VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI – 590 018

A DBMS Mini-Project Report
on

"NAMMA RTO"

Submitted in partial fulfillment of the requirements for the 5th semester of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi

Submitted by:

ARJUNA REDDY G	**1RN17CS017**
LIKITH CHANDAN M	**1RN17CS047**

Under the Guidance of:

Dr. Bhavanishankar K	**Dr. Shashidhara H R**	**Mrs. Mamatha Jajur**
Assistant Professor	**Associate Professor**	**Assistant Professor**
Dept. of CSE	**Dept. of CSE**	**Dept. of CSE**

ESTD:2001
An Institute with a Difference

Department of Computer Science and Engineering
RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

2019-2020

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road,

Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ESTD:2001

An Institute with a Difference

CERTIFICATE

Certified that the DBMS mini-project work entitled **"NAMMA RTO"** has been successfully carried out by **Arjuna Reddy G** bearing USN **1RN17CS017** and **Likith Chandan M** bearing USN **1RN17CS047** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the **5th semester Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the mini-project requirements of DBMS lab of 5th semester BE in CSE.

Dr. Bhavanishankar K **Assistant Professor** **Dept. of CSE**	**Dr. G T Raju** **Vice Principal** **Prof. and Head** **Dept. of CSE**

External Viva:

Name of the Examiners	**Signature with Date**
1.	
2.	

ACKNOWLEDGMENTS

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We would like to thank **Dr. H N Shivashankar**, Director, RNSIT, Bengaluru, for his moral support towards completing our project.

We are grateful to **Dr. M K Venkatesha,** Principal, RNSIT, Bengaluru, for his support towards completing this mini project.

We would like to thank **Dr. G T Raju**, Vice Principal, Prof. & Head, Department of Computer Science & Engineering, RNSIT, Bengaluru, for his valuable suggestions and expert advice.

We deeply express my sincere gratitude to my guide **Dr. Bhavanishankar K, Dr. H .R. Shashidhara** and **Mrs. Mamatha Jajur**, Department of CSE, RNSIT, Bengaluru, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of department of Computer Science & Engineering, RNSIT, Bengaluru for their constant support and encouragement.

Date:	ARJUNA REDDY G	1RN17CS017
Place: Bengaluru	LIKITH CHANDAN M	1RN17CS047

ABSTRACT

RTO Office administration is a government transport organization and the main purpose of this office is to issue DL, for vehicle registration etc. In this project we propose the computerized system of managing all the data works. It helps to bring the smooth flow of work and also increase efficiency. This report highlights the requirements and also give guidelines that are necessary for the development of the project work mentioned above. This also gives guidelines in brief about how to design the project. Without a computerized system to maintain records any organisation may face difficulties in its functioning. To maintain records manually in paper to a very inefficient and tedious effort. Therefore, there is a need to implement a software that can efficiently do all this work. In RTO Office there is a need to maintain a large amount of data mainly like DL data, Registrations data, LL data etc and the proposed software will efficiently store and process the data according to the user's needs.

CONTENTS

Title	Page No.
1. **Introduction**	**1**
1.1 Database technologies	1
1.2 Characteristics of database approach	2
1.3 Applications of DBMS	3
1.4 Problem description/ statement	4
2. **Requirements Analysis**	**5**
2.1 Hardware Requirements	5
2.2 Software Requirements	5
2.3 Functional Requirements	5
2.4 End User Requirements	6
2.4.1 Java Swings	6
2.4.2 Java	8
2.4.3 MySQL	8
3. **Database Design**	**10**
3.1 Entities, Attributes and Relationships	10
3.2 ER Diagram	11
3.3 Relational Schema	12
4. **Implementation**	**13**
4.1 Database connectivity	13
4.2 Pseudo code For Major Functionalities	16
5. **Results, snapshots and discussions**	**33**
6. **Conclusion and Future Enhancements**	**38**

FIGURES USED

Figure No.	Page No.
Fig 3.2: ER diagram	11
Fig 3.3: Relational Schema	12
Fig 5.1: Login page	34
Fig 5.2: Control(main) page	34
Fig 5.3: Users page	35
Fig 5.4: Add_user page	35
Fig 5.5: DL page	36
Fig 5.6: LL page	36
Fig 5.7: Registrations page	37
Fig 5.8: Add_RC page	37
Fig 5.9: Transfers page	38
Fig 5.10: Add_Transfers page	38

CHAPTER 1

INTRODUCTION

1.1 DATABASE TECHNOLOGIES

The essential feature of database technology is that it provides an internal representation (model) of the external world of interest. Examples are the representation of a particular date/time/flight/aircraft in airline reservation or of item code/item description/quantity on hand/reorder level/reorder quantity in a stock control system. The technology involved is concerned primarily with maintaining the internal representation consistent with external reality; this involves the results of extensive R&D over the past 30 years in areas such as user requirements analysis, data modelling, process modelling, data integrity, concurrency, transactions, file organisation, indexing, rollback and recovery, persistent programming, object-orientation, logic programming, deductive database systems, active database systems... and in all these (and other) areas there remains much to be done.

The essential point is that database technology is a CORE TECHNOLOGY with links to:

- Information management / processing
- Data analysis / statistics
- Data visualization / presentation
- Multimedia and hypermedia
- Office and document systems
- Business processes, workflow, CSCW (computer-supported cooperative work)

Relational DBMS is the modern base technology for many business applications. It offers flexibility and easy-to-use tools at the expense of ultimate performance. More recently relational systems have started to extend their facilities in the directions of information retrieval, object-orientation and deductive/active systems leading to the so-called 'Extended Relational Systems'.

Information Retrieval Systems started with handling library catalogues and extended to full free-text utilizing inverted index technology with a lexicon or thesaurus. Modern systems utilize some KBS (knowledge-based systems) techniques to improve retrieval.

Object-Oriented DBMS started for engineering applications where objects are complex, have versions and need to be treated as a complete entity. OODBMSs share many of the OOPL features such as identity, inheritance, late binding, overloading and overriding. OODBMSs have found favour in engineering and office systems but have not yet been successful in traditional application areas.

Deductive / Active DBMS have emerged over the last 20 years and combine logic programming technology with database technology. This allows the database itself to react to external events an to maintain dynamically its integrity with respect to the real world.

1.2 CHARACTERISTICS OF DATABASE APPROACH

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics

- **Real-world entity** − A modern DBMS is more realistic and uses real world entities to design its architecture. It uses the behaviour and attributes too. For example, a school database may use students as an entity and their age as an attribute.
- **Relation-based tables** − DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.
- **Isolation of data and application** − A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.
- **Less redundancy** − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.
- **Consistency** − Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.
- **Query Language** − DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.
- **ACID Properties** − DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied on transactions, which

manipulate data in a database. ACID properties help the database stay healthy in multitransactional environments and in case of failure.

- **Multiuser and Concurrent Access** − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them. o Multiple views − DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements. o Security − Features like multiple views offer security to some extent where users are unable to access data of other users and departments.

DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

1.3 APPLICATIONS OF DBMS

Applications where we use Database Management Systems are:

- **Telecom**: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
- **Industry**: Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System**: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.

- **Education sector**: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.
- **Online shopping**: You must be aware of the online shopping websites such as Amazon, Flip kart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

1.4 PROBLEM DESCRIPTION/STATEMENT

A manual management of data without help of a software in the field of RTO management an lead to many mistakes some of which may have serious consequences. Therefore, there is a need to implement an efficient software to do this work and prevent those mistakes.

In such cases this project provides a software to handle the work. The main aim of this project is to create a database that can store and manage various RTO related information like Driver’s License, Learner’s License, Vehicle Registrations etc.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 HARDWARE REQUIREMENTS

The Hardware requirements are very minimal and the program can be run on most of the machines.

• Processor	:	Pentium4 processor or higher
• Processor Speed	:	2.4 GHz
• RAM	:	1 GB
• Storage Space	:	40 GB
• Monitor Resolution	:	1024*768 or 1336*768 or 1280*1024

2.2 SOFTWARE REQUIREMENTS

• Operating System	:	Windows XP or higher
• Front end	:	Java Swing Application
• Back end	:	Java – Eclipse/Xampp MySQL

2.3FUNCTIONAL REQUIREMENTS

- The system must have a password protected access system such that only people with authenticated credential are allowed to access the functions of the system.
- The system must allow the users to add new data into the DL, LL, Registrations, Transfers tables.
- The system must provide functions for displaying the contents of the above mentioned tables.
- The system must automatically create a DL record from the LL details of a client when he passes the driving test.
- The system is a stand-alone software and not a web-based application.

2.4 END USER REQUIREMENTS

The main users of the product would be RTO personnel only. They should have login credential so that only selected staff can access the software. The RTO personnel has the responsibility of adding new users, maintaining and adding LL, DL, RC and transfers information.

2.4.1 Java Swings

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element.

Swing is a highly modular-based architecture, which allows for the "plugging" of various custom implementations of specified framework interfaces: Users can provide their own custom implementation(s) of these components to override the default implementations using Java's inheritance mechanism.

Swing is a **component-based framework**, whose components are all ultimately derived from the javax.swing.JComponent class. Swing objects asynchronously fire events, have bound properties, and respond to a documented set of methods specific to the component. Swing components are Java Beans components, compliant with the Java Beans Component
Architecture specifications - Since early versions of Java, a portion of the Abstract Window Toolkit (AWT) has provided platform-independent APIs for user interface components. In AWT, each

component is rendered and controlled by a native peer component specific to the underlying windowing system.

By contrast, Swing components are often described as *lightweight* because they do not require allocation of native resources in the operating system's windowing toolkit. The AWT components are referred to as heavyweight components.

Much of the Swing API is generally a complementary extension of the AWT rather than a direct replacement. In fact, every Swing lightweight interface ultimately exists within an AWT heavyweight component because all of the top-level components in Swing (JApplet, JDialog, JFrame, and JWindow) extend an AWT top-level container.

Swing's high level of flexibility is reflected in its inherent ability to override the native host operating system (OS)'s GUI controls for displaying itself. Swing "paints" its controls using the Java 2D APIs, rather than calling a native user interface toolkit. Thus, a Swing component does not have a corresponding native OS GUI component, and is free to render itself in any way that is possible with the underlying graphics GUIs.

However, at its core, every Swing component relies on an AWT container, since (Swing's) JComponent extends (AWT's) Container. This allows Swing to plug into the host OS's GUI management framework, including the crucial device/screen mappings and user interactions, such as key presses or mouse movements. Swing simply "transposes" its own (OS-agnostic) semantics over the underlying (OS-specific) components. So, for example, every Swing component paints its rendition on the graphic device in response to a call to component.paint(), which is defined in (AWT) Container. But unlike AWT components, which delegated the painting to their OS-native "heavyweight" widget, Swing components are responsible for their own rendering.

This transposition and decoupling are not merely visual, and extends to Swing's management and application of its own OS-independent semantics for events fired within its component containment hierarchies. Generally speaking, the Swing architecture delegates the task of mapping the various flavours of OS GUI semantics onto a simple, but generalized, pattern to the AWT container. Building on that generalized platform, it establishes its own rich and complex GUI semantics in the form of the JComponent model.

2.4.2 Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer lowlevel facilities than either of them.

One design goal of Java is portability, which means that programs written for the Java platform must run similarly on any combination of hardware and operating system with adequate runtime support. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to architecture-specific machine code. Java bytecode instructions are analogous to machine code, but they are intended to be executed by a virtual machine (VM) written specifically for the host hardware. End users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a web browser for Java applets.

2.4.3 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second-generation Open Source company that unites Open Source values and methodology with a successful business model.

- MySQL is a database management system.
  Database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database

management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- MySQL is a relational database management system.
  Relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.
- MySQL software is Open Source.
  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. The MySQL Database Server is very fast, reliable, and easy to use.
- MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years.

CHAPTER 3

DATABASE DESIGN

3.1 Entities, Attributes and Relationships

- **Admin Entity:** Name, Adminid, Password, Phone
- **Users Entity:** Fname, Mname, Lname, Aadhar, Address, Phno, D_test
- **Registration Entity:** Regno, Engno, Fuel, Cov, Doi, Doe, Model, Ownerid
- **Driving_Licence Entity:** DLno, Cov, Doi, Doe, Usr_aadhar
- **Learning_Licencen Entity:** LLno, Cov, Doi, Doe, T_date, Usr_aadhar

3.2 ER Diagram

The fig 3.2 represents the Entity-Relationship diagram

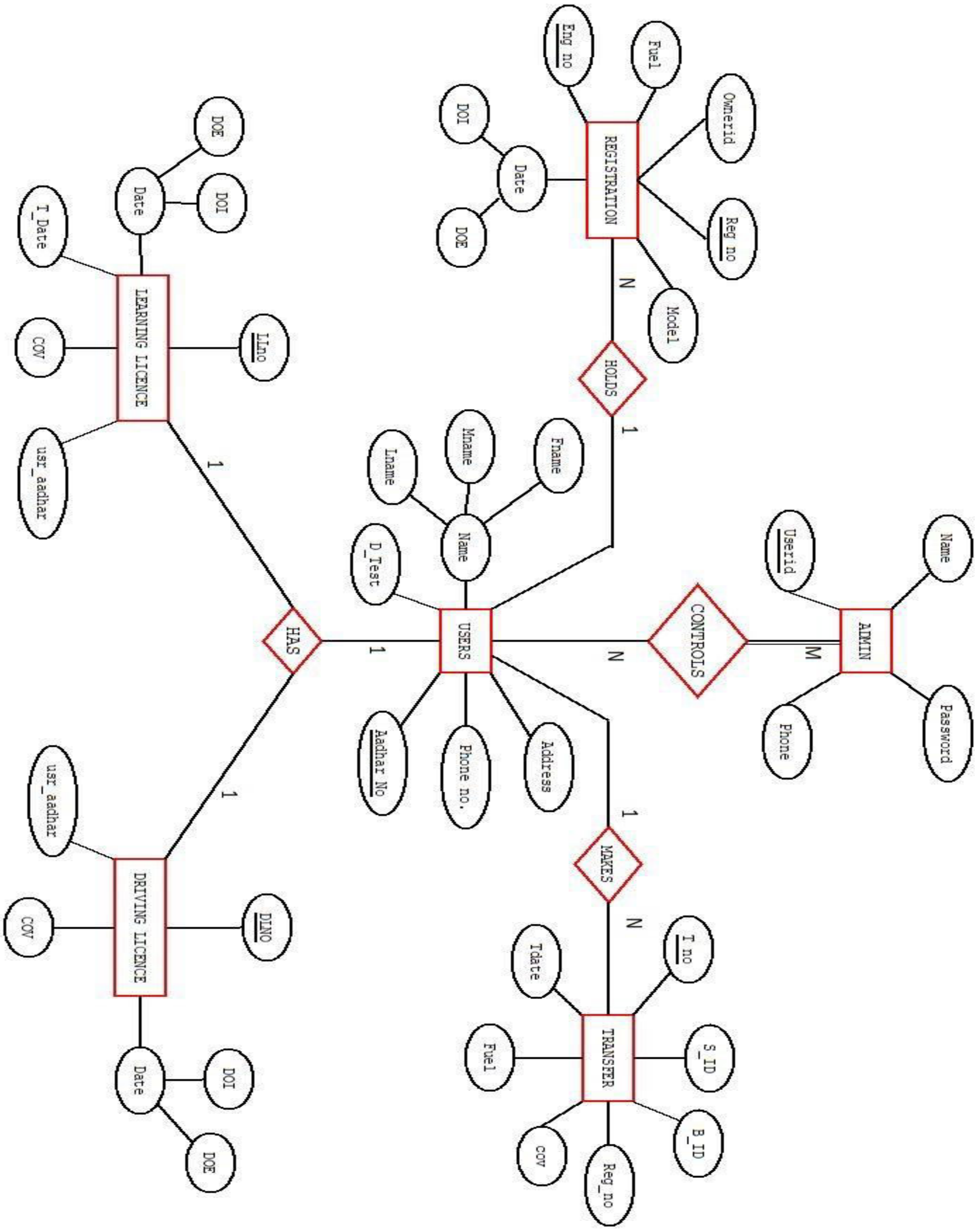


Fig 3.2

3.3 Relational Schema

The fig 3.3 shows the Relational Schema

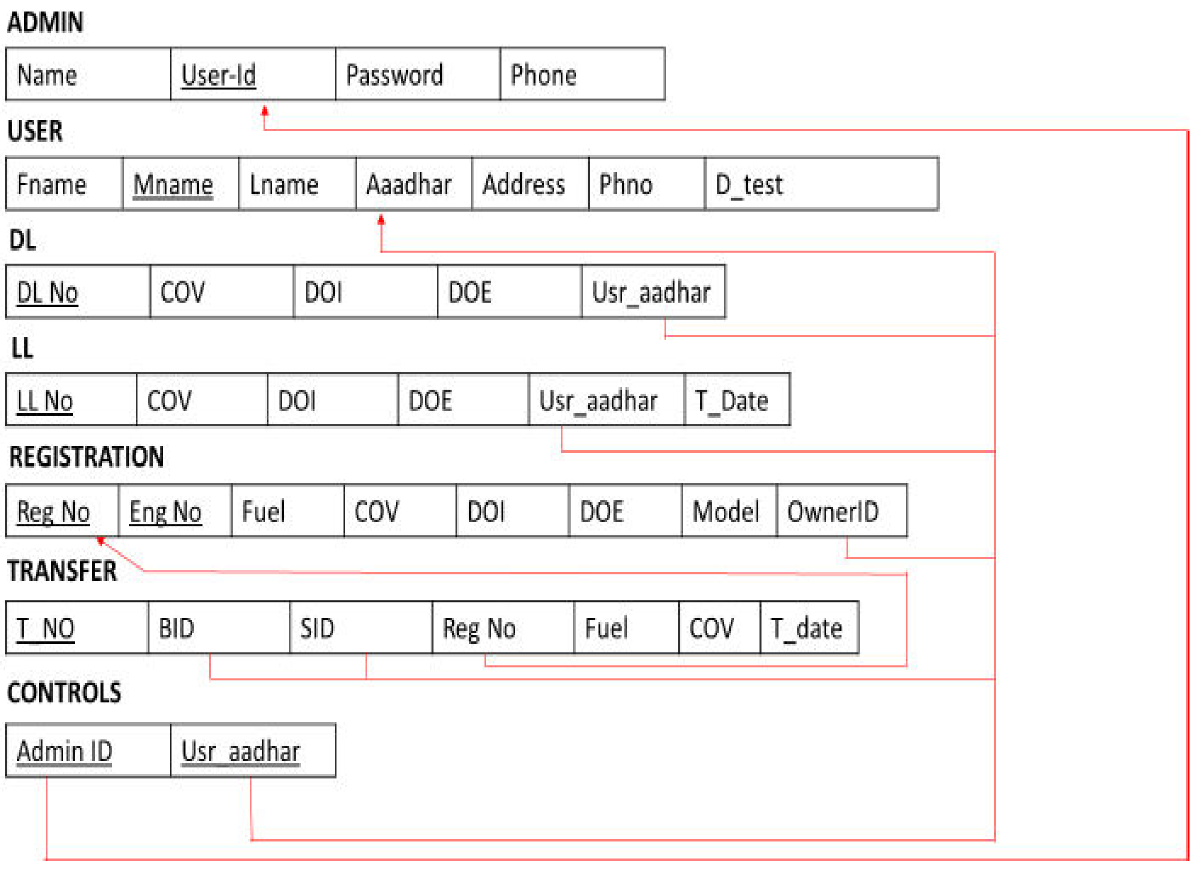


Fig. 3.3

CHAPTER 4

IMPLEMENTATION

4.1 DATABASE CONNECTIVITY

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

FUNCTIONALITY

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection. JDBC represents statements using one of the following classes:

- Statement – the statement is sent to the database server each and every time.
- Prepared Statement – the statement is cached and then the execution path is predetermined on the database server allowing it to be executed multiple times in an efficient manner.
- Callable Statement – used for executing stored procedures on the database.

Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many rows were affected in the database. These statements do not return any other information.

Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

There is an extension to the basic JDBC API in the javax.sql.

JDBC connections are often managed via a connection pool rather than obtained directly from the driver.

JDBC DRIVERS

JDBC drivers are client-side adapters (installed on the client machine, not on the server) that convert requests from Java programs to a protocol that the DBMS can understand. Types Commercial and free drivers provide connectivity to most relational-database servers. These drivers fall into one of the following types:

a. Type 1 that calls native code of the locally available ODBC driver.

b. Type 2 that calls database vendor native library on a client side. This code then talks to database over the network.

c. Type 3, the pure-java driver that talks with the server-side middleware that then talks to the database.

d. Type 4, the pure-java driver that uses database native protocol.

Note also a type called an internal JDBC driver - a driver embedded with JRE in Java-enabled SQL databases. It is used for Java stored procedures.

Steps to connect to the database in java

There are 5 steps to connect any java application with the database in java using JDBC.

They are as follows:

1. Register the driver class
2. Creating connection
3. Creating statement

4. Executing queries

5. Closing connection

Register the driver class

The forName() method of Class class is used to register the driver class. This method is used to dynamically load the driver class.

Syntax of forName() method :

public static void forName(String className)throws ClassNotFoundException .

Create the connection object

The getConnection() method of DriverManager class is used to establish connection with the database.

Syntax of getConnection() method:

- public static Connection getConnection(String url)throws SQLException
- public static Connection getConnection(String url,String name,String password) throws SQLException

Create the Statement object

The createStatement() method of Connection interface is used to create statement. The object of statement is responsible to execute queries with the database.

Syntax of createStatement() method:

public Statement createStatement()throws SQLException

Execute the query

The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.

Syntax of executeQuery() method:

public ResultSet executeQuery(String sql)throws SQLException

Close the connection object

By closing connection object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

Syntax of close() method:

```
public void close()throws SQLException
```

4.2 Pseudo Code for Major Functionalities

Trigger:

```
CREATE TRIGGER `trig1` AFTER UPDATE ON `users` FOR EACH    ROW
BEGIN
 IF (NEW.D_Test="pass")
 THEN
 CALL test1(NEW.aadhar);
END IF
END
```

Stored Procedure:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `test1` (IN `id` INT)  BEGIN

INSERT INTO drivinglicense(cov,doi,usr_aadhar)

SELECT cov,T_Date,usr_aadhar

FROM learninglicence

WHERE usr_aadhar_=id;
```

DELETE FROM learninglicence where usr_aadhar;

END

Pseudo Code:

Admin page

```
public class Admin_Login
{

        JFrame frame;
        private JTextField textField;
        private JPasswordField passwordField;
        private Container c;

        // Launching the application
        public static void main(String[] args)
        {
                EventQueue.invokeLater(new Runnable()
                {
                        public void run()
                        {
                                try
                                {
                                        Admin_Login window = new Admin_Login();
                                        window.frame.setVisible(true);
                                } catch (Exception e)
                                {
                                        e.printStackTrace();
                                }
                        }
                });
        }

        // Creating the application
        public Admin_Login()
        {
                initialize();
        }

        // Initializing the contents of the frame
        private void initialize()
        {
                JButton btnLogin = new JButton("Login");
                btnLogin.setBackground(Color.DARK_GRAY);
                btnLogin.setForeground(new Color(255, 255, 255));
                btnLogin.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {
                                if (e.getSource() == btnLogin) {
                                        try
                            {
                              Class.forName("com.mysql.jdbc.Driver");
```

```
                                                Connection conn = null;
                                                conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                                                System.out.print("Database is connected !");
                                                Statement stmt = conn.createStatement();

                                                // SELECT query
                                                @SuppressWarnings("deprecation")
                                                                String q1 = "select * from admin WHERE adminid= '"
                                                +textField.getText() + "' AND password = '" + passwordField.~~getText~~() + "'";
                                                ResultSet rs = stmt.executeQuery(q1);
                                                if (rs.next())
                                                {
                                                  JOptionPane.showMessageDialog(frame,"login  successful");
                                                    Controls ab=new Controls();
                                                     frame.setVisible(false);
                                                    ab.frame.setVisible(true);
                                                }
                                                else
                                                {
                                                  JOptionPane.showMessageDialog(frame,"Wrong password or adminid");
                                                }
                                                conn.close();
                                            }
                                            catch(Exception ex)
                                            {
                                                System.out.println(ex);
                                            }
                                        }
                                }
                        });
                        btnLogin.setFont(new Font("Sylfaen", Font.BOLD, 22));
                        btnLogin.setBounds(557, 385, 110, 42);
                        frame.getContentPane().add(btnLogin);

                        JLabel lblNewLabel = new JLabel("New label");
                        lblNewLabel.setForeground(new Color(128, 0, 0));
                        lblNewLabel.setIcon(newImageIcon("C:\\Users\\LC\\eclipse-
workspace\\RTOproject\\images\\pic1.jpg"));
                        lblNewLabel.setBounds(0, 0, 1182, 686);
                        frame.getContentPane().add(lblNewLabel);
                }
}
```

Controls page

```
public class Controls
{
JFrame frame;
        /**
         * Launch the application.
         */
        public static void main(String[] args)
        {
                EventQueue.invokeLater(new Runnable()
                {
                        public void run()
```

```
                    {
                              try
                              {
                                        Controls window = new Controls();
                                        window.frame.setVisible(true);
                              } catch (Exception e)
                              {
                                        e.printStackTrace();
                              }
                    }
          });
}

/**
 * Create the application.
 */
public Controls()
{
          initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize()
{
          JButton btnUsers = new JButton("Users");
          btnUsers.setFont(new Font("Sylfaen", Font.PLAIN, 20));
          btnUsers.addActionListener(new ActionListener()
          {
                    public void actionPerformed(ActionEvent arg0)
                    {
                              frame.setVisible(false);
                              Users usr=new Users();
                              usr.frame.setVisible(true);
                    }
          });
          btnUsers.setBounds(497, 207, 176, 44);
          frame.getContentPane().add(btnUsers);

          JButton btnRegistrations = new JButton("Registrations");
          btnRegistrations.setFont(new Font("Sylfaen", Font.PLAIN, 20));
          btnRegistrations.addActionListener(new ActionListener()
          {
                    public void actionPerformed(ActionEvent arg0)
                    {
                              frame.setVisible(false);
                              Registrations usr=new Registrations();
                              usr.frame.setVisible(true);
                    }
          });
          btnRegistrations.setBounds(497, 321, 176, 44);
          frame.getContentPane().add(btnRegistrations);

          JButton btnTransfers = new JButton("Transfers");
          btnTransfers.setFont(new Font("Sylfaen", Font.PLAIN, 20));
          btnTransfers.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent arg0)
                        {
                                frame.setVisible(false);
                                Transfer usr=new Transfer();
                                usr.frame.setVisible(true);
                        }
                });
                btnTransfers.setBounds(497, 438, 176, 44);
                frame.getContentPane().add(btnTransfers);

                JButton btnPrev = new JButton("prev");
                btnPrev.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnPrev.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent arg0)
                        {
                                frame.setVisible(false);
                                Admin_Login al=new Admin_Login();
                                al.frame.setVisible(true);
                        }
                });
                btnPrev.setBounds(0, 642, 102, 44);
                frame.getContentPane().add(btnPrev);

                JLabel lblNewLabel = new JLabel("New label");
                lblNewLabel.setIcon(new ImageIcon ("C:\\Users\\LC\\eclipseworkspace\\RTOproject
                                                        \\images\\pic8alter.png"));
                lblNewLabel.setBounds(0, 0, 1182, 686);
                frame.getContentPane().add(lblNewLabel);
        }
}
```

Users page

```
public class Users
{
    /*
         Create the application.
         */
        public Users()
        {
                initialize();
        }

        /**
         * Initialize the contents of the frame.
         */
        private void initialize()
        {
                frame = new JFrame();
                frame.setBounds(100, 100, 1200, 733);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.getContentPane().setLayout(null);

                JTextArea textArea = new JTextArea();
                textArea.setTabSize(15);
                textArea.setWrapStyleWord(true);
                textArea.setFont(new Font("Monospaced", Font.PLAIN, 16));
                textArea.setBounds(0, 0, 1182, 531);
                frame.getContentPane().add(textArea);
                try
                {
                        Class.forName("com.mysql.jdbc.Driver");
        Connection conn = null;
        conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
        System.out.print("Database is connected !");
        Statement stmt = conn.createStatement();
        String q1="select * from users;";
        ResultSet rs = stmt.executeQuery(q1);
        textArea.setText("FName\t  MName\t  LName\t  Phone_No\t  Address\t  Aadhar\n");
        while(rs.next())
        {
          String pre=textArea.getText();
          String      data1=rs.getString("FName");String      data2      =      rs.getString("MName");String      data3      =
rs.getString("LName");
          String data4 = rs.getString("aadhar");String data5 = rs.getString("address");String data6 = rs.getString("phno");
          textArea.setText(pre+data1+"\t  "+data2+"\t  "+data3+"\t  "+data6+"\t  "+data5+"\t  "+data4+"\n");
        }
                        conn.close();
                }
                catch(Exception e) {};

                JButton btnAddUser = new JButton("Add User");
                btnAddUser.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnAddUser.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent arg0)
                        {
                                frame.setVisible(false);
                                Add_User usr=new Add_User();
                                usr.frame.setVisible(true);
                        }
                });
                btnAddUser.setBounds(190, 563, 136, 48);
                frame.getContentPane().add(btnAddUser);

                JButton btnDl = new JButton("DL");
                btnDl.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnDl.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent arg0)
                        {
                                frame.setVisible(false);
                                Driving_License usr=new Driving_License();
                                usr.frame.setVisible(true);
                        }
                });
                btnDl.setBounds(510, 563, 136, 48);
                frame.getContentPane().add(btnDl);

                JButton btnLl = new JButton("LL");
                btnLl.setFont(new Font("Sylfaen", Font.PLAIN, 20));
```

```
		btnLl.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent arg0)
			{
				frame.setVisible(false);
				Learning_License usr=new Learning_License();
				usr.frame.setVisible(true);
			}
		});
		btnLl.setBounds(838, 563, 136, 48);
		frame.getContentPane().add(btnLl);

		JButton btnPrev = new JButton("prev");
		btnPrev.setFont(new Font("Sylfaen", Font.PLAIN, 20));
		btnPrev.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent arg0)
			{
				Controls c=new Controls();
				c.frame.setVisible(true);
				frame.setVisible(false);
			}
		});
		btnPrev.setBounds(0, 647, 93, 39);
		frame.getContentPane().add(btnPrev);

		JLabel lblNewLabel = new JLabel("New label");
		lblNewLabel.setIcon(newImageIcon("C:\\Users\\LC\\eclipse-
						workspace\\RTOproject\\images\\pic8.png"));
		lblNewLabel.setBounds(0, 0, 1182, 686);
		frame.getContentPane().add(lblNewLabel);
	}
}
```

Add_User page

```
public class Add_User
{
	/**
	 * Create the application.
	 */
	public Add_User()
	{
		initialize();
	}

	/**
	 * Initialize the contents of the frame.
	 */
	private void initialize()
	{
		JButton btnPrev = new JButton("prev");
		btnPrev.setIcon(null);
		btnPrev.setFont(new Font("Tahoma", Font.PLAIN, 20));
		btnPrev.setBounds(0, 643, 106, 43);
		btnPrev.addActionListener(new ActionListener()
{
        public void actionPerformed(ActionEvent arg0)
        {
                Users u=new Users();
                frame.setVisible(false);
                u.frame.setVisible(true);
        }
});
frame.getContentPane().setLayout(null);
frame.getContentPane().add(btnPrev);

JButton btnSubmit = new JButton("Submit");
btnSubmit.addActionListener(new ActionListener()
{
        public void actionPerformed(ActionEvent e)
        {

                if (e.getSource() == btnSubmit)
                {
                        try
            {
                                Class.forName("com.mysql.jdbc.Driver");
                        Connection conn = null;
                        conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                        System.out.print("Database is connected !");
                        Statement stmt = conn.createStatement();

                        String q1 = "insert into users(FName,MName,LName,aadhar,address,phno)
                                  values('"+textField.getText()+"','"+textField_4.getText()+"','"+
                                  textField_5.getText()+"','"+textField_1.getText()+"','"+
                                 textField_2.getText()+"','"+textField_3.getText()+"')";

                        int a=stmt.executeUpdate(q1);
                        label.setText("User Successfully Inserted.");
                        conn.close();
            }
            catch(Exception ex)
            {
              System.out.println(ex);
            }
            }
        }
});
btnSubmit.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnSubmit.setBounds(460, 480, 129, 42);

frame.getContentPane().add(btnSubmit);

JButton btnUpdate = new JButton("Update");
btnUpdate.addActionListener(new ActionListener()
{
        public void actionPerformed(ActionEvent e)
        {

                if (e.getSource() == btnUpdate)
                {
                        try
                        {
                                        Class.forName("com.mysql.jdbc.Driver");
                                Connection conn = null;
                                conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                                System.out.print("Database is connected !");
                                Statement stmt = conn.createStatement();

                                String q1 = "update users set D_Test='" +textField_6.getText() + "'  where
                                                aadhar="+textField_1.getText();
                                int a=stmt.executeUpdate(q1);
                                label.setText("User Successfully Inserted.");
                                conn.close();
                        }
                        catch(Exception ex)
                        {
                          System.out.println(ex);
                        }
                        }
                    }
            });
            btnUpdate.setFont(new Font("Tahoma", Font.PLAIN, 20));
            btnUpdate.setBounds(613, 480, 129, 42);

            frame.getContentPane().add(btnUpdate);



            JLabel lblNewLabel = new JLabel("New label");
            lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 11));
            lblNewLabel.setIcon(newImageIcon("C:\\Users\\LC\\eclipse-
                            workspace\\RTOproject\\images\\pic8alter.png"));
            lblNewLabel.setBounds(0, 11, 1182, 686);
            frame.getContentPane().add(lblNewLabel);

    }
}
```

Add_RC page

```
public class Add_RC
{

    /**
     * Create the application.
     */
    public Add_RC()
    {
            initialize();
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize()
    {
            JButton btnSubmit = new JButton("Submit");
```

```
                btnSubmit.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnSubmit.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {

                                if (e.getSource() == btnSubmit)
                                {
                                        try
                            {
                                                     Class.forName("com.mysql.jdbc.Driver");
                                      Connection conn = null;
                                      conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                                      System.out.print("Database is connected !");
                                      Statement stmt = conn.createStatement();

                                      String q1 = "insert into rc values('" +textField.getText() +
                                                   "','" + textField_1.getText()+"','"+textField_2.getText()+
                                                   "','"+textField_3.getText()+"','"+textField_4.getText()+
                                                   "','"+textField_5.getText()+"','"+textField_6.getText()+
                                                   "',"+textField_7.getText()+")";
                                      int a=stmt.executeUpdate(q1);
                                      conn.close();
                            }
                            catch(Exception ex)
                            {
                               System.out.println(ex);
                            }
                            }
                        }
                });
                btnSubmit.setBounds(513, 576, 127, 44);
                frame.getContentPane().add(btnSubmit);

                JButton btnPrev = new JButton("prev");
                btnPrev.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnPrev.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent arg0)
                        {
                                frame.setVisible(false);
                                Registrations r=new Registrations();
                                r.frame.setVisible(true);
                        }
                });
                btnPrev.setBounds(0, 649, 86, 37);
                frame.getContentPane().add(btnPrev);
                JLabel lblNewLabel_1 = new JLabel("New label");
                lblNewLabel_1.setForeground(Color.WHITE);
                lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 20));
                lblNewLabel_1.setIcon(newImageIcon("C:\\Users\\LC\\eclipse-
                                                 workspace\\RTOproject\\images\\pic8alter.png"));
                lblNewLabel_1.setBounds(0, -1, 1182, 686);
                frame.getContentPane().add(lblNewLabel_1);
        }

}
```

DL page

```
public class Driving_License {

	String pre;

	/**
	 * Create the application.
	 */
	public Driving_License() {
		initialize();
	}
	public void view(JTextArea textArea) {
		try
	  {
				Class.forName("com.mysql.jdbc.Driver");
		  Connection conn = null;
		  conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
		  System.out.print("Database is connected !");
		  Statement stmt1 = conn.createStatement();
		  String q2="select * from drivinglicense;";
		  ResultSet rs = stmt1.executeQuery(q2);
		  textArea.setText("DL_No\tCOV\tDOI\tExpiry_Date\tUser_aadhar\n");
		  while(rs.next())
		  {
		     pre=textArea.getText();
		   String data1 = rs.getString("DLno");String data2 = rs.getString("cov");String data3 =
rs.getString("doi");
		   String data4 = rs.getString("doe");String data5 = rs.getString("usr_aadhar");
		   textArea.setText(pre+data1+"\t"+data2+"\t"+data3+"\t"+data4+"\t"+data5+"\t"+"\n");

		  }
		  conn.close();
	    }
	    catch(Exception ex)
	    {
	       System.out.println(ex);
	    }

	}

	/**
	 * Initialize the contents of the frame.
	 */
	private void initialize() {
		JButton btnSubmit = new JButton("Submit");
		btnSubmit.setFont(new Font("Sylfaen", Font.PLAIN, 20));
		btnSubmit.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{

				if (e.getSource() == btnSubmit)
				{
					try
```

```
                                        {
                                            Class.forName("com.mysql.jdbc.Driver");
                                        Connection conn = null;
                                        conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                                        System.out.print("Database is connected !");
                                        Statement stmt = conn.createStatement();
                                        String q1 = "update drivinglicense set DLno='" +textField.getText() +
                                                    "',doe='"+textField_3.getText()+"'where
                                                            usr_aadhar="+textField_4.getText();
                                        int a=stmt.executeUpdate(q1);

                                        Statement stmt1 = conn.createStatement();

                                        String q2="select * from drivinglicense;";
                                        ResultSet rs = stmt1.executeQuery(q2);
                                        if(rs.last())
                                        {
                                          String data1 = rs.getString("DLno");String data2 = rs.getString("cov");String
                                           data3 = rs.getString("doi");String data4 = rs.getString("doe");String data5 =
                                           rs.getString("usr_aadhar");
                                           textArea.setText("DL_No\t\tCOV\tDOI\tExpiry
                                                   Date\tUser_aadhar\n"+data1+"\t\t"+data2+"\t"+data3+"\t"+data4+"\
                                                   t"+data5+"\t"+"\n");
                                        }
                                        conn.close();
                                }
                                catch(Exception ex)
                                {
                                   System.out.println(ex);
                                }
                                }
                        }
                });

                btnSubmit.setBounds(223, 436, 116, 39);
                frame.getContentPane().add(btnSubmit);

                JButton btnPrev = new JButton("prev");
                btnPrev.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnPrev.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent arg0)
                        {
                                Users u=new Users();
                                frame.setVisible(false);
                                u.frame.setVisible(true);
                        }
                });
                btnPrev.setBounds(0, 647, 84, 39);
                frame.getContentPane().add(btnPrev);



                JButton button = new JButton("View All");
                button.setBackground(UIManager.getColor("Button.background"));
                button.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                button.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {

                                if (e.getSource() == button)
                                {
                                        view(textArea);
                                 }
                        }
                });
                button.setBounds(89, 510, 153, 39);
                frame.getContentPane().add(button);

                JButton button1 = new JButton("New Entries");
                button1.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                button1.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {

                                if (e.getSource() == button1)
                                {
                                        try
                            {
                                                    Class.forName("com.mysql.jdbc.Driver");
                                      Connection conn = null;
                                      conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                                      System.out.print("Database is connected !");
                                      Statement stmt1 = conn.createStatement();

                                      String q2="select usr_aadhar from drivinglicense where DLno='NULL' ";
                                      ResultSet rs = stmt1.executeQuery(q2);
                                      textArea.setText("User_aadhar\n");
                                      while(rs.next())
                                      {
                                        String data1 = rs.getString("usr_aadhar");
                                        textArea.setText(textArea.getText()+"\n"+data1+"\n");
                                      }
                                      conn.close();
                            }
                            catch(Exception ex)
                            {
                               System.out.println(ex);
                            }

                          }
                        }
                });
                button1.setBounds(317, 509, 153, 40);
                frame.getContentPane().add(button1);

                JLabel lblNewLabel = new JLabel("New label");
                lblNewLabel.setIcon(new ImageIcon("C:\\Users\\LC\\eclipse-workspace\\RTOproject\\images\\pic8 –
                Copy (2).png"));
                lblNewLabel.setBounds(0, 11, 1182, 686);
                frame.getContentPane().add(lblNewLabel);
        }
```

```
}
```

LL page

```
public class Learning_License
{
            String pre;

            /**
             * Create the application.
             */
            public Learning_License()
            {
                        initialize();
            }

            /**
             * Initialize the contents of the frame.
             */
            private void initialize()
            {
                        frame = new JFrame();
                        frame.setBounds(100, 100, 1200, 733);
                        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                        frame.getContentPane().setLayout(null);

                        JTextArea textArea = new JTextArea();
                        textArea.setTabSize(12);
                        textArea.setFont(new Font("Monospaced", Font.PLAIN, 15));
                        textArea.setBounds(487, 0, 697, 686);
                        frame.getContentPane().add(textArea);
                        textArea.setText(" LL_No\tCOV\tDOI\tExp_Date\t T_Date\tUser_aadhar\n");

                        try
                        {
                                    Class.forName("com.mysql.jdbc.Driver");
                      Connection conn = null;
                      conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                      System.out.print("Database is connected !");
                      Statement stmt1 = conn.createStatement();
                      String q2="select * from learninglicence;";
                      ResultSet rs = stmt1.executeQuery(q2);
                      while(rs.next())
                       {
                        pre=textArea.getText();
                        String data1 = rs.getString("LLno");String data2 = rs.getString("cov");String data3 =
rs.getString("doi");
                        String data4 = rs.getString("doe");String data5 = rs.getString("usr_aadhar");String data6=
                                                                                    rs.getString("T_Date");
                        textArea.setText(pre+" "+data1+"\t"+data2+"\t"+data3+"\t"+data4+"\t "+data6+"\t"+data5+"\n");

                       }
                      conn.close();
        }
        catch(Exception ex)
        {
```

```
        System.out.println(ex);
    }

                JButton btnSubmit = new JButton("Submit");
                btnSubmit.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnSubmit.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {
                                //2017-01-26   2014-05-23
                                if (e.getSource() == btnSubmit)
                                {
                                    try
                                    {
                                                Class.forName("com.mysql.jdbc.Driver");
                                        Connection conn = null;
                                        conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                                        System.out.print("Database is connected !");
                                        Statement stmt = conn.createStatement();
                                        String q1 = "insert into learninglicence values('" +textField.getText() +
                                                     "','" + textField_1.getText()+"','"+textField_2.getText()+
                                                     "','"+textField_3.getText()+"','"+textField_4.getText()+"','"
                                                         +textField_5.getText()+"')";
                                        int a=stmt.executeUpdate(q1);

                                        Statement stmt1 = conn.createStatement();

                                        String q2="select * from learninglicence;";
                                        ResultSet rs = stmt1.executeQuery(q2);
                                        if(rs.last())
                                        {
                                          String data1 = rs.getString("LLno");String data2 = rs.getString("cov");String
                                          data3 = rs.getString("doi");
                                          String data4 = rs.getString("doe");String data5 = rs.getString("usr_aadhar");
                                          textArea.setText(textArea.getText()+data1+"\t"+data2+"\t"+data3
                                                                   +"\t"+data4+"\t"+data5+"\t"+"\n");
                                        }
                                        conn.close();
                                    }
                                catch(Exception ex)
                                {
                                   System.out.println(ex);
                                }
                                }
                        }
                });
                btnSubmit.setBounds(238, 538, 117, 38);
                frame.getContentPane().add(btnSubmit);

                JButton btnPrev = new JButton("prev");
                btnPrev.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnPrev.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent arg0)
                        {
                                Users u=new Users();
                                frame.setVisible(false);
                                u.frame.setVisible(true);
                        }
                });
                btnPrev.setBounds(0, 647, 83, 39);
                frame.getContentPane().add(btnPrev);




                JLabel lblNewLabel = new JLabel("New label");
                lblNewLabel.setIcon(new ImageIcon("C:\\Users\\LC\\eclipse-workspace\\RTOproject\\images\\pic8 –
                                                                        Copy (2).png"));
                lblNewLabel.setBounds(0, 0, 1182, 686);
                frame.getContentPane().add(lblNewLabel);

        }

}
```

Add_Transfer page

```
public class Add_Transfer
{
        /**
         * Create the application.
         */
        public Add_Transfer()
        {
                initialize();
        }

        /**
         * Initialize the contents of the frame.
         */
        private void initialize()
        {


                JButton btnSubmit = new JButton("Submit");
                btnSubmit.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnSubmit.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {

                                if (e.getSource() == btnSubmit)
                                {
                                    try
                                    {
                                                Class.forName("com.mysql.jdbc.Driver");
                                        Connection conn = null;
                                        conn = DriverManager.getConnection("jdbc:mysql://localhost/rto","root", "");
                                        System.out.print("Database is connected !");
                                        Statement stmt = conn.createStatement();

                                        String q1 = "insert into transfer values( "+textField.getText() +
                                                     "," + textField_1.getText()+","+textField_2.getText()+
                                                     ",'"+textField_3.getText()+"','"+textField_4.getText()+
                                                     "','"+textField_5.getText()+"','"+textField_6.getText()+"')";
                                          int a=stmt.executeUpdate(q1);
                                          conn.close();
                                  }
                                  catch(Exception ex)
                                  {
                                     System.out.println(ex);
                                  }
                                  }
                          }
                });
                btnSubmit.setBounds(564, 568, 130, 45);
                frame.getContentPane().add(btnSubmit);

                JButton btnPrev = new JButton("prev");
                btnPrev.setFont(new Font("Sylfaen", Font.PLAIN, 20));
                btnPrev.addActionListener(new ActionListener()
                {
                          public void actionPerformed(ActionEvent arg0)
                          {
                                  frame.setVisible(false);
                                  Transfer t=new Transfer();
                                  t.frame.setVisible(true);
                          }
                });
                btnPrev.setBounds(0, 650, 98, 36);
                frame.getContentPane().add(btnPrev);

                JLabel lblNewLabel_1 = new JLabel("New label");
                lblNewLabel_1.setIcon(newImageIcon("C:\\Users\\LC\\eclipse-
                                                             workspace\\RTOproject\\images\\pic8alter.png"));
                lblNewLabel_1.setBounds(0, -1, 1182, 686);
                frame.getContentPane().add(lblNewLabel_1);


        }
```

CHAPTER 5

RESULTS, SNAPSHOTS AND DISCUSSIONS

Fig 5.1 allows user(admin) to login into the software

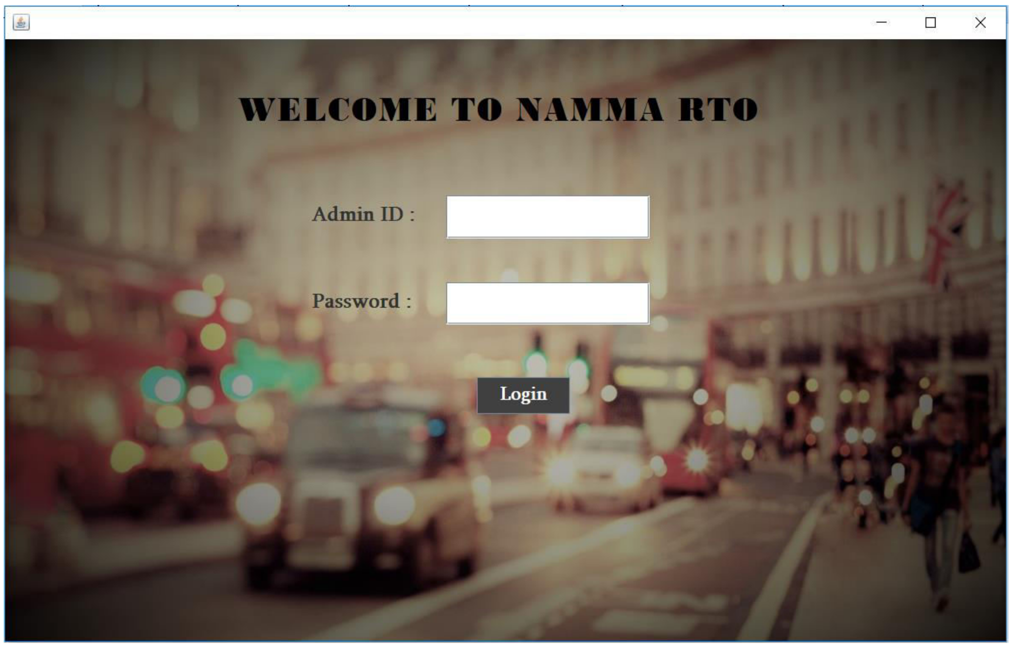


Fig 5.1

Fig 5.2 shows the controls page which allows the admin to choose different functionalities

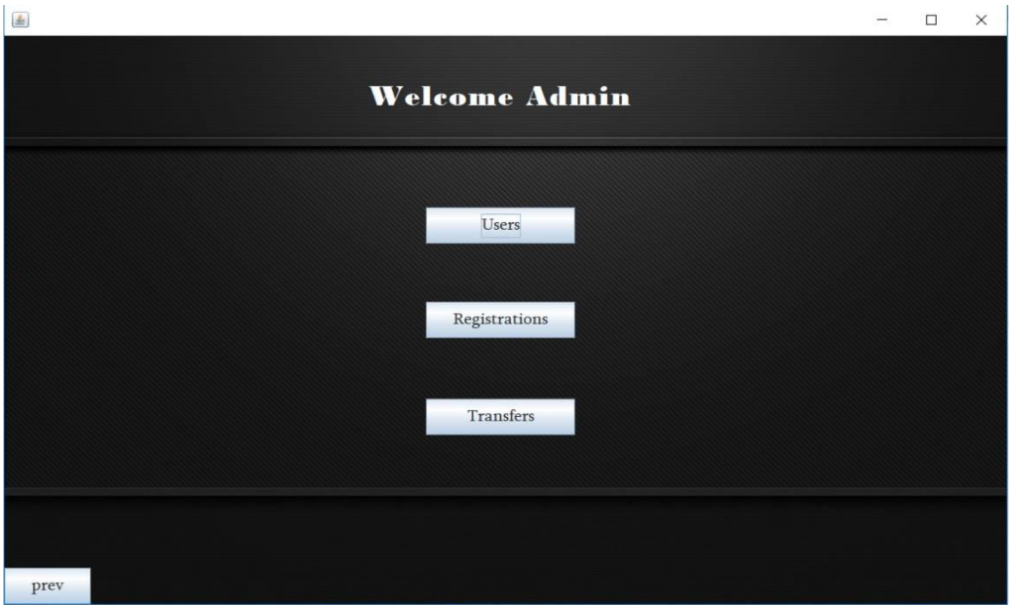
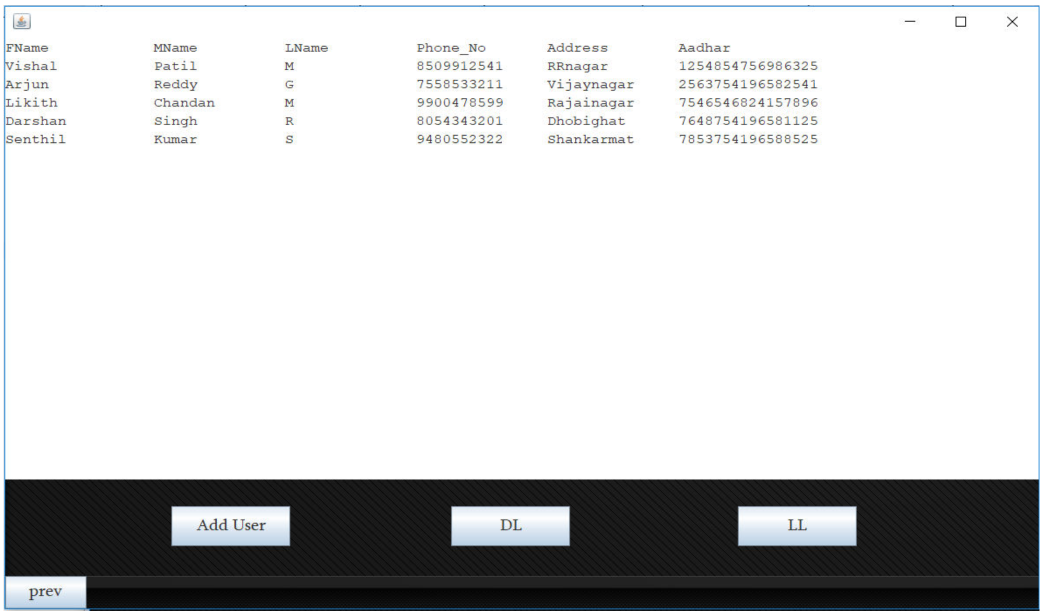


Fig 5.2

Fig 5.3 allows the admin to view the users details



```
FName          MName        LName        Phone_No     Address      Aadhar
Vishal         Patil        M            8509912541   RRnagar      1254854756986325
Arjun          Reddy        G            7558533211   Vijaynagar   2563754196582541
Likith         Chandan      M            9900478599   Rajainagar   7546546824157896
Darshan        Singh        R            8054343201   Dhobighat    7648754196581125
Senthil        Kumar        S            9480552322   Shankarmat   7853754196588525
```

Fig 5.3

Fig 5.4 allows the admin to add users

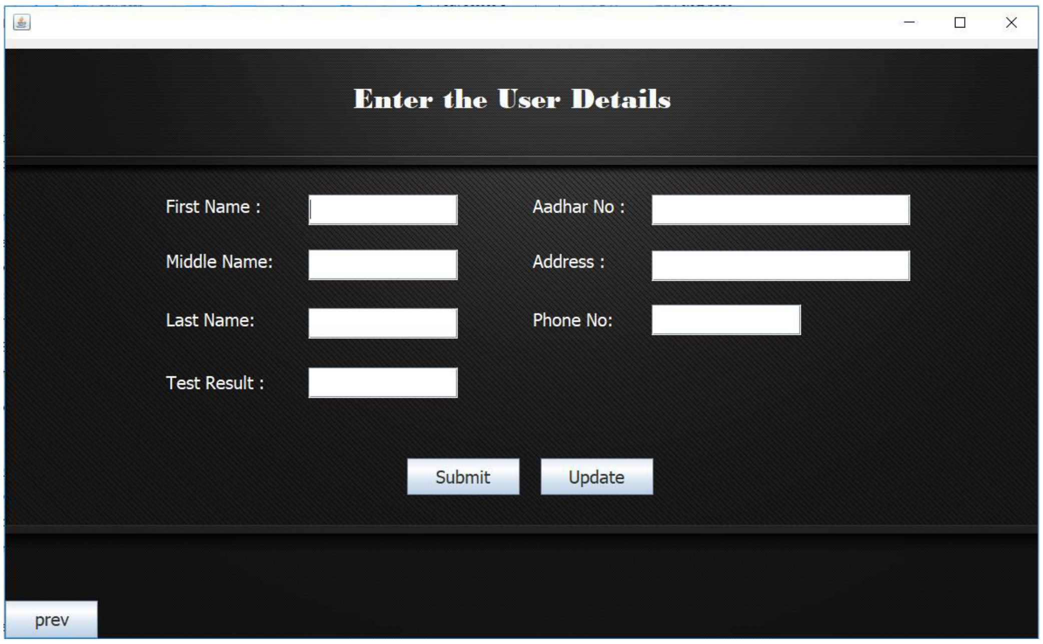
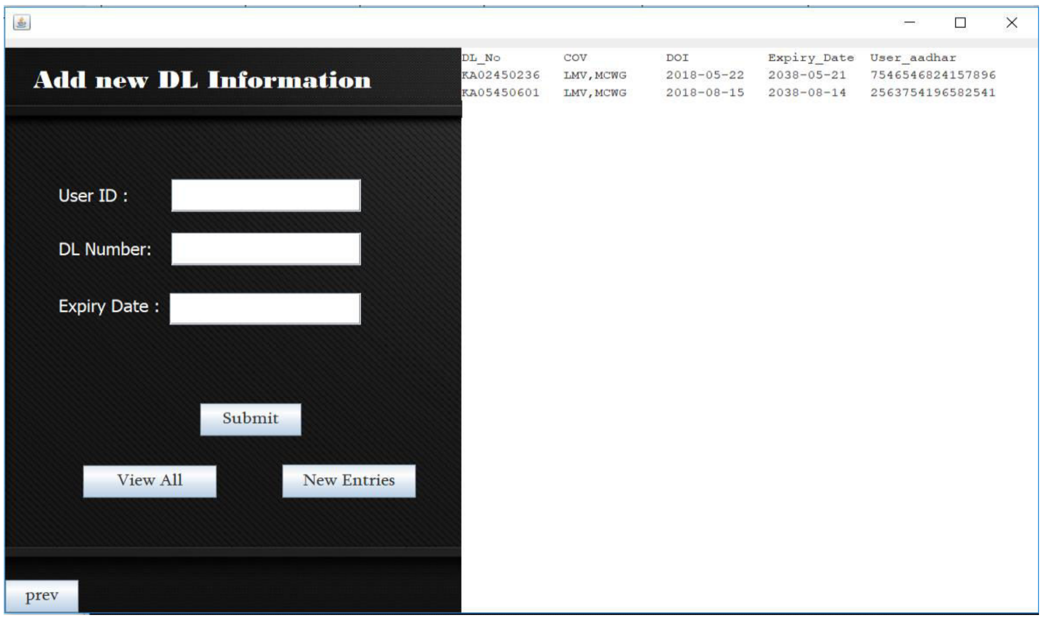


Fig 5.4

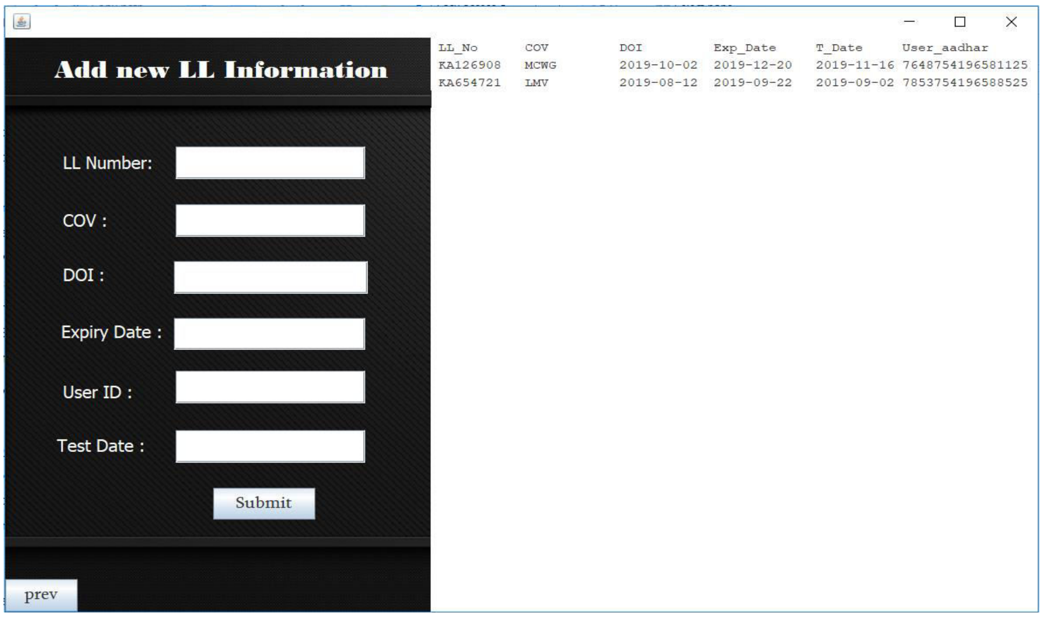
5.5 allows the admin to view and add users DL info (This is where the stored procedure comes into play)



DL_No	COV	DOI	Expiry_Date	User_aadhar
KA02450236	LMV,MCWG	2018-05-22	2038-05-21	7546546824157896
KA05450601	LMV,MCWG	2018-08-15	2038-08-14	2563754196582541

Fig 5.5

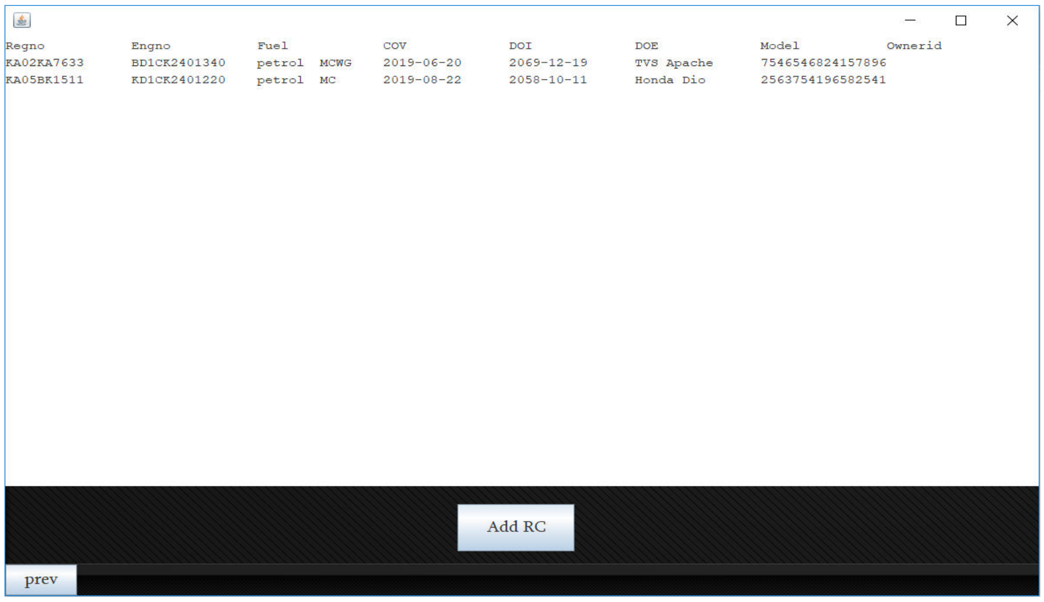
Fig 5.6 allows the admin to view and add LL info



LL_No	COV	DOI	Exp_Date	T_Date	User_aadhar
KA126908	MCWG	2019-10-02	2019-12-20	2019-11-16	7648754196581125
KA654721	LMV	2019-08-12	2019-09-22	2019-09-02	7853754196588525

Fig 5.6

Fig 5.7 allows the admin to view vehicle registrations



```
Regno         Engno          Fuel          COV           DOI           DOE           Model            Ownerid
KA02KA7633    BD1CK2401340   petrol  MCWG  2019-06-20    2069-12-19    TVS Apache    7546546824157896
KA05BK1511    KD1CK2401220   petrol  MC    2019-08-22    2058-10-11    Honda Dio     2563754196582541
```

Fig 5.7

Fig 5.8 shows the admin to add new vehicle registration info

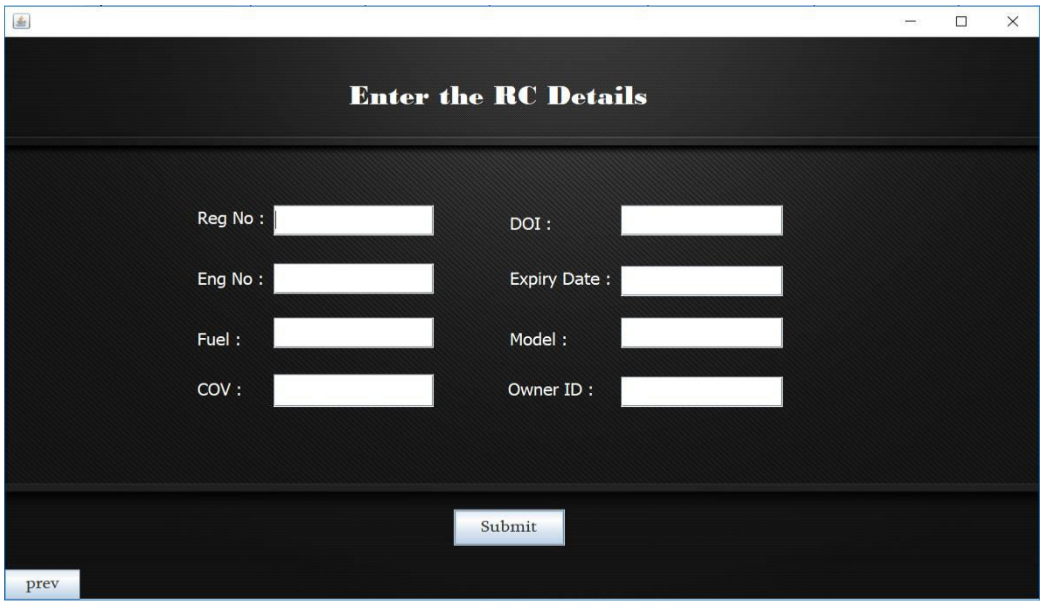
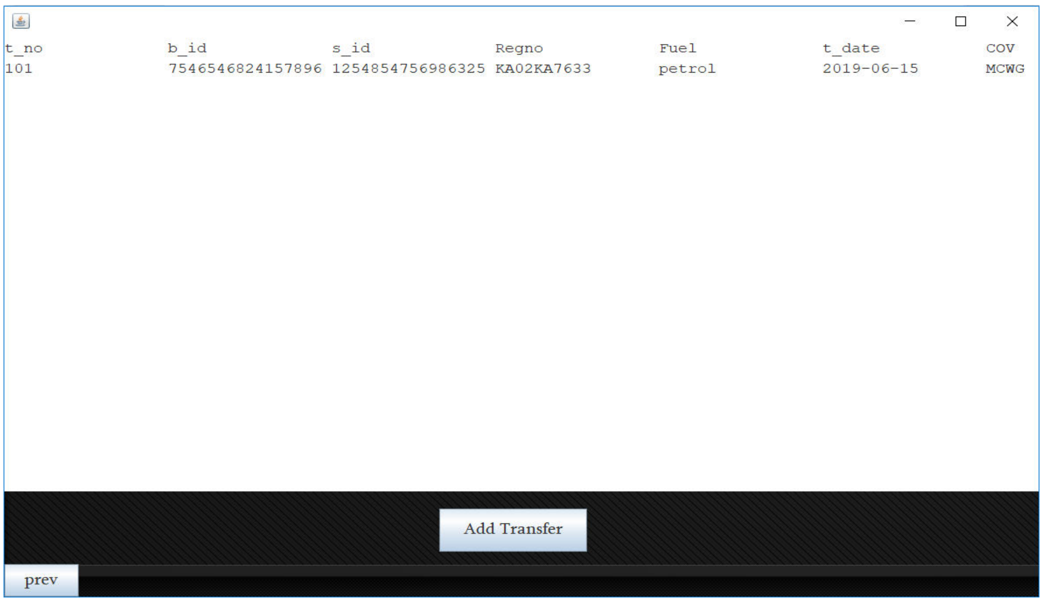


Fig 5.8

Fig 5.9 allows the admin to view vehicle transfers



```
t_no         b_id             s_id             Regno        Fuel         t_date       COV
101          7546546824157896 1254854756986325 KA02KA7633   petrol       2019-06-15   MCWG
```

Fig 5.9

Fig 5.10 allows admin to add new vehicle transfer info

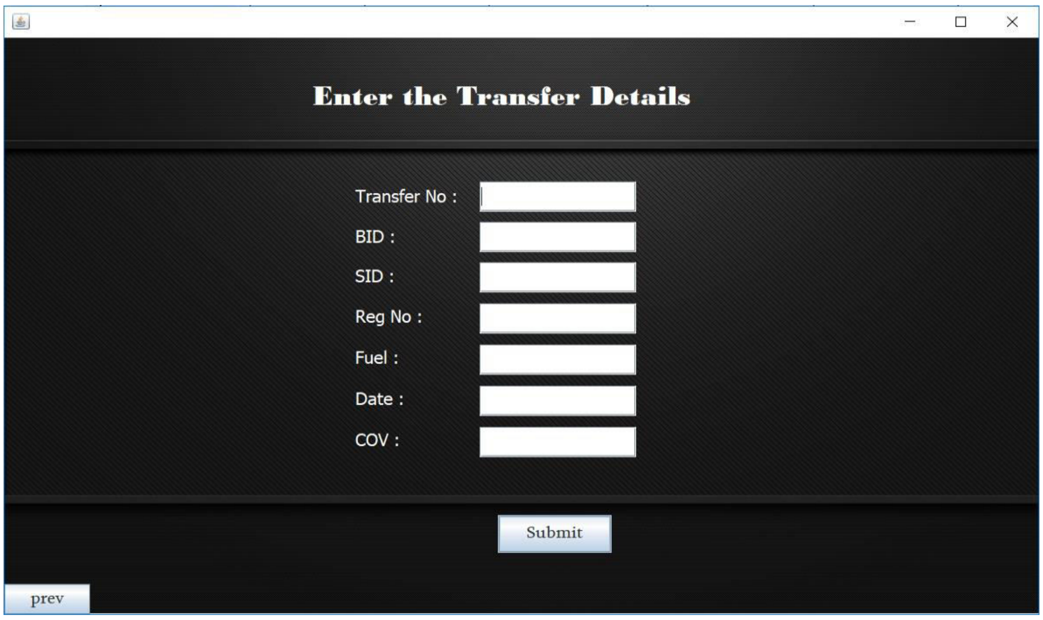


Fig 5.10

CHAPTER 6

CONCLUSION, FUTURE ENHANCEMENTS

This system was mainly designed to reduce the manual work of storing, managing and updating information and also makes it easier for the user. It also provides errorless and flexible reports regarding the DL details, LL details, Transfer details etc. This project was implemented successfully and was satisfied all the problems stated in the problem statement.

There is always a room for improvement in any software package, however good and efficient it may be. But the improvement thing is that the system should be flexible enough for further modifications. Considering this important factor, the system is designed in such a way that provisions can be given for further enhancement without affecting the system presently developed.

BIBLIOGRAPHY

- https://www.w3schools.com/sql/sql_ref_keywords.asp
- https://www.w3schools.com/java/java_ref_keywords.asp
- https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/package-summary.html
- https://docs.oracle.com/javase/10/docs/api/javax/swing/package-summary.html
- Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.