RTO MANAGEMENT SYSTEM

Mini project report

Submitted in partial fulfillment of the requirement for
Advanced Database Management System course of
Third year in computer engineering

By

Roll No.	Name of Student
TE A - 11	Harshal Dalvi
TE A - 14	Pranav Firake
TE A - 15	Aditya Gaikar

Under the guidance of

Prof. Nita S. Patil

Nagar Yuvak Shikshan Sanstha's

Datta Meghe College of Engineering

Department of Computer Engineering

Airoli, Navi Mumbai-400708

(Affiliated to university of Mumbai)

ACADEMIC YEAR 2012-2013

CERTIFICATE

DATTA MEGHE COLLEGE OF ENGINEERING

AIROLI, NAVI MUMBAI

CERTIFICATE

This is to certify that

Roll no.	**Name of student**
TE A - 11	Harshal Dalvi
TE A - 14	Pranav Firake
TE A - 15	Aditya Gaikar

have satisfactorily completed the Mini Project

on

RTO Management System

in partial fulfillment requirement for Advanced Database Management System course of Third year in computer engineering in the academic year 2012-2013

As prescribed by the **University of Mumbai** under the guidance of

Guide
Prof. Nita S. Patil

Guide	Head of Department	Principal
Prof. Nita S. Patil	Prof. A. P. Pande	Dr. S.D.Sawarkar

ACKNOWLEDGEMENT

We would like to express my sincere gratitude towards the Computer Engineering Department of Datta Meghe College of Engineering.

It gives us the pleasure to present this project on “License Management System” and would like to take this opportunity to thank everyone who has been a part of making this project.

Our gratitude and sincere thanks to Principal Dr. S. D. Sawarkar and Prof. A.P. Pande (HoD of Computer Department) for providing all the necessary system facilities and privileges that enabled us to complete the project in time. I would also like to thank Mrs. Nita S. Patil for her guidance, genuine suggestions and constant encouragement, without which this project would have been a success.

Finally I would like to thank my colleagues and friends who helped me in completing the project successfully.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 INTRODUCTION OF THE PROJECT

Project name is “RTO Agency Management System”. An *RTO agency* is basically an easy interface between the people and the government registration to apply for license and to register their vehicles.

This RTO Management System has been developed using Java Swing and My SQL in the Netbeans interface.

This system provides the benefits of streaming of operations, enhanced administration and control, improved response, cost control and improved efficiency and profitability.

The goal of any system development is to develop and implement the system cost effectively; user friendly and most suited to the user analysis is the heart of the processor. Analyze the study of various operations performed by the system and their relationship within and outside of the system. During analysis, data collected on the files, decisions points and transactions handled by the present system. Different kinds of tools are used in analysis of which interviewer is a common one.

RTO Management System is a computerized management system. This system has been developed to form whole management system including Applicants, Vehicle registrations, Records, etc. This system also keeps the record of other assets besides software of this organization. The proposed system will keep a track of Applicants, Vehicle registrations, Records and generation of the report regarding the present status.

1.2 FEASIBILITY STUDY

1.2.1 OBJECTIVE OF THE PROJECT

The project RTO Management System is aimed to help the users apply for license, register their vehicles and to keep a track of their daily records. Also it is essential in handling other aspects. There are following main objectives of the RTO:

- Keep records of daily applications of applicants.
- Keep records of their vehicles registered.
- Give the applicants their License Number.

1.2.2 SCOPE OF THE PROJECT

It can be used in RTO for maintaining the above mentioned details.

It efficiently maintains the details about the applicant. Simultaneously updates changes made to any data, item in the entire data base. It is faster than manual system.

1.3 ORGANIZATION OF THE REPORT

We have divided the report into 5 units. Further, each unit is divided into a number of chapters which aids better understanding. The first unit introduces our topic. Also we have discussed the scope and the objectives of the report.

Second unit enumerates the existing system and proposed system for its betterment. Also we discuss the concepts of requirement collection and analysis.

The third unit is the most important unit where we discuss the overall design of the database. We see the front-end and back-end design in detail which includes the EER diagram, its mapping, modeling and physical database design.

The fourth unit focuses on the implementation part. We have attached the GUI snapshots along with the coding print-out.

The fifth & the final unit look into the validation of the report and its acceptance testing. It also comprises about the future scope, Summary and conclusion. It also sheds light towards the bibliography and references.

2. LITERATURE REVIEW

A literature review is a body of text that aims to review the critical points of current knowledge including substantial findings as well theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and such do not report any new or original experience work.

2.1 EXISTING SYSTEM

The present existing system of RTO management is done manually through registers which is too difficult and lengthy. In order to maintain records, different registers are maintained which is too complicated to use. The jobs done are as follows:

- License record: Searching the personal details of the applicant and his license in the registers is a very tough and time consuming job.
- Vehicle record: In this records of vehicles are fed into the register of the RTO which is centralized and the vehicles are got back to after being properly registered with the RTO.
- Records of accidents and infractions: In this records of different accident records are maintained. The different categories are also maintained.
- It is very difficult and time consuming to maintain all the details in to the registers when the organization is very big.
- Modifying the records is not very easy, since for a minute change the whole record would be required to re-enter. For instance, when the address of any applicant changes the locating that applicant and doing modification in the register is very tedious.
- It also takes much time for doing entry and checking of records.
- There is also possibility of mismatching of entries during entry of records.
- To make and maintain different reports is also very tough and tedious job
- Information is stored into registers, which requires large storage space.
- Searching particular record is very tedious job, since as a time passes the condition of registers become sad.
- Every year, they have to maintain new register.

2.2 PROPOSED SYSTEM

To overcome the drawbacks of the present system and ease the working of such consultancies our project comes into the picture. Our software is multifunctional which includes the details of Applicants, Vehicles and Records.

- The APPLICANT FILE is the one which contains all the details of applicant which are kept for a long time.
- The VEHICLE FILE is the one which contains all the details of vehicles which are registered by different applicants.
- The RECORD FILE is one which contains all the details of the accidents and infractions committed by a license holder.
- This system provides search facility for management of Applicant, Vehicle or Record.
- Backup allows the data to transfer at removable device.
- This system is totally GUI based, hence it is easy to use.

2.3 REQUIREMENT COLLECTION AND ANALYSIS

This module facilitates the capturing of applicant information & records at different entry points, enables queries at different workstations and generates reports at required frequency. The following details are part of this module:

Applicant & vehicle details:

- App ID
- Name
- Date of Birth
- Address
- License Number
- Chassis ID
- City
- Model make
- Confirmation receipt
- Comprehensive Search Engine for monthly analysis

3. DESIGN

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical storage parameters needed to generate a design in Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structure used to store the data. In the relational model these are the tables and views. In an object database the entities and the relationship map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as a part of the overall database application within the database management system (DBMS).

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- Determine the relationship between the different data elements.
- Superimpose a logical structure upon the data on the basis of these relationships.

Database Design is basically divided into two parts:

1. Front end design
2. Back end design

3.1 FRONT END DESIGN

Swing is the primary Java GUI widget toolkit. It is part of Oracle's Java Foundation Classes (JFC) — an API for providing a graphical (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 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 box 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.

The Standard Widget Toolkit (SWT) is a competing toolkit originally developed by IBM and now maintained by the Eclipse community. SWT's implementation has more in common with the heavyweight components of AWT. This confers benefits such as more accurate fidelity with the underlying native windowing toolkit, at the cost of an increased exposure to the native platform in the programming model.

There has been significant debate and speculation about the performance of SWT versus Swing; some hinted that SWT's heavy dependence on JNI would make it slower when the GUI component and Java need to communicate data, but faster at rendering when the data model has been loaded into the GUI, but this has not been confirmed either way. A fairly thorough set of benchmarks in 2005 concluded that neither Swing nor SWT clearly outperformed the other in the general case.

3.2 BACK END DESIGN

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that runs as a server providing multi-user access to a number of databases.

It is named after co-founder Michael Widenius' daughter, My. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

Techworld.com reviewed MySQL Server 5.0 in November 2005 and stated "MySQL is free, supports advanced concepts such as replication, and performs extremely well in the average case. It's always been popular, and version 5.0 fills a few glaring holes in 4.1. The developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good".

community-linuxmint.com reviewed MySQL 5 and stated "mysql is a fast, stable and true multi-user, multi-threaded sql database server. sql (structured query language) is the most popular database query language in the world. the main goals of mysql are speed, robustness and ease of use.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

Micro life-cycle of database:

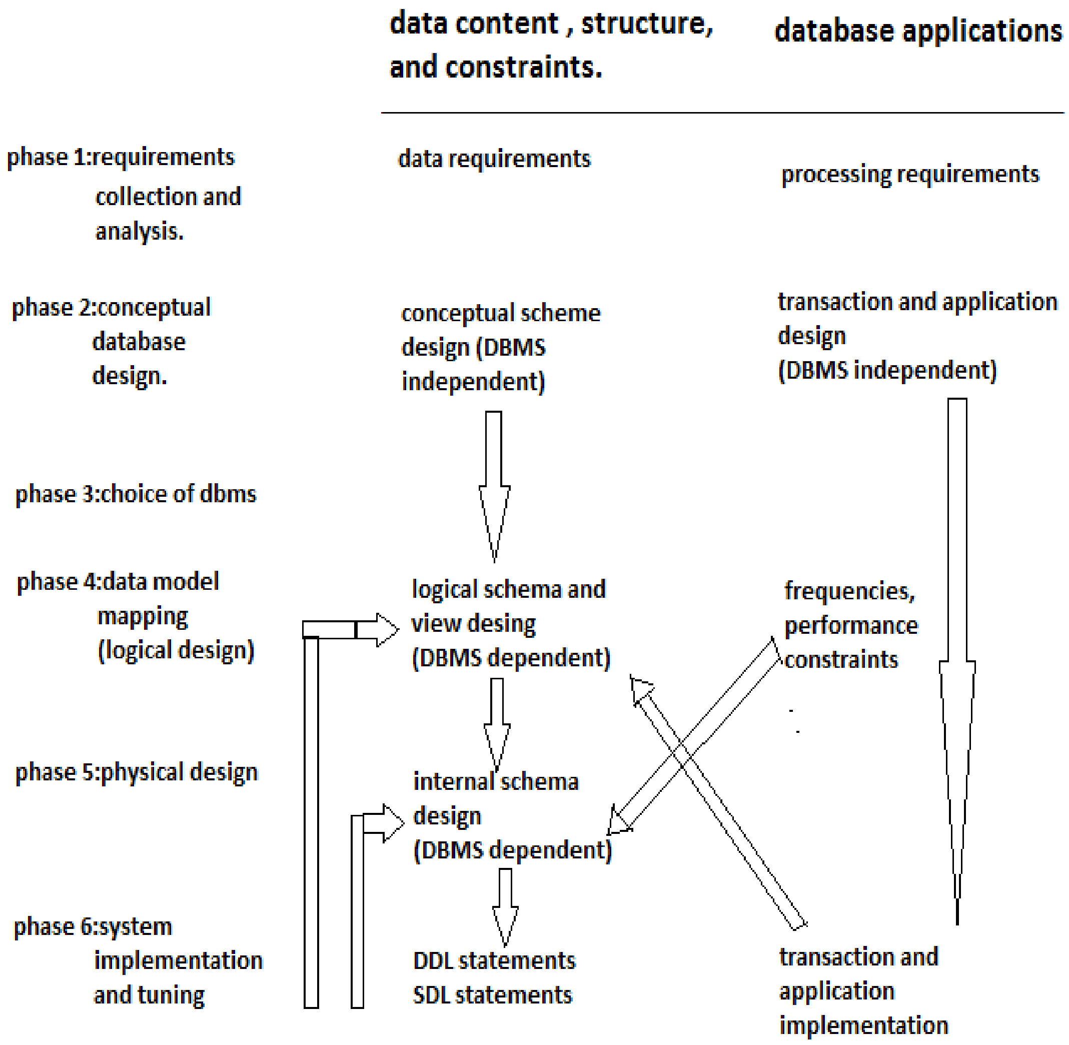


Fig 3.2 Micro Life Cycle

3.2.1 SYSTEM DEFINITION

The purpose of the project entitled as **RTO Management System** to computerize the front office management of the system to develop software which is user friendly, simple, fast and cost effective. It deals with the collection of applicant info (app_id, name, address, dob, etc.), vehicle info (regname, regmodel, regdate, chaid, etc.) and record info (lic_id, noacc, infra, amtfined, etc.) Traditionally it was done manually. The main function of the system is to register and store applicant details and vehicle list details (lic_id, app_no, address, regmodel, etc.) and retrieve this detail as and when required. And also to able to delete these details and among other functions. Vehicles are of 3 types – Two Wheeler, Four Wheeler and transport vehicles. Four Wheeler has fuel type attribute specifying the fuel used(e.g. Petrol , Diesel , CNG etc) Transport vehicle has type attribute specifying purpose vehicle is going to used for (e.g. Passenger or goods purpose).

An applicant is one who wishes to apply for license or vehicle registration. License is granted to applicant after filling the application form and similarly vehicle registration.

3.2.2 CONCEPTUAL DATABASE DESIGN

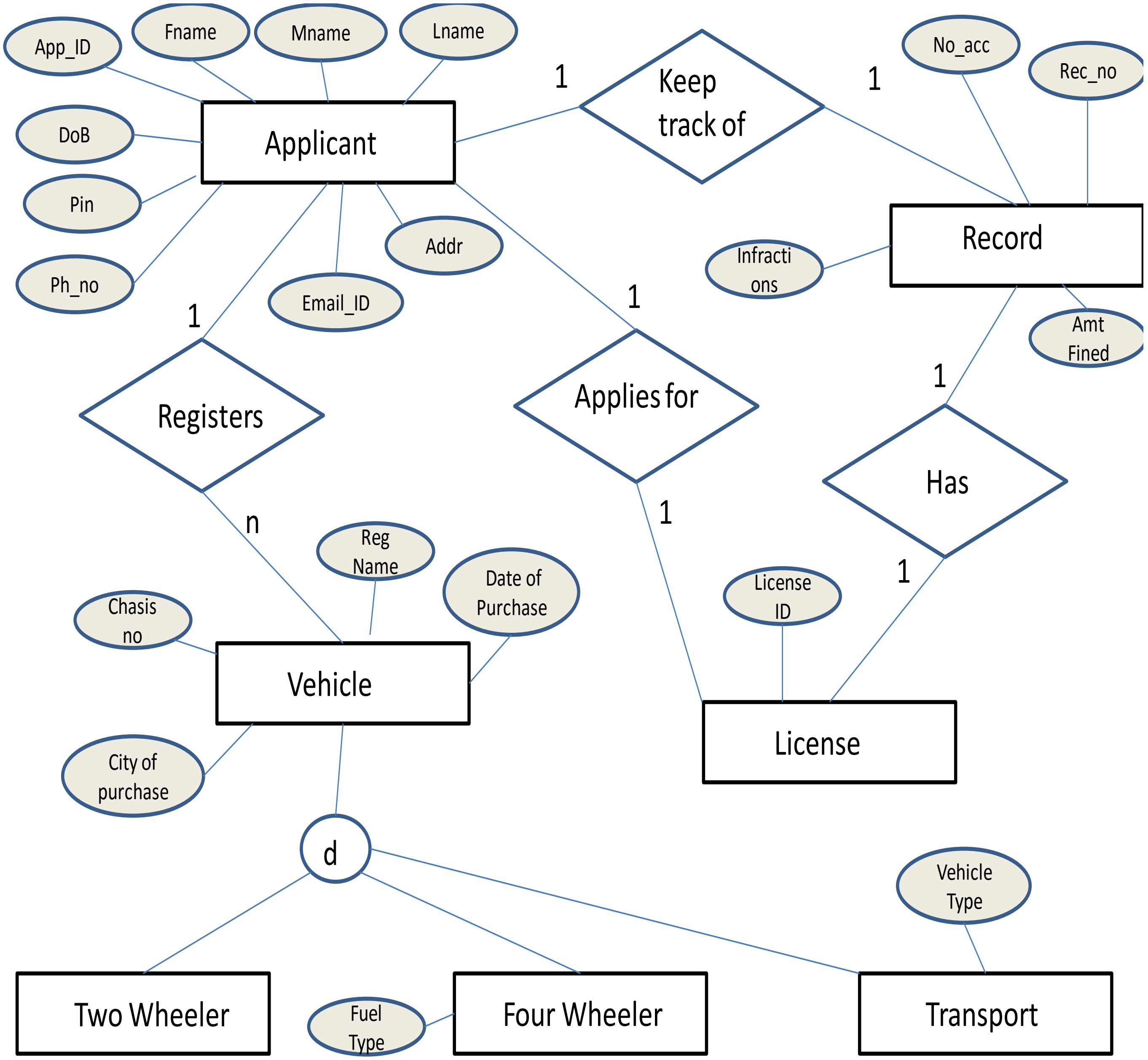


Fig 3.2.2 Conceptual Schema Design

3.2.3. CHOICE OF DBMS

SOFTWARE SPECIFICATIONS

- Operating System : Windows XP/Vista/2007
- Front end : Java Swing (Netbeans 7.0)
- Back end : MYSQL
- Design Tool : Data Flow Diagram

HARDWARE SPECIFICATIONS

- Processor : X86 Compatible processor with 1.7
- GHz Clock speed
- RAM : 512 MB or more
- Hard disk : 20 GB or more
- Monitor : VGA/SVGA
- Keyboard : 104 Keys
- Mouse : 2 buttons/ 3 buttons

3.2.4 DATA MODELING

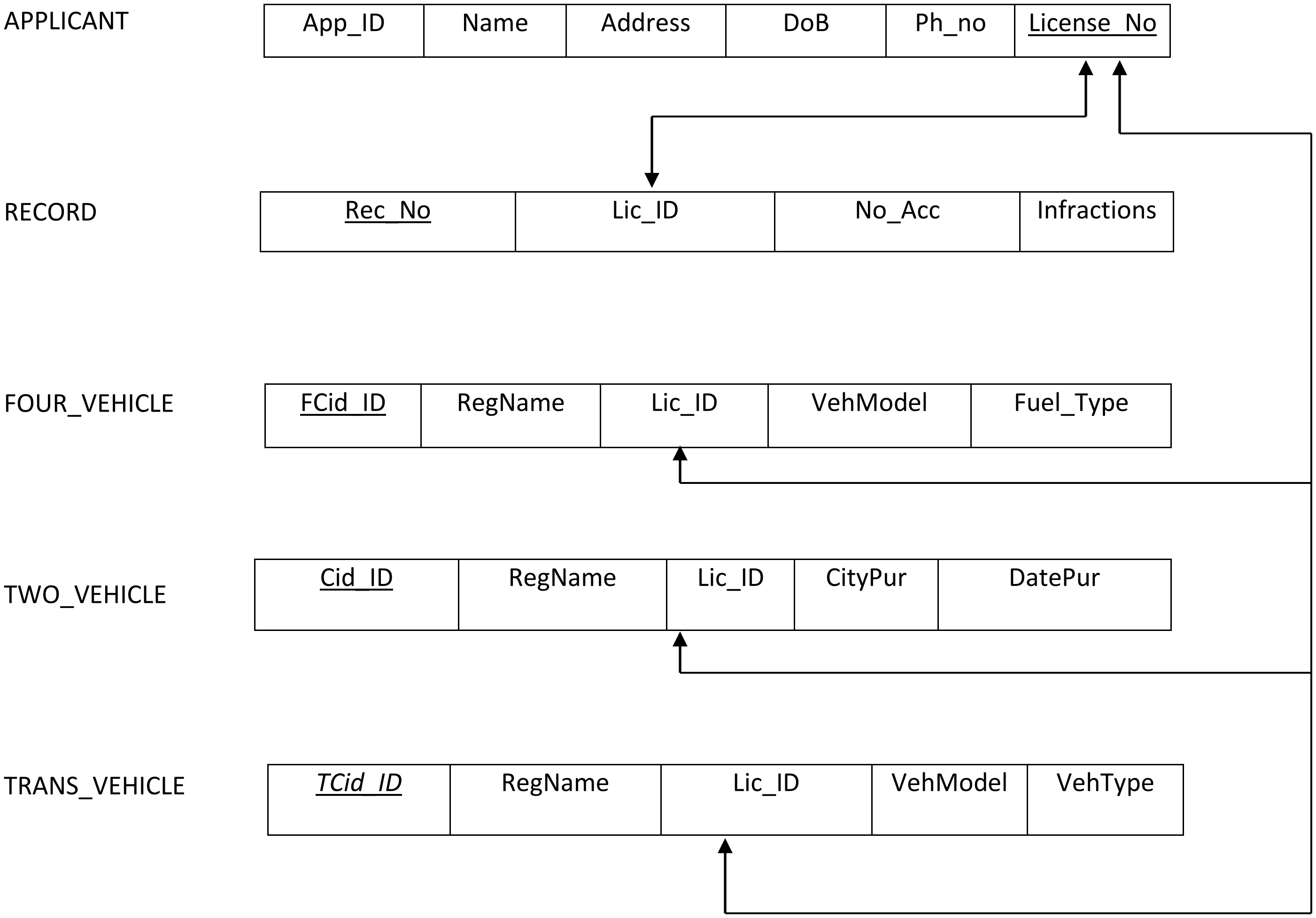


Table 3.2.4 Mapping Tables

3.2.5 PHYSICAL DATABASE DESIGN

This comprises the process of specifying the conceptual, external, internal database definitions, creating empty database files, and implementing the software application. The following tables in their respective database design are used in the project.

Table 3.1 Applicant

Field Name	Data Type	Size	Key
App_ID	Number	Integer	Primary key
Fname	Text	10	Not Null
Mname	Text	10	Not Null
Lname	Text	10	Not Null
Address	Text	30	Not Null
Pin	Text	6	Not Null
DoB	Text	10	Not Null
E_id	Text	20	Not Null
Ph_no	Text	10	Not Null
License_ID	Text	6	Not Null

Table 3.2 Four wheeler Vehicle Registration

Field Name	**Data Type**	**Size**	**Key**
Fc_ID	Text	5	Primary key
FReg_Name	Text	20	Not Null
Flic_ID	Text	6	Not Null
FV-Model	Text	20	Not Null
FDat_Pur	Text	10	Not Null
Fcit_Pur	Text	10	Not Null
F_Type	Text	10	Not Null

Table 3.3 Two Wheeler Vehicle Registration

Field Name	**Data Type**	**Size**	**Key**
Cha_ID	Text	5	Primary key
Reg_Name	Text	20	Not Null
Lic_ID	Text	6	Not Null
V-Model	Text	20	Not Null
Dat_Pur	Text	10	Not Null
Cit_Pur	Text	10	Not Null

Table3.4 Transport Vehicle Registration

Field Name	Data Type	Size	Key
tc_ID	Text	5	Primary key
TReg_Name	Text	20	Not Null
TLic_ID	Text	6	Not Null
TV-Model	Text	20	Not Null
TDat_Pur	Text	10	Not Null
TCit_Pur	Text	10	Not Null
Veh_Type	Text	10	Not Null

Table3.5 Record

Field Name	Data Type	Size	Key
Rec_no	Text	3	Primary key
Lic_no	Text	6	Not Null
No_Acc	Text	2	Not Null
Infra	Text	20	Not Null
Amt_Fined	Text	8	Not Null

4. IMPLEMENTATION

Database Implementation Procedure

Stage 1: Define Scope of the Database Project

- Identify which organizational subdivisions will be served by the database
- Define which functions within these organizations will utilize the database
- Identify which existing and planned applications will be converted to the database system
- Prepare proposal for management and obtain go-ahead

Stage 2: Organize Database Project

- Pick users for design team
- Select database Administrator(DBA)
- Establish regular meeting and periodic management reporting for design team

Stage 3: Select Database Management System Products

- Document requirements in formal proposal requests
- Select DBMS vendor

Stage 4: Develop Initial Implementation Plan and schedule

- Identify files that will be converted
- Identify programs within applications specified
- Estimate programmer hours needed to modify applications programs
- Estimate user clerical support needed to verify data using conversion
- Develop implementation schedule

Stage 5: Design Database

- Complete detailed information requirements
- Identify data requirements

- Determine data structure and complete design specifications
- Review and approve design specifications

Stage 6: Perform Training

- Develop training requirements and training schedule
- Train programmers in the use of DML (Data Manipulation Language)
- Train DBA in DCML (Data Manipulation Control Language) and DDL (Data Definition Language)

Stage 7: Install and Test Database

- Code DMCL, schema and subschema
- Modify representative programs
- Code conversion programs
- Generate the database
- Test and debug
- Review and approved test results

Stage 8: Develop Detailed Conversion Plan

- Make individual programming assignments for each program to be modified and each file to be loaded
- Schedule users to verify and correct file contents
- Schedule computer availability
- Prepare formal written conversion schedule and obtain commitments and from all parties involved
- Approve conversation involved

Stage 9: Convert Existing Applications

- Bring up one application at a time
- Update and regenerate database as required
- Approve revised applications as they are converted
- Begin using database for new applications and programs

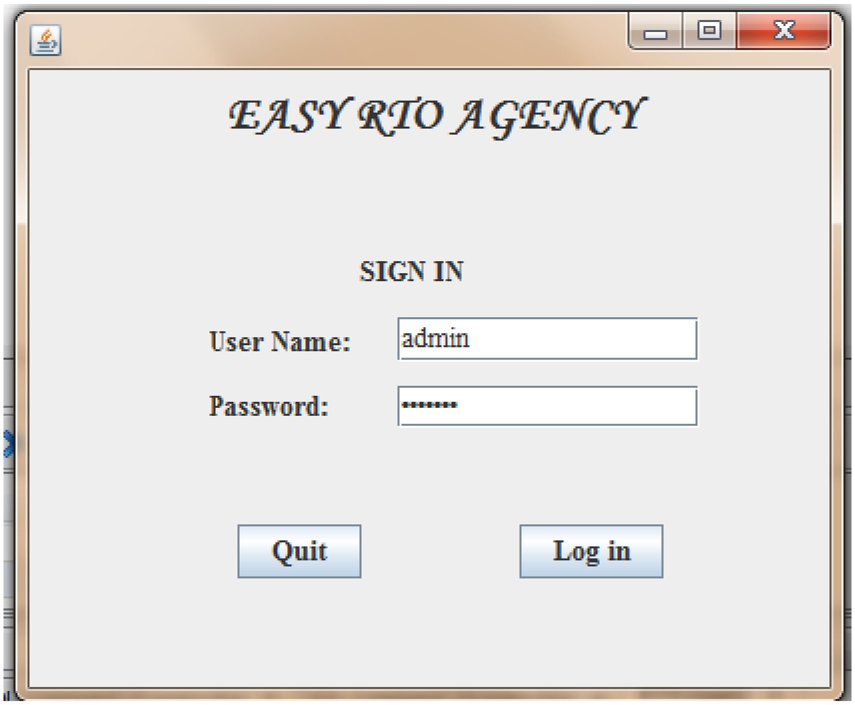
Stage 10: Fine tune database

- Monitor DBMS access statistics and visible performance and modify database as required
- Regenerate database when necessary

Stage 11: Periodically Review database Performance

- Restate organizational goals and information requirement
- Evaluate success of database project
- Begin new projects when required

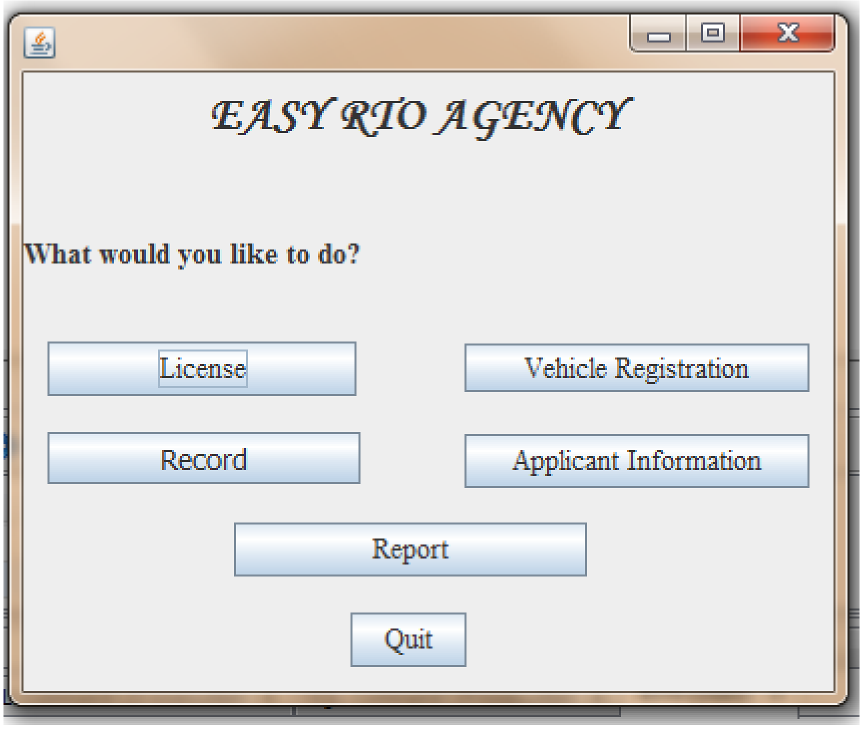
4.1 GUI SNAPSHOTS



User Name: admin

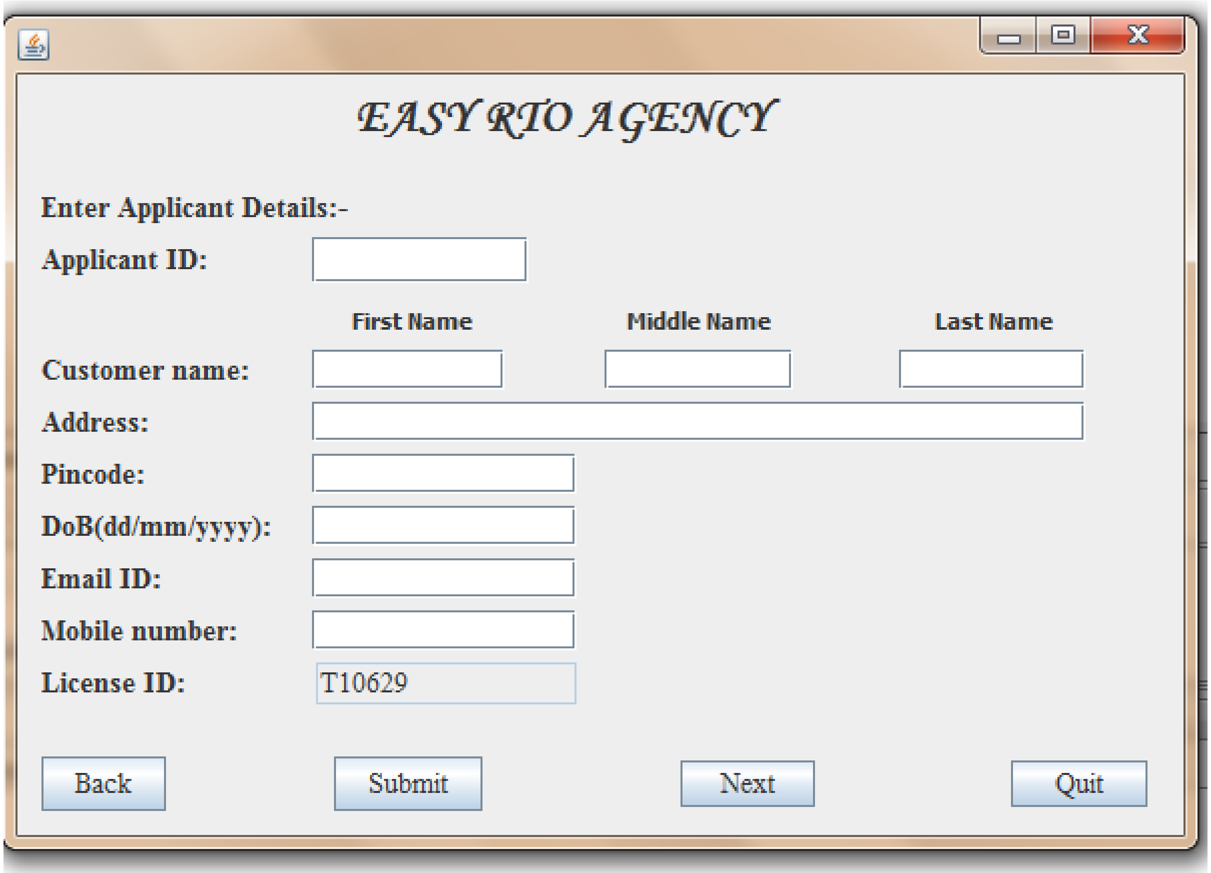
Password: •••••••

Fig 4.1.1 Login Page



What would you like to do?

Fig 4.1.2 Admin Menu



EASY RTO AGENCY

Enter Applicant Details:-

Applicant ID:

	First Name	Middle Name	Last Name
Customer name:			

Address:

Pincode:

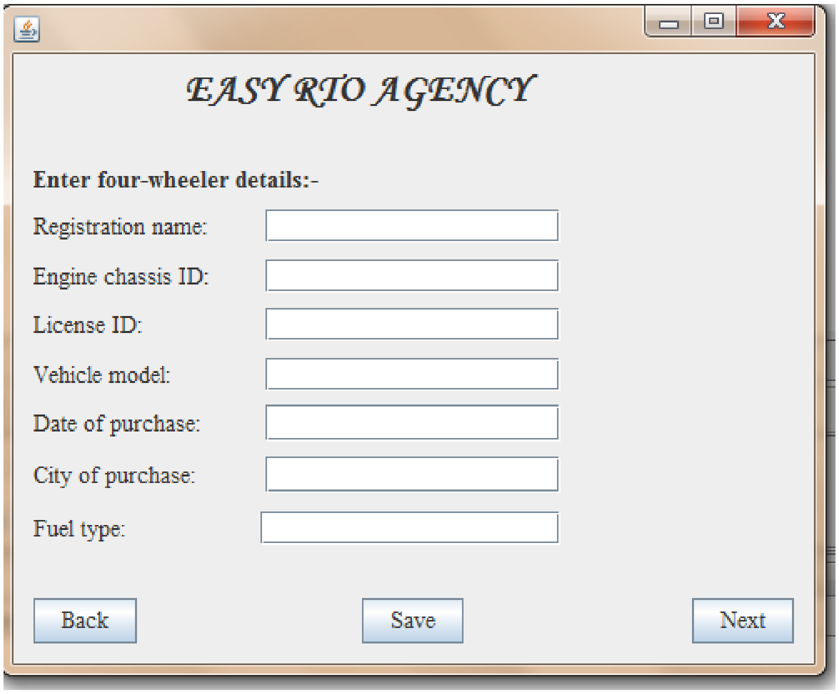
DoB(dd/mm/yyyy):

Email ID:

Mobile number:

License ID: T10629

Fig 4.1.3 Applicant



EASY RTO AGENCY

Enter four-wheeler details:-

Registration name:

Engine chassis ID:

License ID:

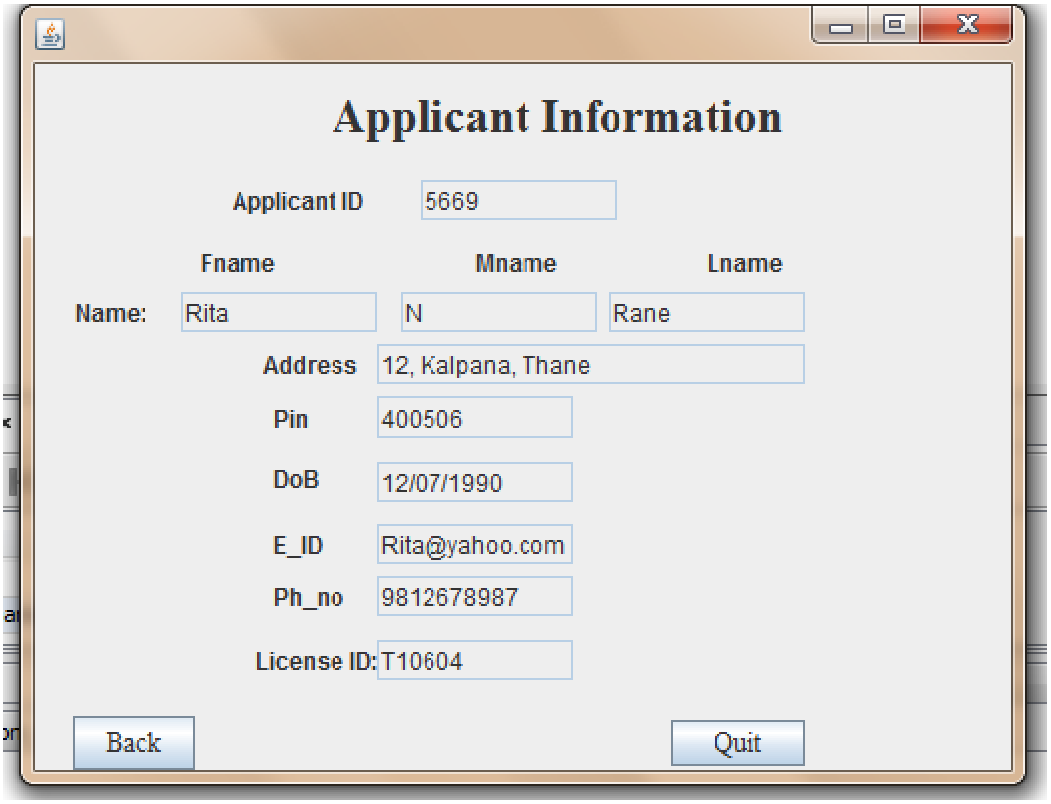
Vehicle model:

Date of purchase:

City of purchase:

Fuel type:

Fig 4.1.4 Vehicle Registration



Applicant Information

Applicant ID 5669

	Fname	Mname	Lname
Name:	Rita	N	Rane

Address 12, Kalpana, Thane

Pin 400506

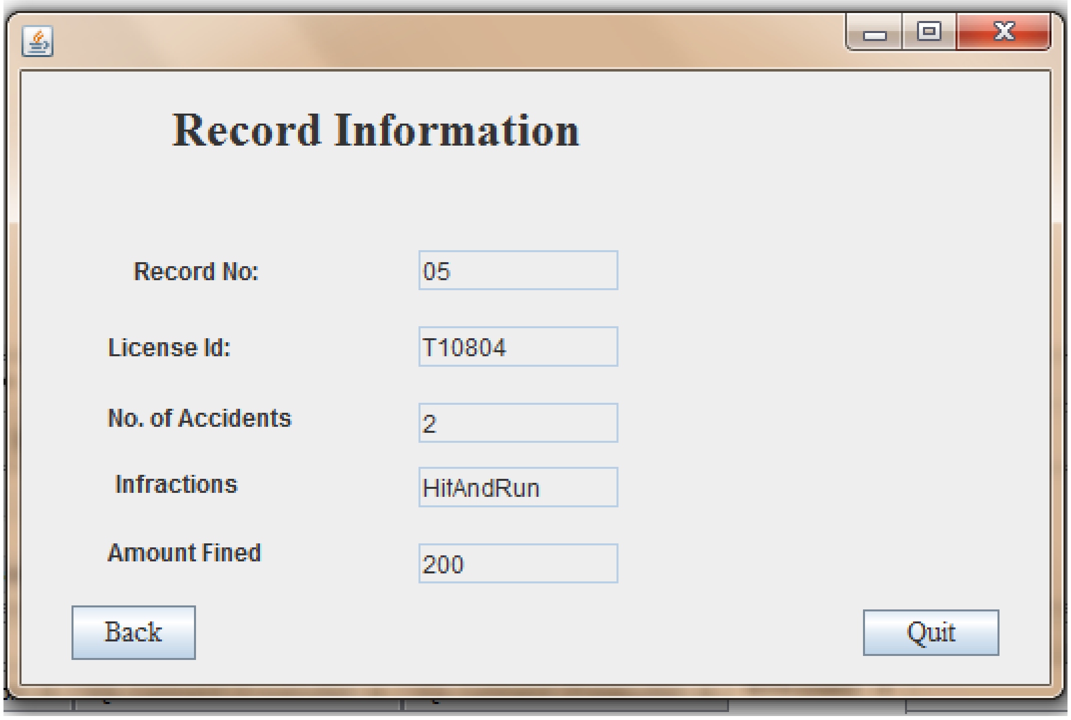
DoB 12/07/1990

E_ID Rita@yahoo.com

Ph_no 9812678987

License ID: T10604

Fig 4.1.5 Applicant Information



Record Information

Record No: 05

License Id: T10804

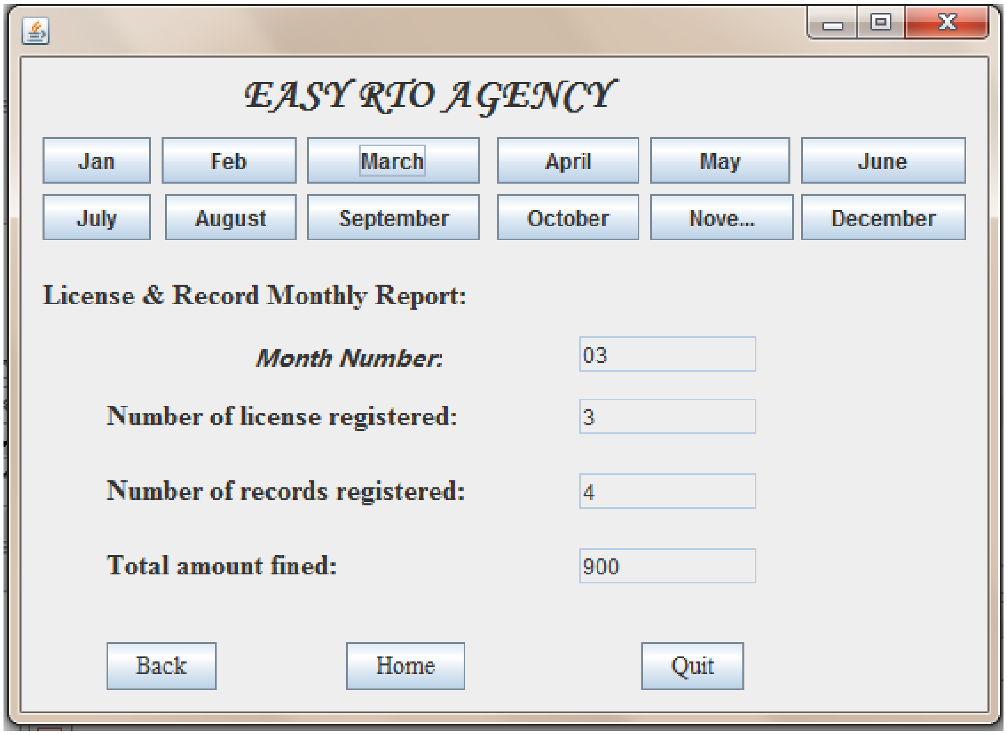
No. of Accidents 2

Infractions HitAndRun

Amount Fined 200

Fig 4.1.6 Record Information

4.2. REPORT GENERATION



License & Record Monthly Report:

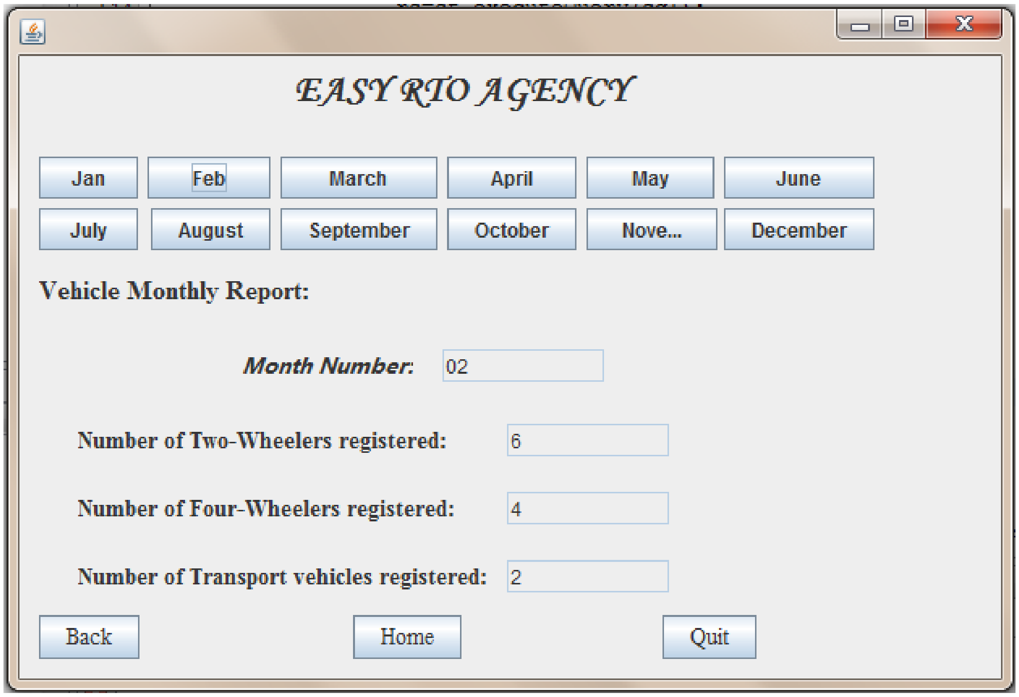
Month Number: 03

Number of license registered: 3

Number of records registered: 4

Total amount fined: 900

Fig 4.2.1 Report 1



Vehicle Monthly Report:

Month Number: 02

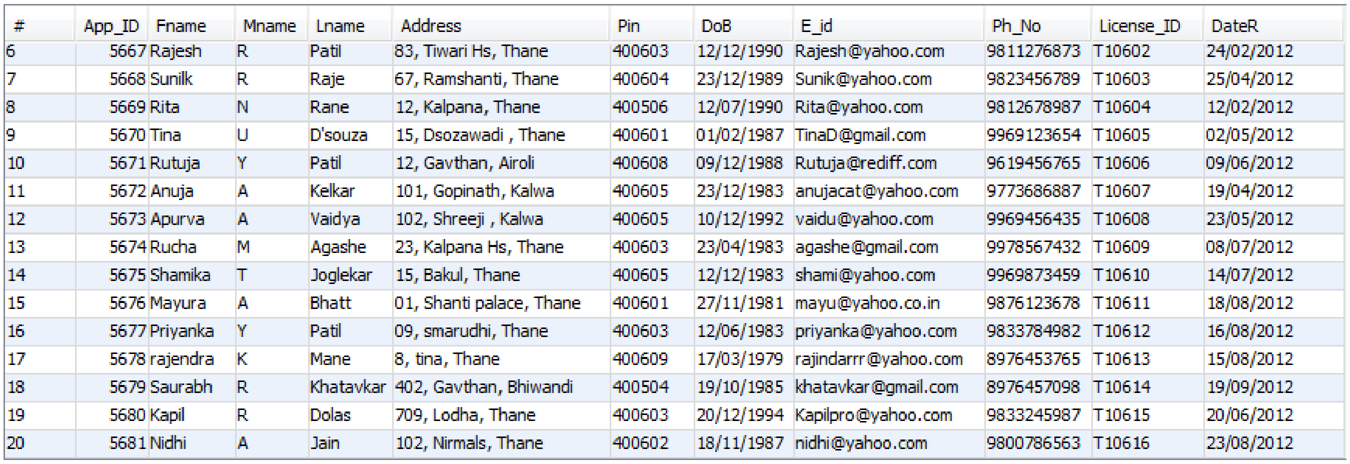
Number of Two-Wheelers registered: 6

Number of Four-Wheelers registered: 4

Number of Transport vehicles registered: 2

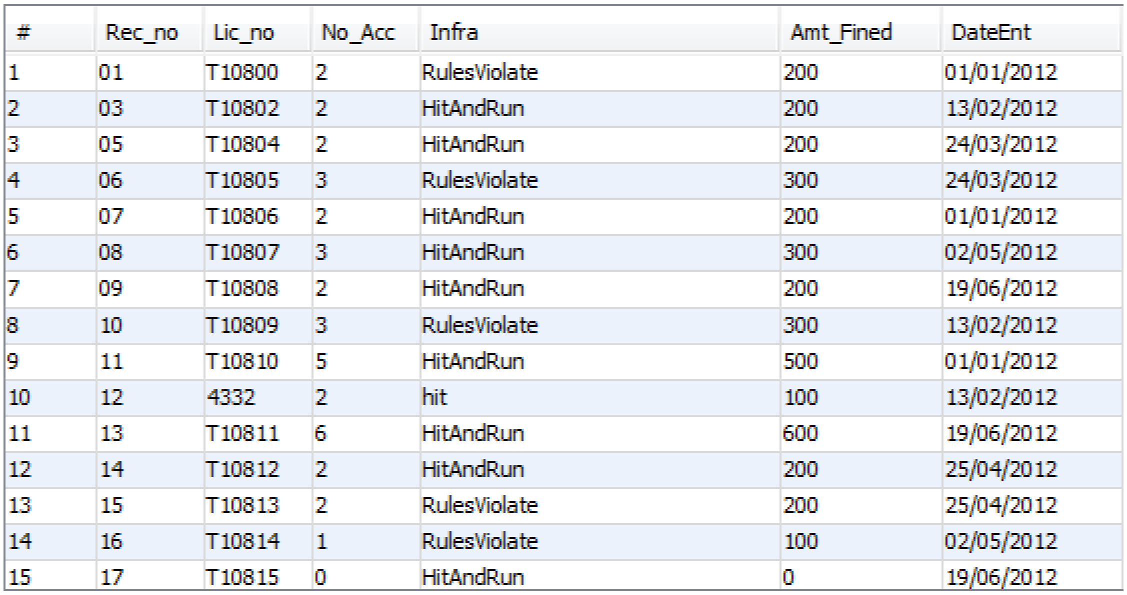
Fig 4.2.2 Report 2

4.3 DATABASE SNAPSHOTS



#	App_ID	Fname	Mname	Lname	Address	Pin	DoB	E_id	Ph_No	License_ID	DateR
6	5667	Rajesh	R	Patil	83, Tiwari Hs, Thane	400603	12/12/1990	Rajesh@yahoo.com	9811276873	T10602	24/02/2012
7	5668	Sunilk	R	Raje	67, Ramshanti, Thane	400604	23/12/1989	Sunik@yahoo.com	9823456789	T10603	25/04/2012
8	5669	Rita	N	Rane	12, Kalpana, Thane	400506	12/07/1990	Rita@yahoo.com	9812678987	T10604	12/02/2012
9	5670	Tina	U	D'souza	15, Dsozawadi , Thane	400601	01/02/1987	TinaD@gmail.com	9969123654	T10605	02/05/2012
10	5671	Rutuja	Y	Patil	12, Gavthan, Airoli	400608	09/12/1988	Rutuja@rediff.com	9619456765	T10606	09/06/2012
11	5672	Anuja	A	Kelkar	101, Gopinath, Kalwa	400605	23/12/1983	anujacat@yahoo.com	9773686887	T10607	19/04/2012
12	5673	Apurva	A	Vaidya	102, Shreeji , Kalwa	400605	10/12/1992	vaidu@yahoo.com	9969456435	T10608	23/05/2012
13	5674	Rucha	M	Agashe	23, Kalpana Hs, Thane	400603	23/04/1983	agashe@gmail.com	9978567432	T10609	08/07/2012
14	5675	Shamika	T	Joglekar	15, Bakul, Thane	400605	12/12/1983	shami@yahoo.com	9969873459	T10610	14/07/2012
15	5676	Mayura	A	Bhatt	01, Shanti palace, Thane	400601	27/11/1981	mayu@yahoo.co.in	9876123678	T10611	18/08/2012
16	5677	Priyanka	Y	Patil	09, smarudhi, Thane	400603	12/06/1983	priyanka@yahoo.com	9833784982	T10612	16/08/2012
17	5678	rajendra	K	Mane	8, tina, Thane	400609	17/03/1979	rajindarrr@yahoo.com	8976453765	T10613	15/08/2012
18	5679	Saurabh	R	Khatavkar	402, Gavthan, Bhiwandi	400504	19/10/1985	khatavkar@gmail.com	8976457098	T10614	19/09/2012
19	5680	Kapil	R	Dolas	709, Lodha, Thane	400603	20/12/1994	Kapilpro@yahoo.com	9833245987	T10615	20/06/2012
20	5681	Nidhi	A	Jain	102, Nirmals, Thane	400602	18/11/1987	nidhi@yahoo.com	9800786563	T10616	23/08/2012

Fig 4.3.1 Applicant Table



#	Rec_no	Lic_no	No_Acc	Infra	Amt_Fined	DateEnt
1	01	T10800	2	RulesViolate	200	01/01/2012
2	03	T10802	2	HitAndRun	200	13/02/2012
3	05	T10804	2	HitAndRun	200	24/03/2012
4	06	T10805	3	RulesViolate	300	24/03/2012
5	07	T10806	2	HitAndRun	200	01/01/2012
6	08	T10807	3	HitAndRun	300	02/05/2012
7	09	T10808	2	HitAndRun	200	19/06/2012
8	10	T10809	3	RulesViolate	300	13/02/2012
9	11	T10810	5	HitAndRun	500	01/01/2012
10	12	4332	2	hit	100	13/02/2012
11	13	T10811	6	HitAndRun	600	19/06/2012
12	14	T10812	2	HitAndRun	200	25/04/2012
13	15	T10813	2	RulesViolate	200	25/04/2012
14	16	T10814	1	RulesViolate	100	02/05/2012
15	17	T10815	0	HitAndRun	0	19/06/2012

Fig 4.1.1 Record Table

4.4 CODING

Insertion code:

```
private void submitActionPerformed(java.awt.event.ActionEvent evt) {
 try {
          String App_ID=app_ID.getText();
          String Fname =fnm.getText();
          String Mname =mnm.getText();
          String Lname =lnm.getText();
          String Address = addr.getText();
          String Pin = pin.getText();
          String DoB =dob.getText();
          String E_id=eid.getText();
          String Ph_No=phno.getText();


if(fnm.getText().toString().equals("")||mnm.getText().toString().equals("")||lnm.getText().toS
tring().equals("")||addr.getText().toString().equals("")||dob.getText().toString().equals("")||ph
no.getText().toString().equals("")||pin.getText().toString().equals("")||eid.getText().toString().
equals(""))
        {
           JOptionPane.showMessageDialog(rootPane, "No textfield can be left blank", "Blank
textfield", JOptionPane.ERROR_MESSAGE);
           return;
        }

        boolean valid=validateEmail(E_id);

      if(!valid)
      {
         JOptionPane.showMessageDialog(rootPane, "Invalid email", "Invalid",
JOptionPane.ERROR_MESSAGE);
         return;
      }

       if(Pin.length()<6 || Pin.length()>6 )
      {
        JOptionPane.showMessageDialog(rootPane, "Pincode lengh invalid", "Error",
JOptionPane.ERROR_MESSAGE);
        return;
 String sql="Insert into applicant
values("+App_ID+",'"+Fname+"','"+Mname+"','"+Lname+"','"+Address+"','"+Pin+"','"+DoB
+"','"+E_id+"','"+Ph_No+"','"+L_id+"')";
       Connection conn=Welcome_frame.createConnection();
       Statement st;
       st = conn.createStatement();
       st.executeUpdate(sql);
       JOptionPane.showMessageDialog(rootPane, "Details registered", "Success",
JOptionPane.PLAIN_MESSAGE);




       flag=true;
            app_ID.setText("");
            fnm.setText("");
            mnm.setText("");
            lnm.setText("");
            addr.setText("");
            pin.setText("");
            dob.setText("");
            eid.setText("");
            phno.setText("");

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

                    // TODO add your handling code here:
    }
```

5. VALIDATION AND ACCEPTANCE TESTING

VALIDATION

User id and password is used for validating the user. User id and password as individual fields

And as a combination are checked against the database fields. On account of wrong login credentials, user gets an appropriate message.

We validate all the fields of form i.e. in the text field of name only text is allowed, similarly for number and date. We also validate search option on every form if information about particular user is not available then appropriate message should be displayed.

Project will provide proper validation for monthly data report of sale and purchase of products.

RTO MANAGEMENT SYSTEM offers all the features described in the Problem Definitions. Thus validation of RTO MANAGEMENT SYSTEM was successfully.

TESTING

The following type of testing was done on this project:

1. Connection to Database

a. A. STATUS :OK

2. Data Retrieval:

a. Connection to database was established successfully, data can be retrieved accurately.

b. STATUS: Integrity constraints imposed and code tweaked to solve the problem.

The project has been tested for various parameters such that it does not generate

Any errors or exceptions and has been accepted and can be used successfully.

6. CONCLUSION AND FUTURE WORK

1. RTO Management System helps in maintaining a totally secured database of applicant and vehicle information. This information can be available at your fingertips.
2. RTO Management System helps in improved user interface between the people and the RTO so that the people can easily apply for the license and vehicle registration.
3. RTO Management System helps in improving efficiency, both on the cost. This is achieved by avoiding duplications, repetitions, delays, missing records and confusions.
4. RTO Management System helps to force orderliness and standardization of the applicant records and procedures in the agency and increasing accuracy & completeness of accident records of people.
5. RTO Management System helps as a good managerial tool to provide total, cost-effective and enhanced functions.
6. RTO Management System helps in gathering information to meet management challenges.

7. BIBLIOGRAPHY AND REFERENCES

- Advanced Database Management System by Korth

- Software Engineering (A Practioner's Approach) Roger S Pressman, Fifth Edition, Tata Mc Graw-Hill.

- System Analysis & Design Method Sixth Edition Whitten, Bentley, Elittman Tata Mc Graw-Hill.

- System Analysis & Design Fourth Edition Igor Hawryszkiewycz.

- Advanced Database Management System by Navathe

<u>WEBSITES:</u>

- www.google.com

- www.answers.com

- http://javabooks4u.blogspot.in

- www.youtube.com