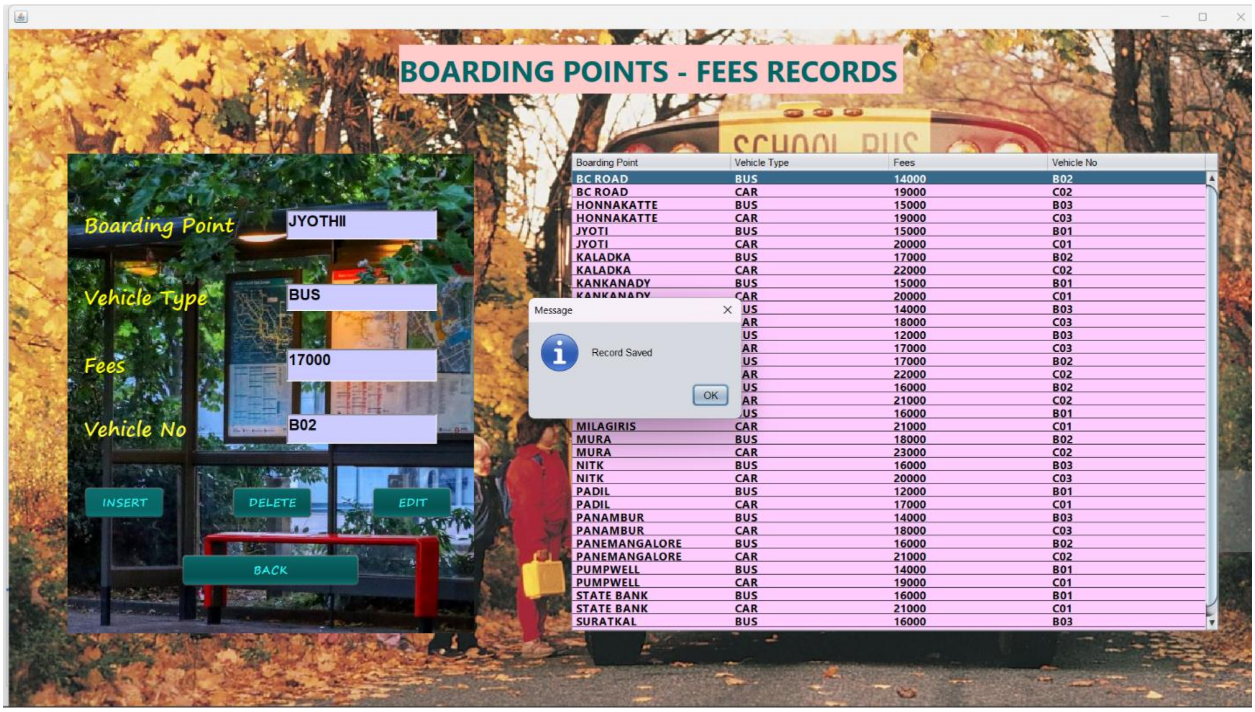
Vehicle Type: BUS

Fees: 17000

Vehicle No: B02

Boarding Point	Vehicle Type	Fees	Vehicle No
BC ROAD	BUS	14000	B02
BC ROAD	CAR	19000	C02
HONNAKATTE	BUS	15000	B03
HONNAKATTE	CAR	19000	C03
JYOTI	BUS	15000	B01
JYOTI	CAR	20000	C01
KALADKA	BUS	17000	B02
KALADKA	CAR	22000	C02
KANKANADY	BUS	15000	B01
KANKANADY	CAR	20000	C01
KULOOR	BUS	14000	B03
KULOOR	CAR	18000	C03
LALBHAG	BUS	12000	B03
LALBHAG	CAR	17000	C03
MANI	BUS	17000	B02
MANI	CAR	22000	C02
MELKAR	BUS	16000	B02
MELKAR	CAR	21000	C02
MILAGIRIS	BUS	16000	B01
MILAGIRIS	CAR	21000	C01
MURA	BUS	18000	B02
MURA	CAR	23000	C02
NITK	BUS	16000	B03
NITK	CAR	20000	C03
PADIL	BUS	12000	B01
PADIL	CAR	17000	C01
PANAMBUR	BUS	14000	B03
PANAMBUR	CAR	18000	C03
PANEMANGALORE	BUS	16000	B02
PANEMANGALORE	CAR	21000	C02
PUMPWELL	BUS	14000	B01
PUMPWELL	CAR	19000	C01
STATE BANK	BUS	16000	B01
STATE BANK	CAR	21000	C01
SURATKAL	BUS	16000	B03

Figure 6.7 Delete Record



BOARDING POINTS - FEES RECORDS

Boarding Point: JYOTHII

Vehicle Type: BUS

Fees: 17000

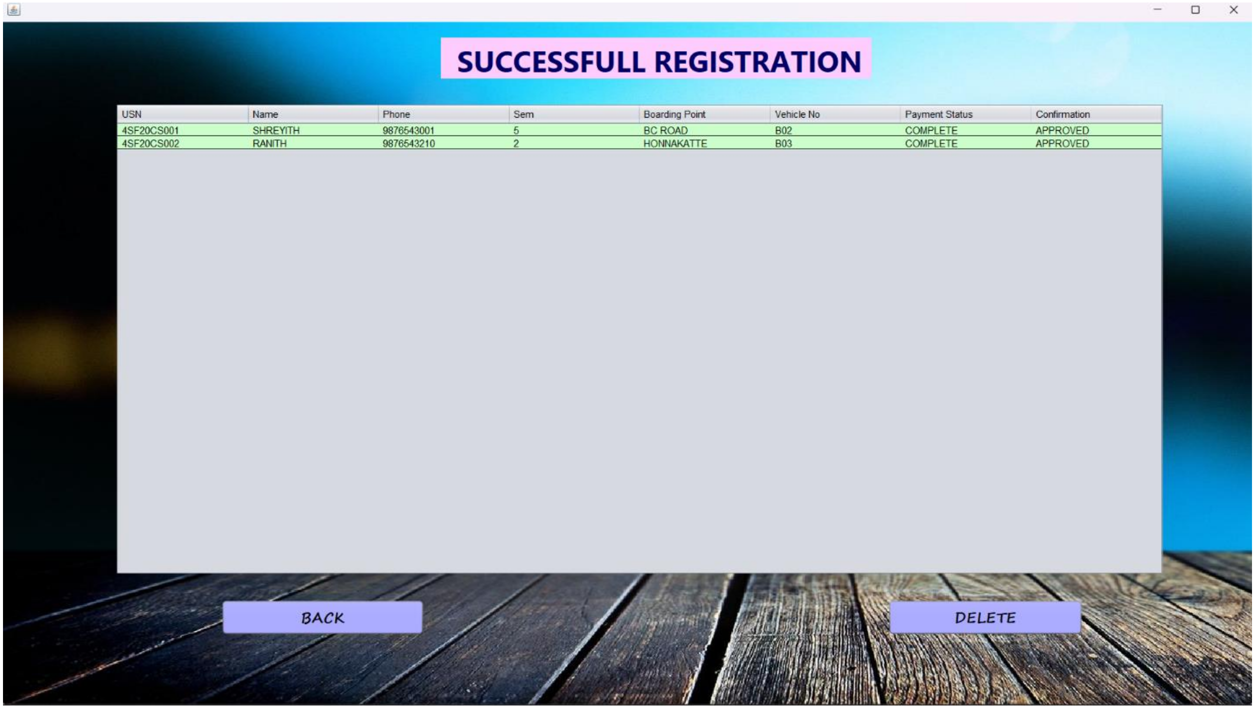
Vehicle No: B02

Message

Record Saved

Boarding Point	Vehicle Type	Fees	Vehicle No
BC ROAD	BUS	14000	B02
BC ROAD	CAR	19000	C02
HONNAKATTE	BUS	15000	B03
HONNAKATTE	CAR	19000	C03
JYOTI	BUS	15000	B01
JYOTI	CAR	20000	C01
KALADKA	BUS	17000	B02
KALADKA	CAR	22000	C02
KANKANADY	BUS	15000	B01
KANKANADY	CAR	20000	C01
	US	14000	B03
	AR	18000	C03
	US	12000	B03
	AR	17000	C03
	US	17000	B02
	AR	22000	C02
	US	16000	B02
	AR	21000	C02
	US	16000	B01
MILAGIRIS	CAR	21000	C01
MURA	BUS	18000	B02
MURA	CAR	23000	C02
NITK	BUS	16000	B03
NITK	CAR	20000	C03
PADIL	BUS	12000	B01
PADIL	CAR	17000	C01
PANAMBUR	BUS	14000	B03
PANAMBUR	CAR	18000	C03
PANEMANGALORE	BUS	16000	B02
PANEMANGALORE	CAR	21000	C02
PUMPWELL	BUS	14000	B01
PUMPWELL	CAR	19000	C01
STATE BANK	BUS	16000	B01
STATE BANK	CAR	21000	C01
SURATKAL	BUS	16000	B03

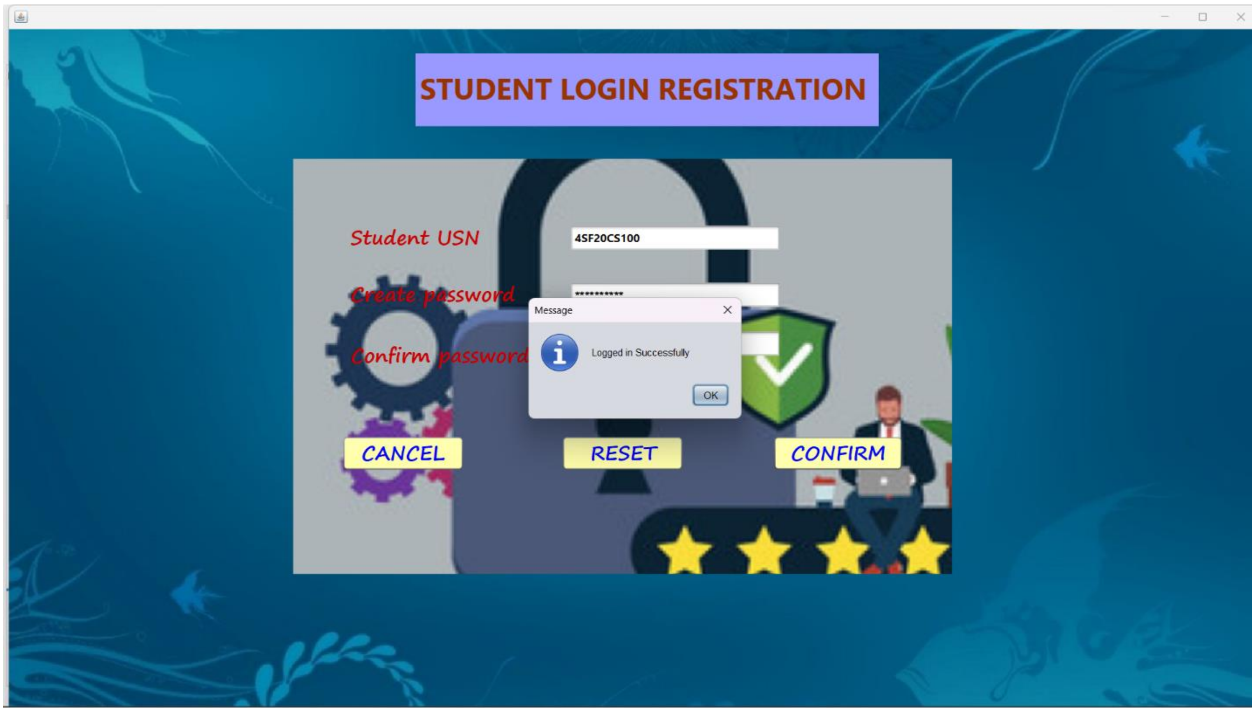
Figure 6.8 Inserting Record



SUCCESSFULL REGISTRATION

USN	Name	Phone	Sem	Boarding Point	Vehicle No	Payment Status	Confirmation
4SF20CS001	SHREYITH	9876543001	5	BC ROAD	B02	COMPLETE	APPROVED
4SF20CS002	RANITH	9876543210	2	HONNAKATTE	B03	COMPLETE	APPROVED

Figure 6.9 successful Registration



STUDENT LOGIN REGISTRATION

Student USN: 4SF20CS100

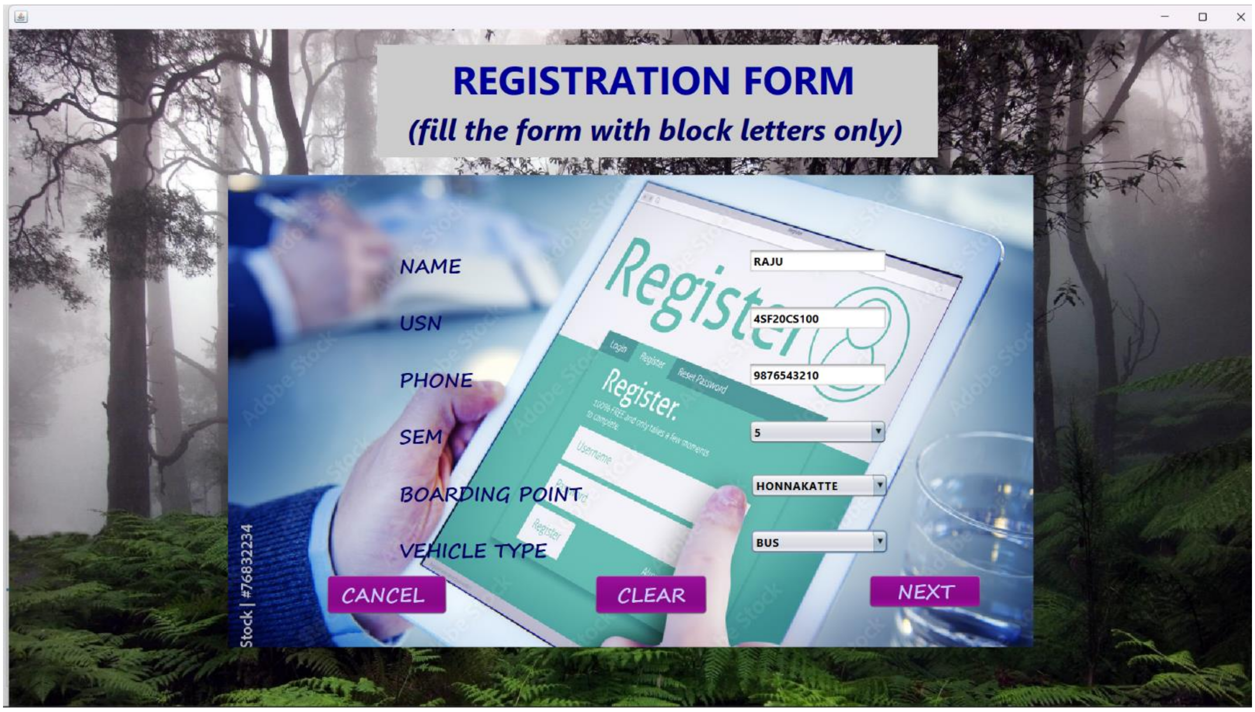
Create password: **********

Confirm password

Message

Logged in Successfully

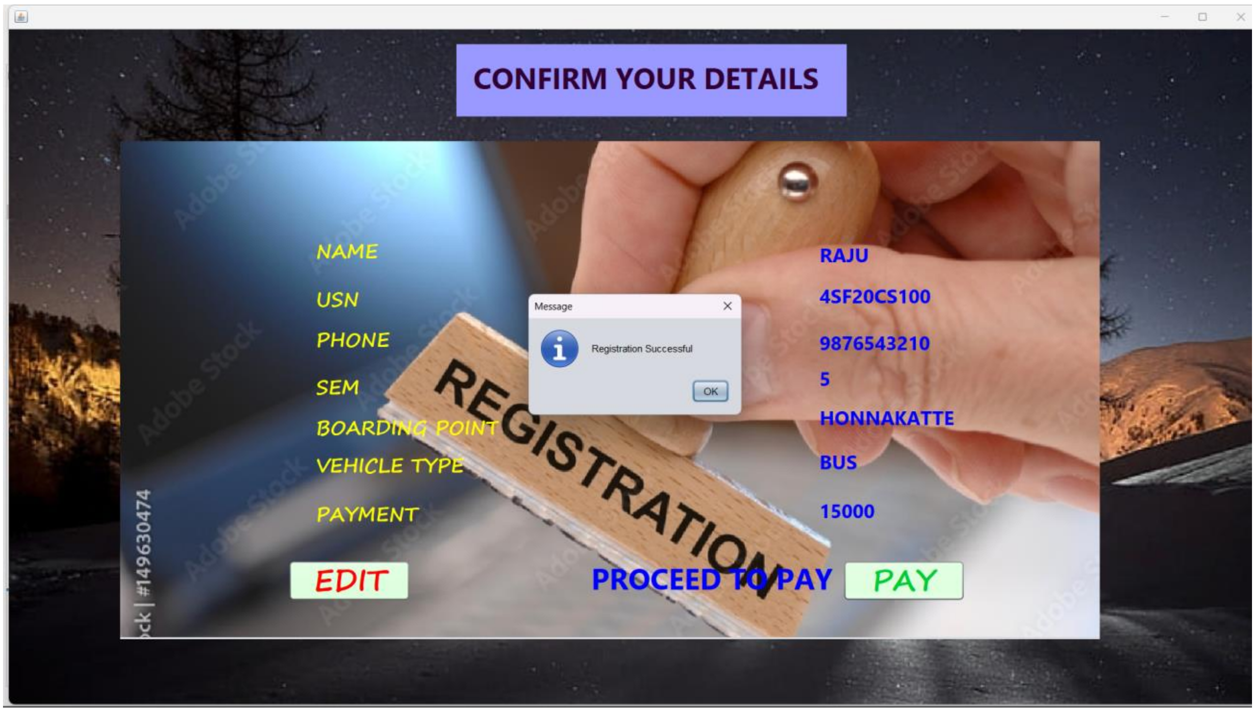
Figure 6.10 Student Login Registration



REGISTRATION FORM
(fill the form with block letters only)

NAME: RAJU
USN: 4SF20CS100
PHONE: 9876543210
SEM: 5
BOARDING POINT: HONNAKATTE
VEHICLE TYPE: BUS

Figure 6.11 Registration Form

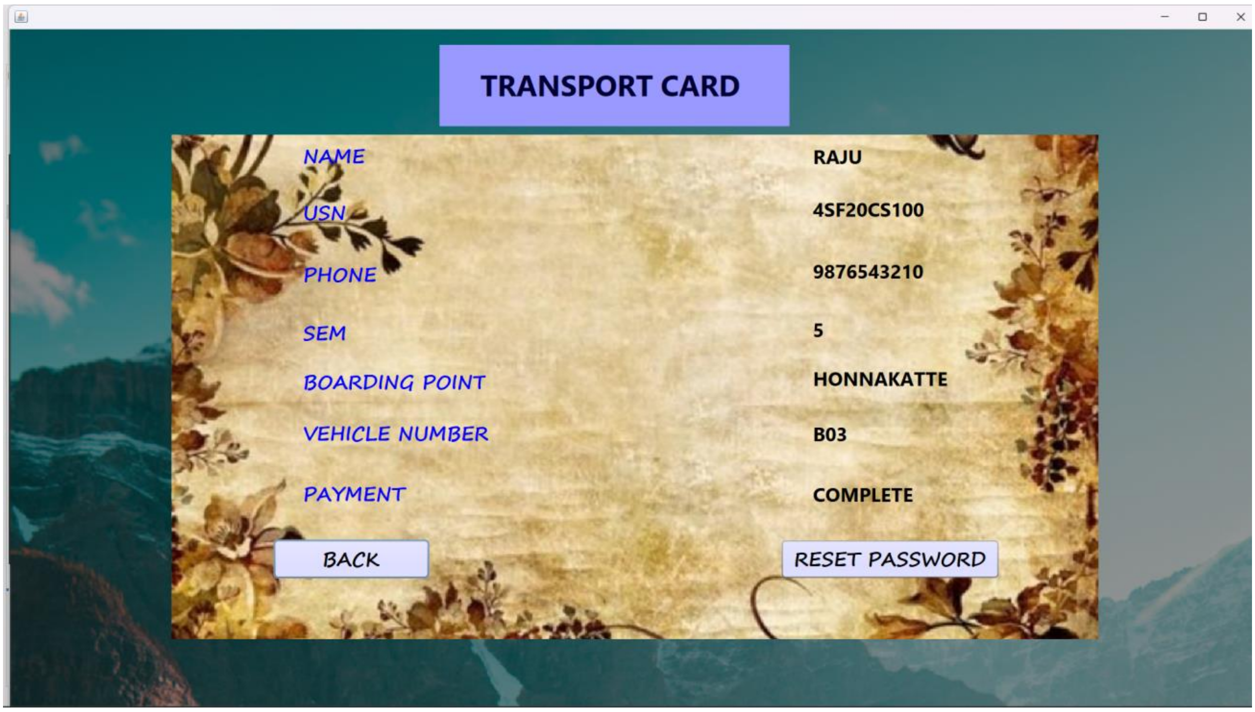


CONFIRM YOUR DETAILS

NAME: RAJU
USN: 4SF20CS100
PHONE: 9876543210
SEM: 5
BOARDING POINT: HONNAKATTE
VEHICLE TYPE: BUS
PAYMENT: 15000

Message
Registration Successful

Figure 6.12 Confirmation Form



TRANSPORT CARD

NAME RAJU

USN 4SF20CS100

PHONE 9876543210

SEM 5

BOARDING POINT HONNAKATTE

VEHICLE NUMBER B03

PAYMENT COMPLETE

Figure 6.13 Transport Card

CHAPTER 7

CONCLUSION

The administrator will manage all the student details. Whenever an administrator enters the bus Pass system it shows the basic information i.e. , Vehicle details, Drivers details, Details of the student who have enrolled to the bus pass system and the the admin will check the details for confirmation. The admin can update and delete operation to all the tables. The total control is under the administrator and the student gets the confirmation if the details have been approved.

Its application: To minimize the use of hard copies. Learnings From project are Applying and understanding the methods of using JDBC, MySQL and JAVA using NetBeans

REFERENCES

[1]. MYSQL:

https://dev.mysql.com/doc/refman/8.0/en/

[2]. JAVA:

https://www.w3schools.com/java/default.asp

[3]. JDBC :

https://www.geeksforgeeks.org/introduction-to-jdbc/

[4]. NETBEANS:

https://netbeans.apache.org/kb/docs/java-se.html

[5]. YOUTUBE:

https://www.youtube.com/watch?v=Ir7FEZRL0hI