CHAPTER 1
INTRODUCTION

The Regional Transport Office or Regional Transport Authority (RTO / RTA) is the organisation of the Indian government responsible for maintaining a database of drivers and a database of vehicles for various states of India.

Some functions of Road Transport Office are listed below:

- Registration of vehicles.
- Issue of Driving and Conductor licenses.
- Issue of Permits for goods and passenger vehicles.
- Collection of motor vehicle taxes.
- Enforcement of the laws relating to motor vehicles.
- Road Safety measures.
- Pollution control & regulation related to motor vehicles.

Computerizing any function in an organization results in reduced operation costs, increased efficiency and minimized errors. Repetitive entry of input is avoided. The laborious process of repetitive computations involving checking, the computer, thus taking away the drudgery of what is otherwise a monotonous work, does additions and compilation. User-friendly application software helps generate any type of report at the flick of a key.

Software gives the client an easy and efficient way to manage the accounts of an organization. Software always reduces the workload of user by avoiding the following:

- Bundles of paper work.
- Entries to balance sheet manually.
- Chances of mistake.
- Difficulties to maintain the system.

CHAPTER 2

AIM & OBJECTIVES

RTO Management System is being developed to be deployed for day-to-day usage in the district Regional Transport Offices. To reduce problem listed above and to carry task of managing the database, this system is being developed.

2.1 AIM OF THE PROJECT

In today's technology everyone wants to feel comfortable and easy, to achieve this want to make use of it. For this, one major step is to replace all the paper work by computer.

This project maintains the database system of RTO. It keeps record of all the new entries, updating and other operation in good and efficient manner. It integrates all the different functions and operations of data management system. No manual scripting is required for generating various registers. This will fasten services.

2.2 OBJECTIVES OF THE SYSTEM

The following are the objectives of RTO Management System:

- Create new license
- Search existing license
- Create new registration card
- Search existing registration card
- Search city by city code

CHAPTER 3

EXISTING SYSTEM

The Existing system of RTO services has been is usage for two years. The existing system is not giving accurate results while doing transactions. It doesn't provide security; anyone enter into the system and can do their own transactions. It is not flexible in generating reports. And many manual processes are made computerized.

The present system has following drawbacks:

- It is not efficient in performing office work in RTO services.
- It includes much manual process and time consuming.
- It is not user friendly.
- Maintains local data base.
- It is not Generating Accurate Reports.

CHAPTER 4

PROBLEM STATEMENT

The software has to be developed for RTO Management System. Transportation and vehicles have become more and more popular to meet the demands of increased population. RTO system equipped with appropriate registration and pollution management will help in achieving easy, efficient, précised system. The system should be standalone in nature. It should be covering all the areas of the RTO management system like registration details, license and other services. Keeping the track of physical stock manually may contain inaccuracy and inconsistency, delay in processing and retrieval of information and difficulties in identifying the physical stocks location. The purpose of RTO management is to ensure availability of data in a sufficient quality and quantity as and when required and also minimize investments in inventories. It is the necessary to prevention of leakage and illegal data manipulation to make the system easily managed and can be secured. This software should be used to show all the details of the vehicles and citizen licensee holder.

CHAPTER 5
METHODOLOGY

Often, a customer defines a set of general objectives for software, but doesn't identify detailed input, processing or output requirement. In other cases, the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human-machine interaction should take. In these, and many other situations, a prototyping paradigm may offer the best approach.

Although prototyping can be used as a standalone process model, it is more commonly used as a technique that can be implemented within the context of any one of the process models. Regardless of the manner in which it is applied, the prototyping paradigm assists the software engineer and the customer to better understand what is to be built when requirement is fuzzy.

The prototyping paradigm begins with the communication. The software engineer and a customer meet and define the overall objectives for the software, identify whatever requirement are known, and outline areas where further definition is mandatory. Prototyping iteration is planned quickly and modelling (in the form of a "Quick Design") occurs. The quick design focuses on a representation of those aspect of the software that will be visible to the customer/ end-user. The quick design leads to the construction of prototype. The prototype is deployed and then evaluated by the customer/user. Feedback is used to refine requirements for the software. Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understands what needs to be done.

Ideally, the prototype server as a mechanism for identifying software requirement. Is a working prototype is built, the developer attempts to make us of existing program fragments or applied tools (e.g. report generators window managers etc.) that enable working programs to be generated quickly.

It is true that both customer and developers like the prototyping paradigm. User get a feel for the actual system, and developer get to build something immediately. Yet, prototyping can be problematic for the following reasons:

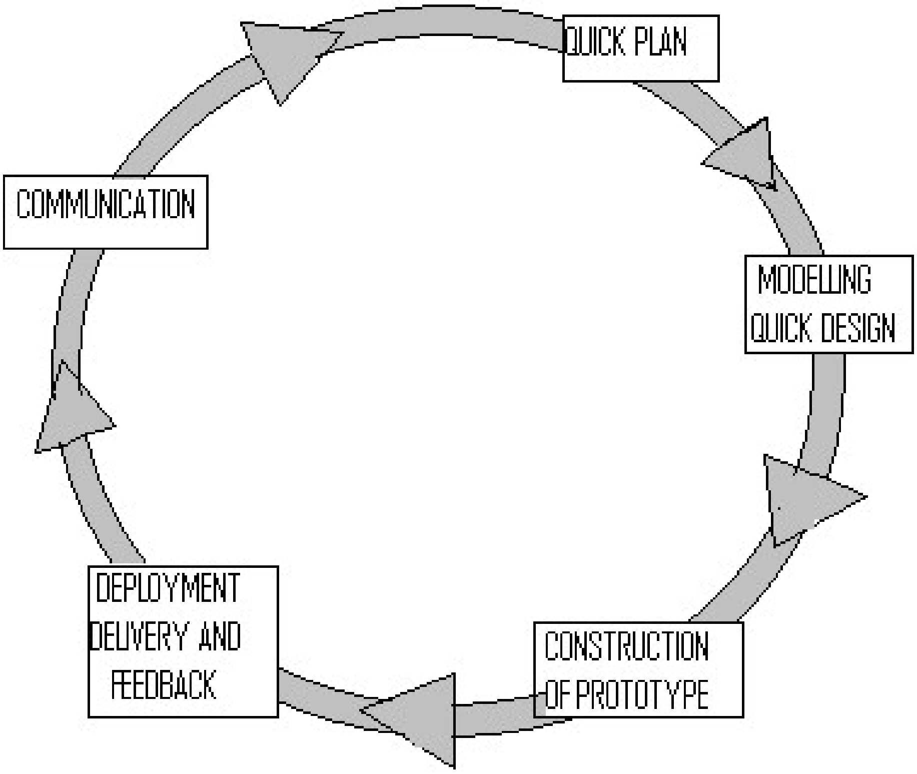


Figure 5.1: The Prototyping Model

- The customers see what appears to be a working version of the software, unaware that the prototype is held together “with chewing gum and baling wire”, unaware that in the rush to get it working we haven’t considered overall software quality or long term and maintainability. When informed that the product must be rebuilt so that high-level of quality can be maintain, the customer cries foul and demands that “a few fixes” be applied to make the prototype a working product. Too often, software development management relents.
- The developer often makes implementation compromises in order to get a prototype working quickly. An inappropriate operating system or programming language may be used simply because it is available and known; an ineffective algorithm may be implemented simply to demonstrate capability. After a time, the developer may become comfortable with these choices and forget all the reasons why they work inappropriate. The less-then-ideal choice has now become an integral part of the system.

Although problem can occur, prototyping can be an effective paradigm for software engineering. The key is to define the rules of the games at the beginning; that is, the customer and developer must both agree that the prototype is built to serve as a mechanism for defining requirements. It is then discarded (at least in the part), and the actual software is engineered with an eye toward quality.

CHAPTER 6

DETAILS OF HARDWARE & SOFTWARE

The system on which the software will be installed should have Java Runtime Environment Package installed on it. The minimum hardware requirements and software requirements are as stated below.

6.1 Minimum Hardware Requirement

Processor	: Pentium Series
Hard Disk	: 10 GB
RAM	: 256 MB

6.2 Software Requirement

Front End	: Java
Back End	: MySQL
Operating System	: Windows Based OS

6.2.1 Java

Java was developed at Sun Microsystems. Work on Java originally began with the goal of creating a platform-independent language and operating system for consumers.

Java has been preferred because:

- Getting started quickly: Although Java is a powerful object-oriented language, it’s easy to learn, especially for programmers already familiar with C or C++.
- Write less and better code: Comparisons of program metrics suggest that a program written in Java can be four times smaller than the same program in C++. The Java language encourages good coding practices, and its garbage collection helps you avoid memory leaks.
- Develop programs faster: The development time may be half as compared to the time in writing the same program in C++.

- Avoid platform dependencies with 100% Pure Java: You can keep your program portable by avoiding the use of libraries written in other languages.

Java has the following features:

- Simple
- Portable
- Object-oriented
- Robust
- Architecture-neutral
- Interpreted
- High performance

6.2.2 SQL

SQL is ANSI standard computer language for Relational Database Management Systems such as MySQL, Access, Sybase, Microsoft SQL Server, Oracle, Ingres etc. It is an industry standard language for creating, updating and querying relational database management systems using simple commands for different operations.

SQL, which talks to a relational database is a language about manipulating sets of data stored in tables. It consists of relatively small number of main commands such Create, Alter, Drop, Select, Insert, Update…etc to accomplish this. SQL has a special version named PL/SQL, which is sometimes termed as a superset of SQL. It bridges the gap between database technology and procedural languages as programmers can use PL/SQL to create programs for validation and manipulation of table, something that was not possible with SQL.

While SQL, also called Interactive SQL, is used to create maintain tables, indexes etc. and also for data manipulation interactively, PL/SQL is used to create programs for validation and manipulation of table data. PL/SQL is thus a language closely related to SQL but allows you to write programs as ordered series of statements.

CHAPTER 7

PROPOSED SYSTEM

Design is multiple-step process the focuses on data structure, software, architecture, procedural details, algorithms etc. and interface between presentation of software that can be accessed for quality before coding begins.

7.1 PROJECT DESIGN

Computer software design changes continuously as new methods: better analysis and border understanding evolved. Software Design is methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However, techniques for software designed do exit, criteria for design qualities are available and design notation can be applied.

Once software requirements have been analysed and specified, software design is the first of three technical activities:

- Design
- Code
- Test that are required to build are verify software.

Each activity transforms information in a manner that ultimately results in validation of the computer software.

The importance of software design can be started with a single quality. Design is the place where quality fostered in software development. Design provides us with the representation of software that accessed for quality.

Design is the only way that we can accurately translate a customer’s requirements into a finished software product or system. Without design, risk of building an unstable system exists – one that will fail when changes are made; one that may difficult to test.

7.2 ARCHITECTURAL DESIGN

Design is the challenging phase of any system. The design phase focuses on the detailed implementation of the system. This phase provides the relevant information and define it how it will be done.

Design is usually defined as following two activities:

1. Architecture
2. Detailed Design

Architecture design is concerned with recognizing the overall system structure, the interfaces and interconnection among program pieces and their precise definitions.

Detailed design is more concerned with the definition and selection of algorithm and data structure that are necessary to fulfil specific software functions.

7.2.1 Architecture Context Diagram

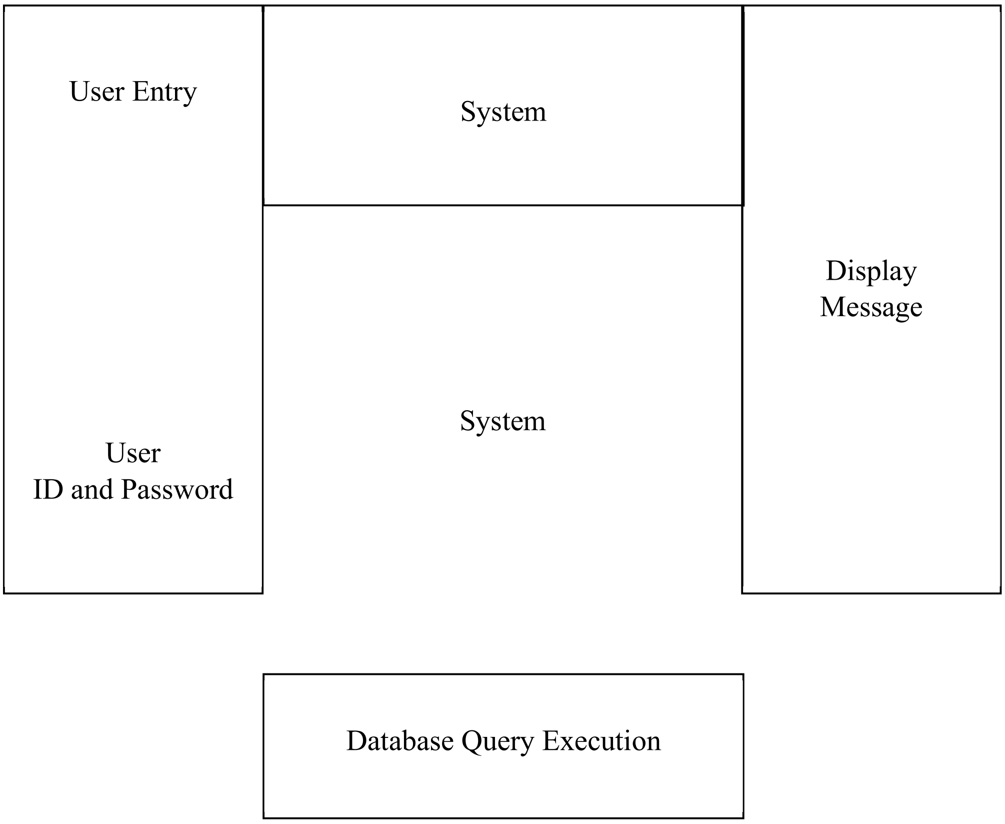


Figure 7.1: Architectural Context Diagram

7.2.2 Control Hierarchy

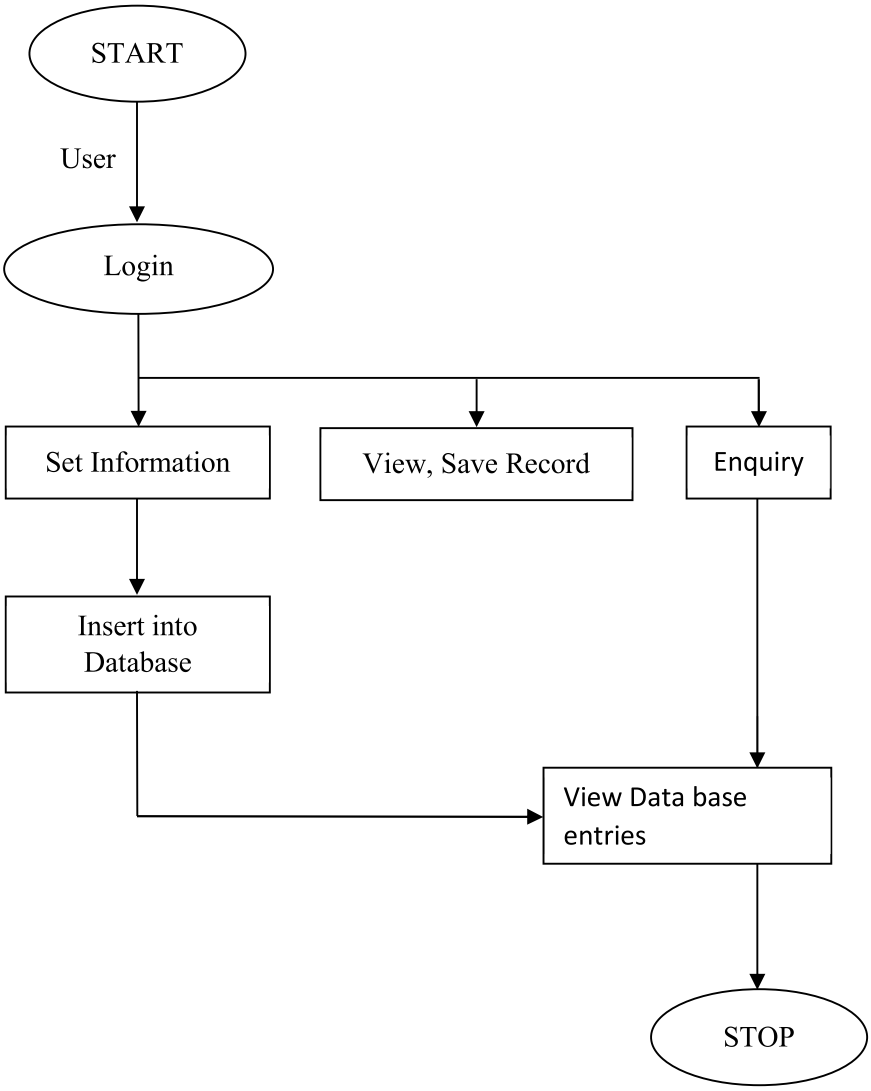


Figure 7.2: Control Hierarchy Diagram

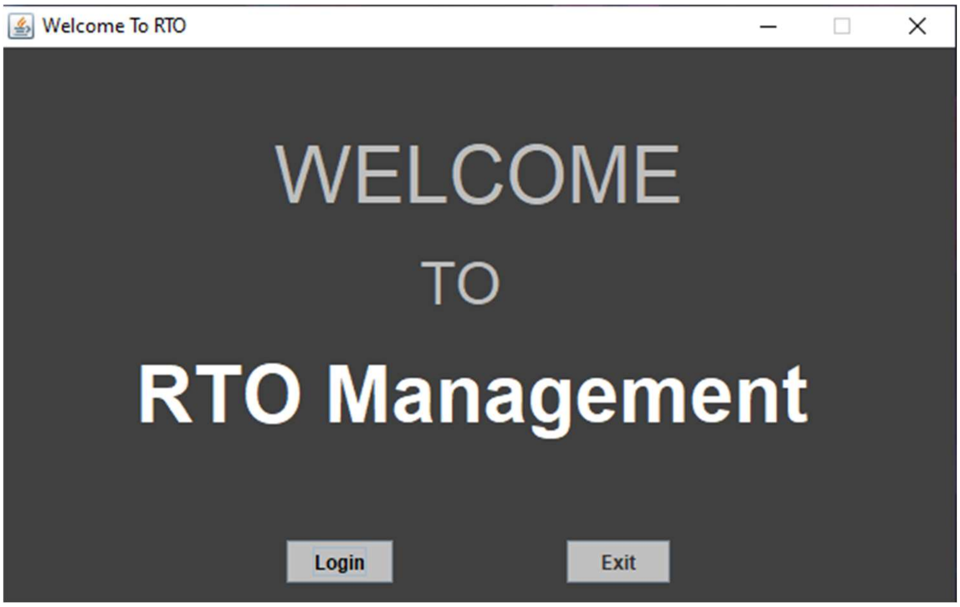
7.3 INTERFACE DESIGN

Interface Design of a software is one of the most important facets of software design. The more intuitive the user interface, the more interested the users are in using the software. For RTO Management System, the user interface of the system is easy to navigate.

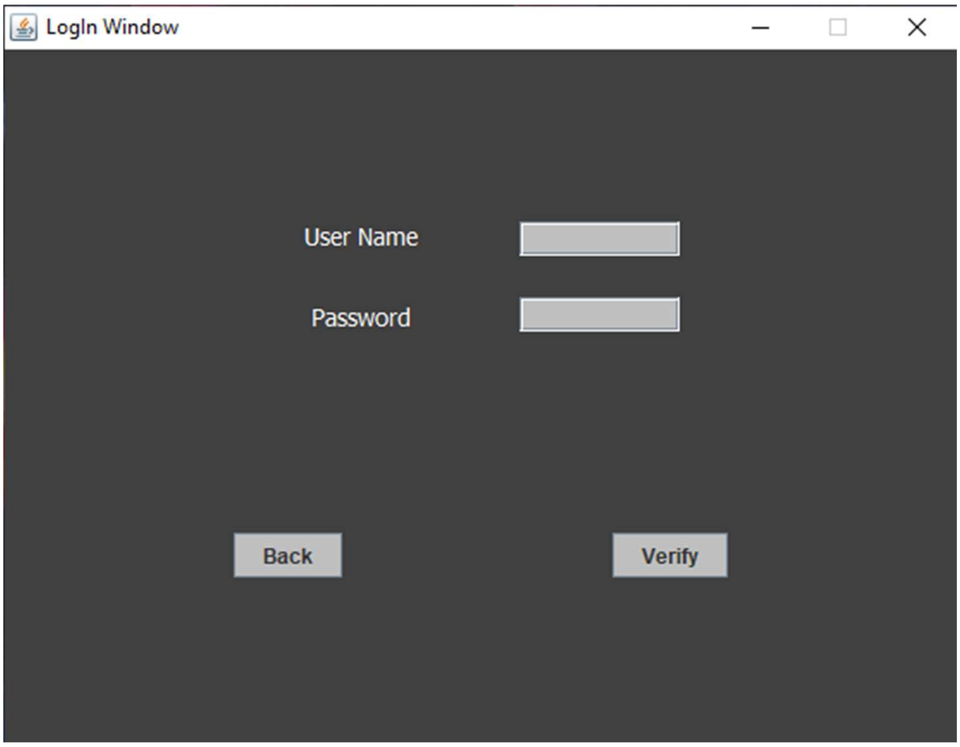
The first window of the project is the Welcome Window.

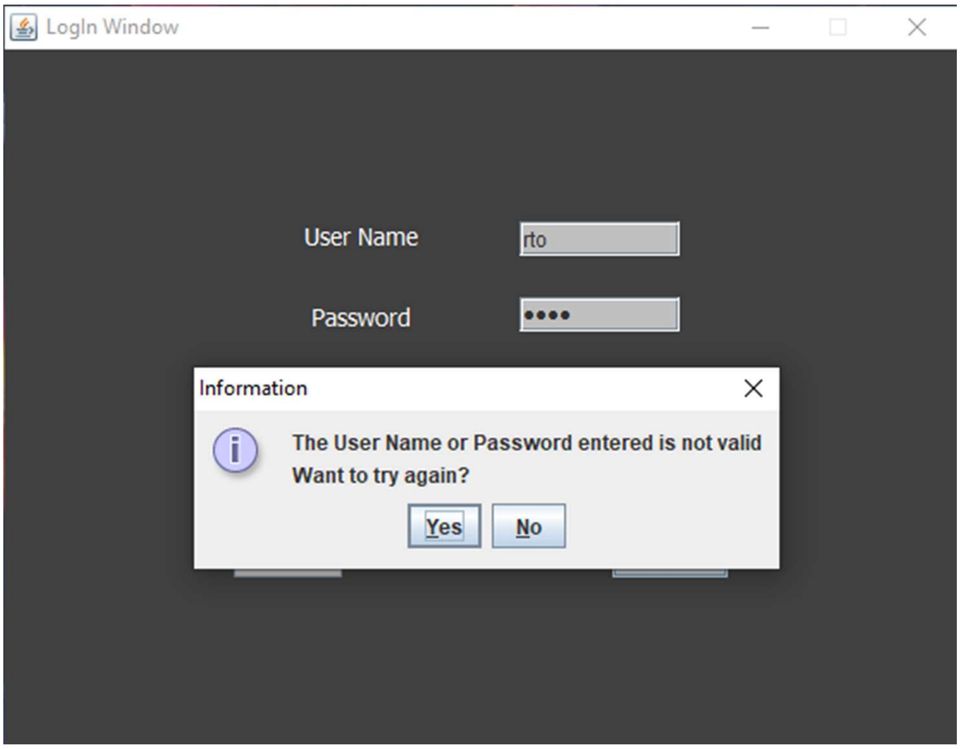
It has two buttons:

1. Login 2. Exit



The second window after welcome window is the Login Window. It is made for security purpose.



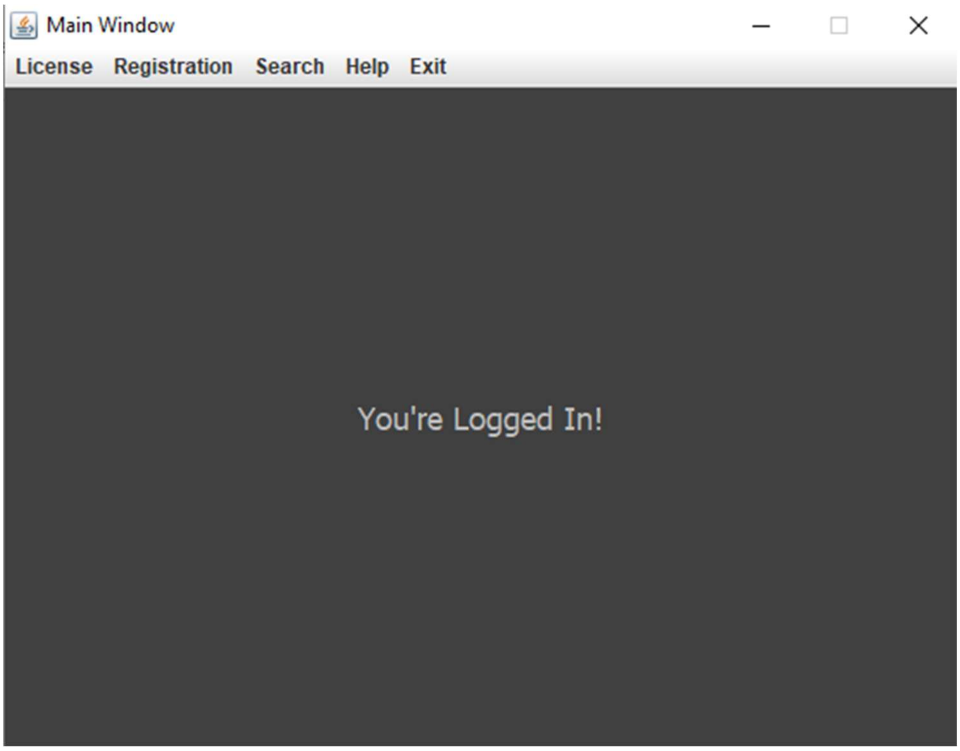


User Name: rto

Password: ••••

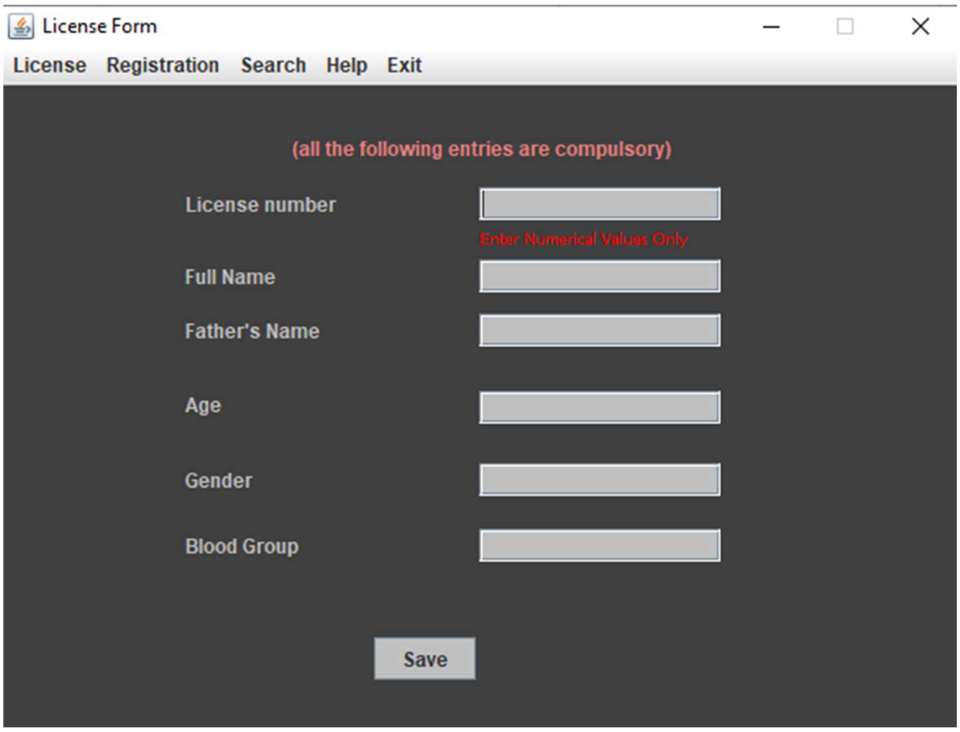
The User Name or Password entered is not valid
Want to try again?

The third window is the Main window. Options in this window are License, Registration, Search, Help, Exit.



You're Logged In!

This is License Form window can be opened by any window. Menus in this window are License, Registration, Search, Help, Exit. This window takes entry for new License.



(all the following entries are compulsory)

License number

Enter Numerical Values Only

Full Name

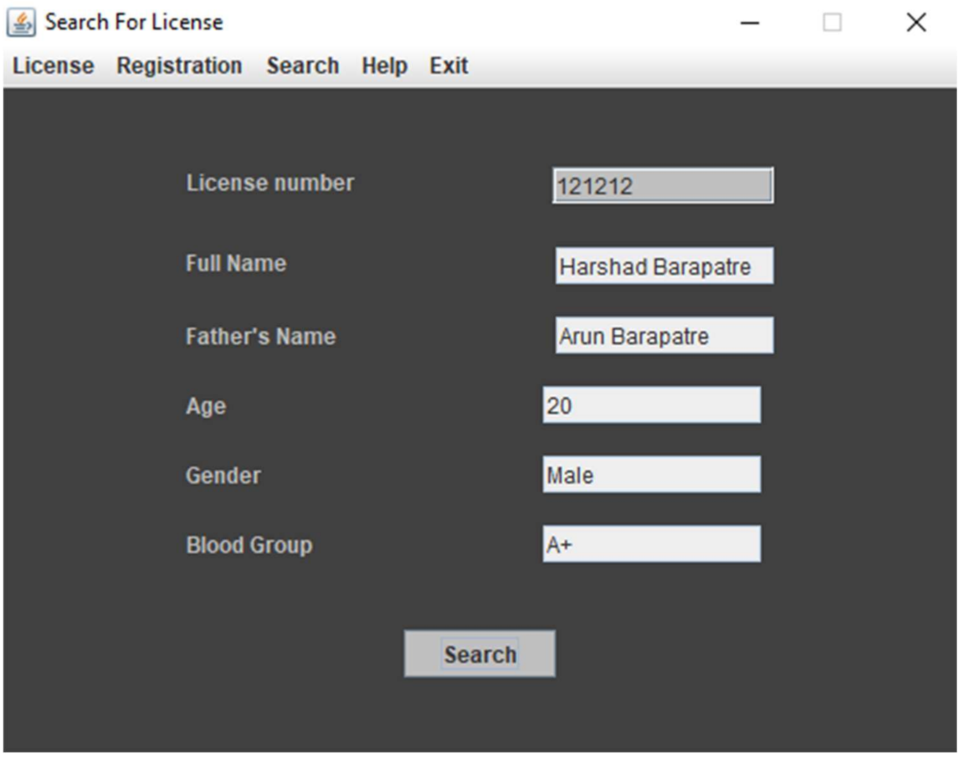
Father's Name

Age

Gender

Blood Group

This is License Search window can be opened by any window. Menus in this window are License, Registration, Search, Help, Exit. This window searches stored details for an entered License no.



License number: 121212

Full Name: Harshad Barapatre

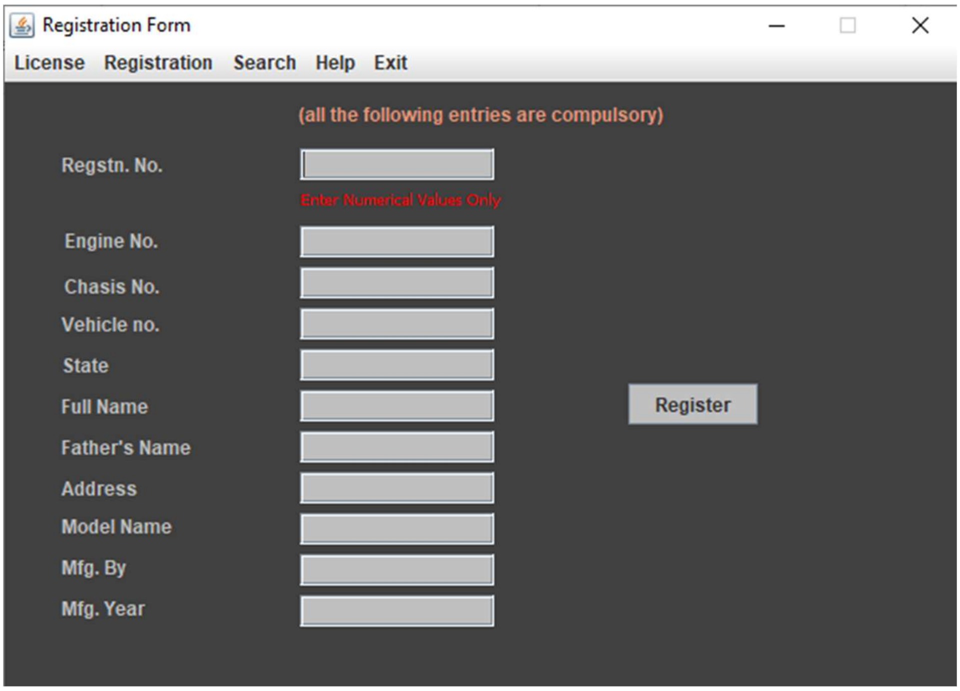
Father's Name: Arun Barapatre

Age: 20

Gender: Male

Blood Group: A+

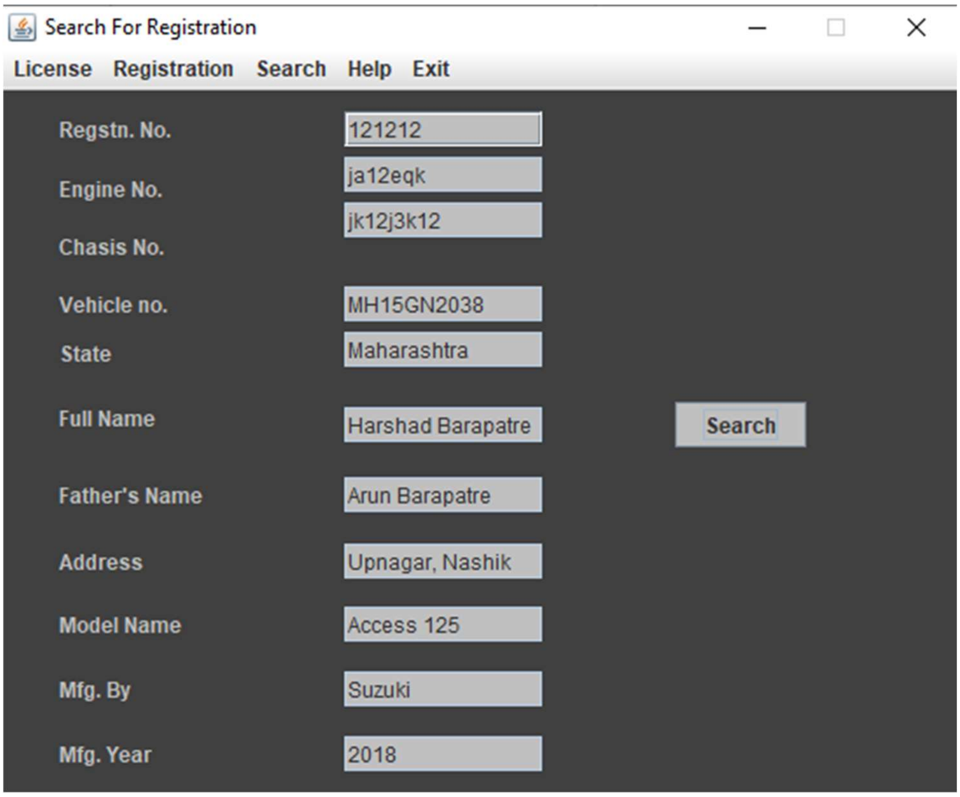
This is Registration Form window can be opened by any window. Menus in this window are License, Registration, Search, Help, Exit. This window takes entry for new License.



(all the following entries are compulsory)

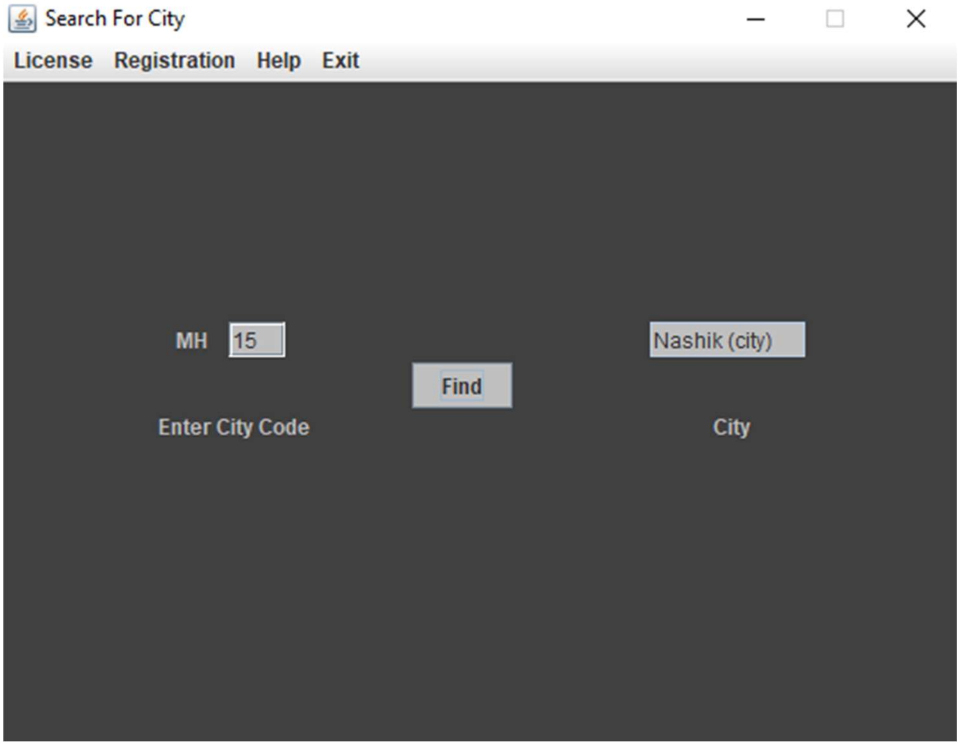
Enter Numerical Values Only

This is Registration Search window can be opened by any window. Menus in this window are License, Registration, Search, Help, Exit. This window searches stored Registration details for an entered registration no.

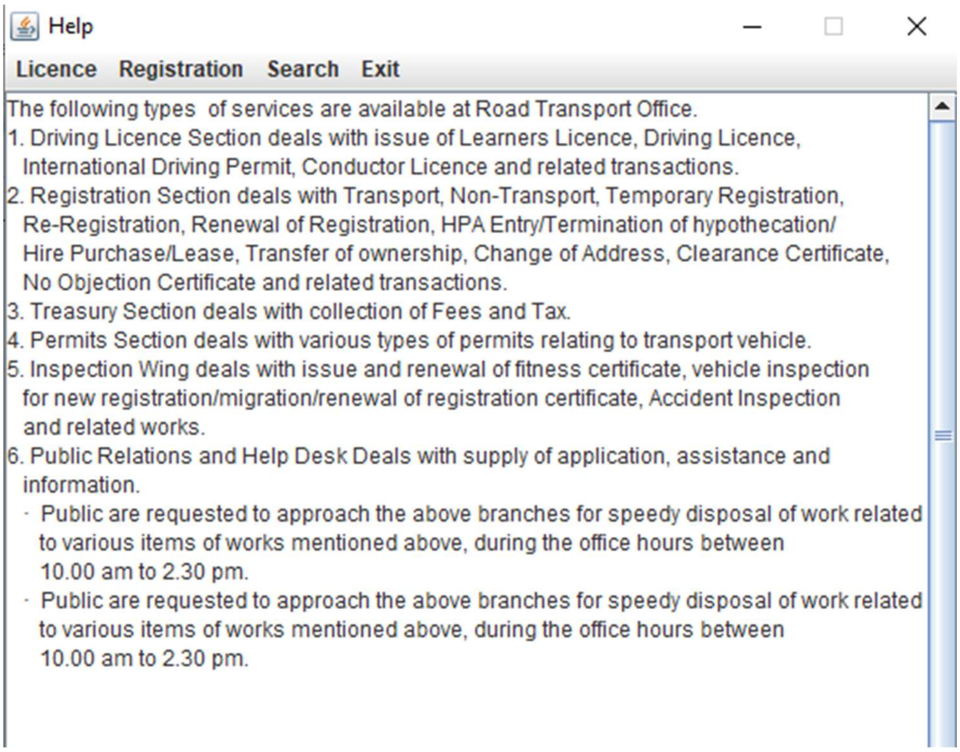


Regstn. No. 121212
Engine No. ja12eqk
Chasis No. jk12j3k12
Vehicle no. MH15GN2038
State Maharashtra
Full Name Harshad Barapatre
Father's Name Arun Barapatre
Address Upnagar, Nashik
Model Name Access 125
Mfg. By Suzuki
Mfg. Year 2018

This is Search City Form window can be opened by any window. Menus in this window are License, Registration, Help, Exit. This window searches for any entered city code



This is Help window can be opened by any window. Menus in this window are License, Registration, Search, Exit. This window has information about RTO.



The following types of services are available at Road Transport Office.

1. Driving Licence Section deals with issue of Learners Licence, Driving Licence, International Driving Permit, Conductor Licence and related transactions.
2. Registration Section deals with Transport, Non-Transport, Temporary Registration, Re-Registration, Renewal of Registration, HPA Entry/Termination of hypothecation/ Hire Purchase/Lease, Transfer of ownership, Change of Address, Clearance Certificate, No Objection Certificate and related transactions.
3. Treasury Section deals with collection of Fees and Tax.
4. Permits Section deals with various types of permits relating to transport vehicle.
5. Inspection Wing deals with issue and renewal of fitness certificate, vehicle inspection for new registration/migration/renewal of registration certificate, Accident Inspection and related works.
6. Public Relations and Help Desk Deals with supply of application, assistance and information.
   - Public are requested to approach the above branches for speedy disposal of work related to various items of works mentioned above, during the office hours between 10.00 am to 2.30 pm.
   - Public are requested to approach the above branches for speedy disposal of work related to various items of works mentioned above, during the office hours between 10.00 am to 2.30 pm.

7.4 ACTIVITY DIAGRAM FOR USER

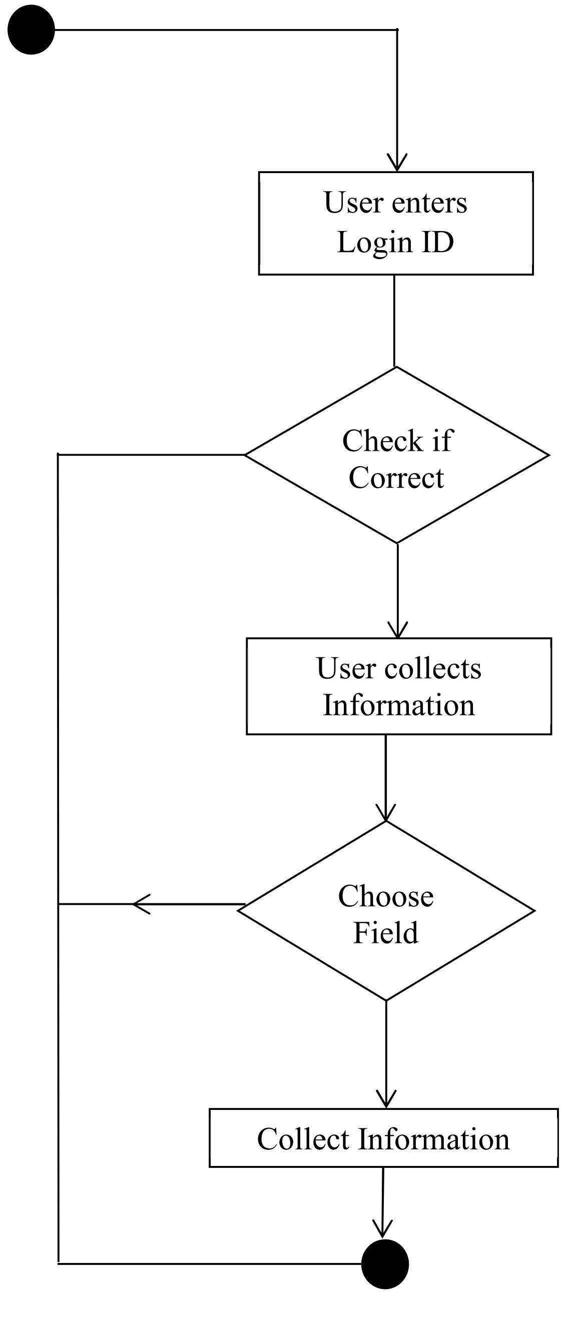


Figure 7.3: Activity Diagram

7.5 DATA DESIGN

License

Field Name	Data type	Description
Licenceno	Number	Primary key
Name	Text	---------------
Fathername	Text	---------------
Age	Number	---------------
Gender	Text	---------------
Bloodgroup	Text	---------------

Registration

Field Name	Data type	Description
Regnno	Number	Primary key
Engineno	Text	---------------
Chasisno	Text	---------------
Vehicleno	Text	---------------
State	Text	---------------
Name	Text	---------------
Fathername	Text	---------------
Address	Text	---------------
Make	Text	---------------
Mfgby	Text	---------------
Mfgyear	Number	---------------

Citycode

Field Name	Data type	Description
Code	Number	---------------
Name	Number	---------------

CHAPTER 8
SCOPE & LIMITATIONS

The future scope and limitations of the proposed RTO Management System are widespread and they are discussed in this chapter.

8.1 LIMITATIONS OF PROJECT

Any system designed is not perfect in its way. It has its own limitations. The system here was a first step to the huge world of software, and so is bound to have its own limitations.

- No checking for the entered data domain corresponding to each field of record, the software lacks this facility.
- If there is any transfer of vehicle papers database doesn’t update accordingly.

8.2 FUTURE ENHANCEMENTS

RTO Management System designed in this project can be further developed and the areas of its applications can be widened. The software can be extended to work with larger databases. It can be extended to work for large levels like networked RTO database access. Furthermore, by connecting the software at each branch of RTO, a larger inter-connecting network of management software can be achieved which can have the following capabilities:

- Accessing license information from different RTO offices.
- Registering vehicles in an RTO office which is in another location.
- More information on the officials can be incorporated.

Thus, scope of the Software can be extended to working live projects.

CHAPTER 9

CONCLUSION

The development of online RTO Management system involved many phases. The approach used is a top-down one concentrating on what first, then how and moving to successive levels of details. The first phase started with a detailed study of the problems and prospects in the existing system.

In the course of this study, many problems were discovered to have hindered the effectiveness of the existing system. These problems, information needs and activities were documented and later used as the basis for system design, which immediately followed the first phase. During this phase, strict adherence was made on proven software engineering principles and practices.

To implement this design, a computer program was then written and tested in phpMyAdmin environment. Waterfall methodology is used and the project follows all the above scope and has the above-mentioned limitations.

The problems of the existing partial computerized system are overcome. It is hoped that effective implementation of this software product would eliminate many problems discovered during systems investigation. The aims and objectives were considered while building the software.

RTO provides the facility of applying licenses online, issuance of permanent license, tax challans, and receiving payments against challans.

- The project has been appreciated by all the users in the organization.
- It is easy to use, since it uses the GUI provided in the user dialog.
- User friendly screens are provided.
- The usage of software increases the efficiency, decreases the effort.
- It has been efficiently employed as a project management mechanism.

REFERENCES

Books Referred

Advanced JAVA 2 How to Program	: By Deitel & Deitel
Software Engineering	: By Roger Pressman
Fundamentals of Database Systems	: By Elmasri Navathe

Websites Referred

- https://www.javatpoint.com/java-swing
- https://www.javatpoint.com/javafx-tutorial
- https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-installing-classpath.html