

1. Introduction

1.1 Background: -

Nowadays, there are car reservations which give much benefit to user. A rental service is a service in which customers arrive to request the hire of a rental unit. It is more convenient than carrying the cost of owning and maintaining the unit.

This project is designed so as to be used by Car Rental Company specializing in renting cars to customers. It is an system through which customers can view available cars, register, view profile and book car.

A car rental or car hire agency is a company that rents automobiles for short period of time for a fee whether in a few hours or a few days or week. It is an extended form of a rental shop, often organized with numerous local branches (which allow a user to return a car to a different location), and primarily located near airports or busy city areas and often complemented by a website allowing reservations.

Car rental agencies primarily serve people who have cars that are temporarily out of reach or out of service, for example travelers who are out of town or owners of damaged or destroyed cars who are awaiting repair or insurance compensation.

Because of the variety of sizes of the cars, our company serves the self-moving industry needs, by renting vans or trucks, and in certain markets other types of cars such as motorcycles or scooters are also offered. In short, it is a system design especially for large, premium and small car rental business. The car rental system provides complete functionality of listing and booking cars.

1.2 Objective: -

Today's world is computer world because most of work is doing with the help of computer. Dependency on computer is behind the few reasons. We cannot easily manage to store large number of data or information single handle. If we will be needing some information or data in urgency then we cannot manage in manually these works are very difficult if we cannot use computer.

REPORT ON VEHICLE RENTING SERVICE

This software is basically updating the manual chemist Inventory system to automated Inventory system. So that organization can manage their record in efficient and organize them.

- The main objective is to automate non-computer environment
- To save manpower.
- It will speed the processing of data and transaction.
- It will provide best security features such as provisions of passwords
- To transform the manual process of hiring car to a computerize system.
- To validate the Rental car system using user satisfaction test.
- To produce the documentation such as Software Requirement Specification (SRS), Software Design Description (SDD) as system development reference.
- To produce a web-based system that allow customer to register and reserve car and for the company to effectively manage their car rental business.
- To ease customer's task whenever they need to rent a car.

1.2.1 System Objective: -

Today's world is computer world because most of work is doing with the help of computer. Dependency on computer is behind the few reasons. We cannot easily manage to store large number of data or information single handle.

If we will be needing some information or data in urgency then we cannot manage in manually these works are very difficult if we cannot use computer.

1.2.2 System Context: -

This section clearly depicts the environment and boundaries of the Car Rental System and the entities with which it interacts. It helps us see how the system fits into the existing scheme of things. What the system will do by itself.

REPORT ON VEHICLE RENTING SERVICE

1. Functional Requirement: -

This Software must request Username and Password for access to data, only after authentication will allow access to the system. The Software must allow input of products data from administrator and secured access.

Requirement analysis is a software engineering technique that is composed of the various tasks that determine the needs or conditions that are to be met for a new or altered product, taking into consideration the possible conflicting requirements of the various users.

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data should the system holds and the interfaces with the user.

The functional requirements identified are:

- a) **Customer's registration:** The system should allow new users to register and generate membership card.
- b) **Reservation of cars:** Customers should be able to use the system to make booking and reservation.
- c) **Automatic update to database once reservation is made or new customer registered:** Whenever there's new reservation or new registration, the system should be able update the database without any additional efforts from the admin.
- e) **Feedbacks to customers:** It should provide means for customers to leave feedback.

2. Non-Functional Requirement: -

In this Software Input error will be returned in red with appropriate message box. System should automatically update after every transaction. It describes

REPORT ON VEHICLE RENTING SERVICE

aspects of the system that are concerned with how the system provides the functional requirements. They are:

- a) **Security:** The subsystem should provide a high level of security and integrity of the data held by the system, only authorized personnel of the company can gain access To the Company's secured page on the system; and only users with valid password and username can login to view user's page.
- b) **Performance and Response time:** The system should have high performance rate when executing user's input and should be able to provide feedback or response within a short time span usually 50 seconds for highly complicated task and 20 to 25seconds for less complicated task.
- c) **Error handling:** Error should be considerably minimized and an appropriate error message that guides the user to recover from an error should be provided. Validation of user's input is highly essential. Also, the standard time taken to recover from an error should be 15 to 20 seconds.
- d) **Availability:** This system should always be available for access at 24 hours, 7 days awake. Also, in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that the business process is not severely affected.
- e) **Ease of use:** Considered the level of knowledge possessed by the users of this system, a simple but quality user interface should be developed to make it easy to understand and required less training.

1.3 Purpose and Scope

1.3.1 Purpose: -

The purpose of this document is to specify requirements and to give guidelines for the development of above said project. In particular it gives guidelines on how to prepare the above said project.

REPORT ON VEHICLE RENTING SERVICE

The advancement in Information Technology and internet penetration has greatly enhanced various business processes and communication between companies (services provider) and their customers of which car rental industry is not left out.

This Car Rental System is developed to provide the following services:

- **Enhance Business Processes:** To be able to use internet technology to project the rental company to the global world instead of limiting their services to their local domain alone, thus increase their return on investment (ROI).
- **Car Reservation:** Tools through which customers can reserve available cars prior to their expected pick-up date or time.
- **Customer's registration:** A registration portal to hold customer's details, monitor their transaction and used same to offer better and improve services to them.
- **Group bookings:** Allows the customer to book space for a group in the case of weddings or corporate meetings (Event management).

1.3.2 Scope: -

This project traverses a lot of areas ranging from business concept to computing field and required to perform several researches to be able to achieve the project objectives.

The area covers include:

- **Car rental industry:** This includes study on how the car rental business is being done, process involved and opportunity that exist for improvement.
- Java Technology used for the development of the application.
- General customers as well as the company's staff will be able to use the system effectively.
- Web-platform means that the system will be available for access 24/7 except when there is a temporary server issue which is expected to be minimal

REPORT ON VEHICLE RENTING SERVICE

- Eco-friendly: The monitoring of the car activity and the overall business becomes easy and includes the least of paper work.
- The software acts as an office that is open 24/7. It increases the efficiency of the management at offering quality services to the customers.
- It provides custom features development and support with the software.

2. SURVEY OF TECHNOLOGIES

2.1 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 low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plug-in for applets).

The latest version is Java 10, released on March 20, 2018, which follows Java 9 after only six months in line with the new release schedule. Java 8 is still supported but there will be no more security updates for Java 9. Versions earlier than Java 8 are supported by companies on a commercial basis; e.g. by Oracle back to Java 6 as of October 2017 (while they still "highly recommend that you uninstall" pre-Java 8 from at least Windows computers).

2.2 Versions

As of 20 March 2018, both Java 8 and 10 are officially supported. Major release versions of Java, along with their release dates:

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
- Java SE 7 (July 28, 2011)
- Java SE 8 (March 18, 2014)
- Java SE 9 (September 21, 2017)
- Java SE 10 (March 20, 2018)

2.3 MySQL: -

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation 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. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

REPORT ON VEHICLE RENTING SERVICE

Release history

| Release | General availability | Latest minor version | Latest release | End of Support |
|----------------|--------------------------------|-----------------------------|-----------------------|-----------------------|
| 5.1 | November 14, 2008; 9 years ago | 5.1.73 | 2013-12-03 | December 2013 |
| 5.5 | December 3, 2010; 7 years ago | 5.5.60 | 2018-04-19 | December 2018 |
| 5.6 | February 5, 2013; 5 years ago | 5.6.40 | 2018-04-19 | February 2021 |
| 5.7 | October 21, 2015; 2 years ago | 5.7.22 | 2018-04-19 | October 2023 |
| 8.0 | April 19, 2018; 4 months ago | 8.0.11 | 2018-04-19 | N/A |

3. Requirements and Analysis

3.1 Problem Definition: -

In a car rental service, a car that can be used temporarily by paying a fee during a specified period. Getting a rental car helps people get around despite the fact they do not have access to their own personal car or don't own a car at all.

The individual who needs a car must contact a rental car company and contract out for a car. This system increases customer retention and simplify car and staff management.

3.2 Planning and Scheduling: -

PERT CHART

A project plan needs to be created to ensure the timely completion of the project. As part of project analysis, we break the project down to a number of stages and use a Gantt chart and PERT chart to describe specific tasks and status. The Work Breakdown Structure of our proposed system “Car Rental System” is shown below.

| ID | Task Name | Duration | Start | Finish |
|----|--------------------|----------|----------|----------|
| 1 | Project Initiation | 3 days | 25/06/19 | 27/06/19 |
| 2 | Draft Project Plan | 4 days | 28/06/19 | 01/07/19 |
| 3 | Analysis Phase | 4 days | 02/07/19 | 05/07/19 |
| 4 | System Design | 10 days | 06/07/19 | 15/07/19 |
| 5 | Modules Design | 5 days | 16/07/19 | 20/07/19 |
| 6 | Coding Phase | 31 days | 21/07/19 | 21/08/19 |
| 7 | Testing Phase | 18days | 22/08/19 | 08/09/19 |
| 8 | Implementation | 12 days | 09/09/19 | 20/09/19 |

REPORT ON VEHICLE RENTING SERVICE

GANTT CHART

| Task No. | Task | June | July | August | September |
|----------|----------------------|------|------|--------|-----------|
| 1 | Requirement Analysis | | | | |
| 2 | Design | | | | |
| 3 | Coding | | | | |
| 4 | Testing | | | | |
| 5 | Implementation | | | | |

3.3 Requirements Specification: -

3.3.1 Software Requirements

- 1 Java/JDK
- 2 Net Beans
- 3 MySQL
- 4 SQL YOG

3.3.2 Hardware Requirements

- Pentium IV Processor
- 512 MB RAM
- 40 GB HDD
- Color Monitor
- Keyboard, Mouse

3.4 Preliminary Product Description: -

It would be a multi user account system in key features. There will three types of main modules in the system

- Admin
- Employee
- Invoice Management

ADMIN

Like every other management system, the car rental management system will have the admin. The admin will be the entity that will monitor the activities and the records of whole system.

Following are some main facts related to the admin of the system. There will be only one admin in the system. Admin can view other user's profile. The admin will have the power to delete any other users from the records or update the data of any other users. Any car or the payment deal will be approved by the admin.

EMPLOYEE

Customers are the reason why I feel to introduce the car rental management system, to make their journey wonderful, to get them fit for the environment they are traveling into.

View the Cars –

You, as a customer, can observe the lists of cars available in the inventory. The user can filter the records of the car based on:

Price - The budget is an important factor. It will be easier to choose a car rather than wondering what if I choose this car and the price is higher. No tension at all, you can analyze the car record and choose your best car.

REPORT ON VEHICLE RENTING SERVICE

Popular Cars: If you want to take the car which is popular in the system rather than thinking about the fact how this car would perform, you better look into it. The already registered customers have given the feedback of their car driving experience.

Car Brand: If you are into brand, you can view the cars of your favorite brand. I have taken that too into the account. The car brand can be BMW, Mercedes, Aston Martin, Honda, Mahindra etc. Just pick your pick.

Rent a Car: After you have selected your favorite car, you can fill the car rental form which is available . You just have to fill some details like for how many days you want to rent the car, or if you want to rent on hourly basis, the car details of the car model you want to rent. After completing the rental form, you can pay suitable amount using net banking, your credit / debit card.

Return a rented Car: The customer can return a car and if all the payments are cleared and the parts of the cars are not damaged, a number will be provided to the customer so that the customer can enter that number into the return car section and the record is cleared from the rent a car system and is moved to rental car history.

View Rental History: You as a user can view the history of the car you have rented in the car rental management system. You can keep track of the amount you have spent, the car you have driven, the number of cars you have rented etc.

Feedback: You as a user can share your experience with the car rental management system. How much you loved it, or hated it. You can give the stars and provide some comments so that the dealer can assist the customers to their best capability they can.

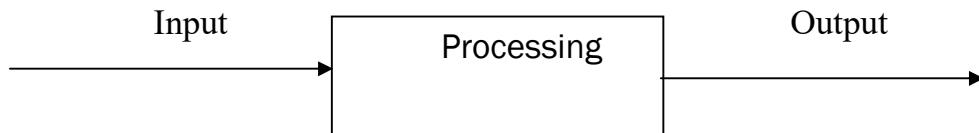
INVOICE MANAGEMENT

After the amount is paid by the customers, the invoice department will generate the bill of the car used and will reflect into the customers' account. This department will also keep the receipts of newly car is brought to the system so that it can further be used for analysis purpose.

3.5 CONCEPTUAL MODELS SYSTEM ARCHETECTURE DESIGN

DEFINING A SYSTEM: -

Collection of components, which are interconnected, and work together to realize some objective, from a system. There are three components in every system, namely input, processing and output



SYSTEM DEVELOPMENT LIFE CYCLE

The System development life cycle (SDLC), or Software development processing systems engineering, information systems and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of software development methodologies.

These methodologies form the framework for planning and controlling the creation of an information system the process. Broadly, following are the different activities to be considered while defining the system development life cycle for the said project:

- Problem Definition
- System Analysis
- Study of existing system
- Drawback of the existing system
- Proposed system

- System Requirement study
- Data flow analysis
- Feasibility study
- System design
- Input Design (Database & Forms)
- Updating
- Query /Report design
- Administration
- Testing
- Implementation
- Maintenance

1 SYSTEM ANALYSIS: -

Systems analysis is the study of sets of interacting entities, including computer systems analysis. This field is closely related to requirements analysis or operations research. It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made.

System development can generally be thought of having two major components: systems analysis and systems design. In System Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements.

Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvement to the system.

2 SYSTEM DESIGN: -

REPORT ON VEHICLE RENTING SERVICE

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

ENTITY RELATION DIAGRAM

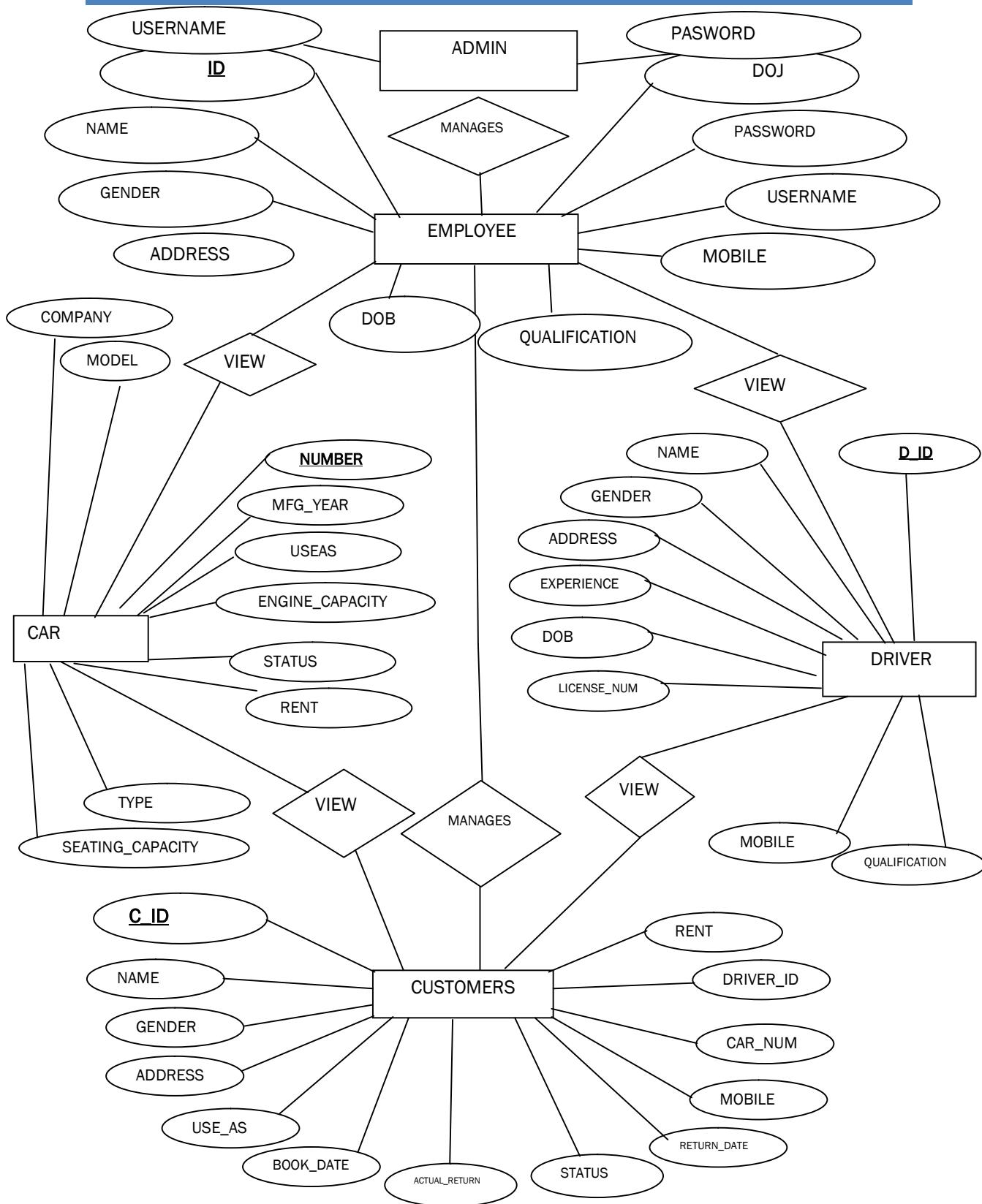
The Entity Relation Model or Entity Relation Diagram (ERD) is a data model or diagram for high-level description of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity relationship diagrams. Such models are typically used in the first stage of Management information system design; they are used for example, to describe information needs and/ or the type of information that is to be stored in the Database during the requirement analysis.

The data modeling technique, however, can be used to describe any ontology (i.e. an overview and classification of used term and their relationships) for a certain universe of discourse (i.e. area of interest).

In the case of design, a Management Information System that is based on a database, the conceptual data model is, a later stage (usually called logical design), mapped to a logical data model such as, relational data model; this is turn in mapped to a physical model during physical design.

Note that sometimes, both of the phases are referred a “physical design”. There are number of conventions for entity-relation diagrams (ERDs). The classical notation is described in the remainder of this article, and mainly related to the conceptual modeling. There is a range of notation more typically employed in physical and logical database design.

REPORT ON VEHICLE RENTING SERVICE



DATA FLOW DIAGRAM

The data flow diagram shows the flow of data within any system. It is an important tool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It's also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD.

In the DFD, four symbols are used and they are as follows.

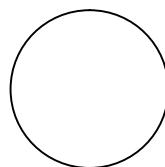
1. A square defines a source (originator) or destination of system data.



2. An arrow identifies data flow-data in motion. It is a pipeline through which information flows.



3. A circle or a “bubble” (Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.

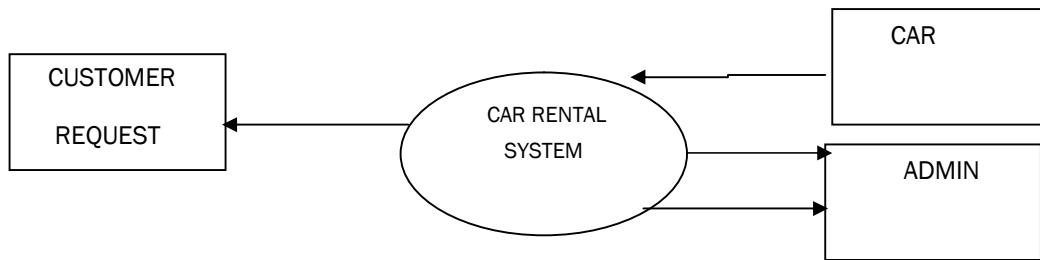


4. An open rectangle is a data store-data at rest, or a temporary Repository of data.



Context Level Data Flow Diagram: -

This level shows the overall context of the system and its operating environment and shows the whole system as just one process. book store is shown as one process in the context diagram; which is also known as zero level DFD, shown below.



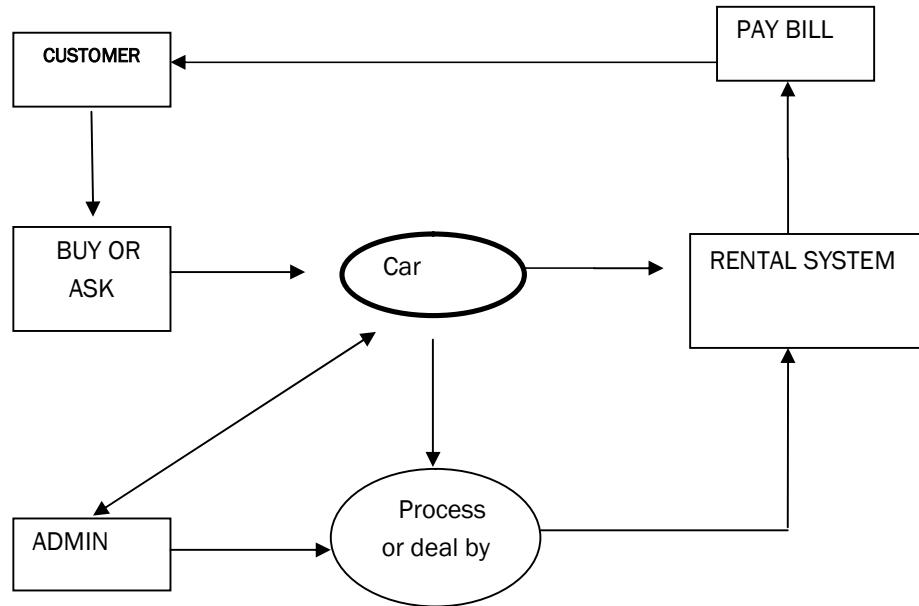
Zero Level DFD

The context diagram plays important role in understanding the system and determining the boundaries. The main process can be broken into sub-processes and system can be studied with more detail; this is where 1st level DFD comes into play.

First Level DFD: -

This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major high-level processes of the system and their interrelation. A process model will have one, and only one, level-1 diagram. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level 1.

REPORT ON VEHICLE RENTING SERVICE



First level DFD

4. DATA TABLE

SNAPSHOT

REPORT ON VEHICLE RENTING SERVICE

LOGIN ADMIN TABLE

| Field Name | Datatype | Len | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | Comment |
|------------|----------|-----|---------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|---------|
| * USERNAME | varchar | 20 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| PASSWORD | int | 11 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| | | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

ADD CAR TABLE

| Field Name | Datatype | Len | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | Comment |
|------------------|----------|-----|---------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|---------|
| * COMPANY | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| MODEL | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| NUMBER | varchar | 20 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| MFG_YEAR | varchar | 40 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| USEAS | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| SEATING_CAPACITY | int | 11 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| TYPE | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ENGINE_CAPACITY | varchar | 11 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| RENT | int | 11 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| STATUS | varchar | 30 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

REPORT ON VEHICLE RENTING SERVICE

ADD CUSTOMER TABLE

| Field Name | Datatype | Len | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | Comment |
|---------------|----------|-----|---------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|---------|
| C_ID | int | 11 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| NAME | varchar | 30 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| GENDER | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ADDRESS | varchar | 50 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| USE_AS | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| BOOK_DATE | varchar | 40 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| RETURN_DATE | varchar | 40 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| MOBILE | bigint | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| CAR_NUM | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| DRIVER_ID | int | 11 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| RENT | int | 11 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ACTUAL_RETURN | varchar | 40 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| STATUS | varchar | 30 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

ADD COMPANY TABLE

| Field Name | Datatype | Len | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | Comment |
|------------|----------|-----|---------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|---------|
| CARNAME | varchar | 30 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| | | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

ADD DRIVER TABLE

| Field Name | Datatype | Len | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | Comment |
|---------------|----------|-----|---------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|---------|
| D_ID | int | 11 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| NAME | varchar | 30 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| GENDER | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ADDRESS | varchar | 40 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| EXPERIENCE | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| DOB | varchar | 40 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| LICENSE_NUM | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| QUALIFICATION | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| MOBILE | bigint | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

REPORT ON VEHICLE RENTING SERVICE

ADD EMPLOYEE TABLE

| Field Name | Datatype | Len | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | Comment |
|---------------|----------|--------|---------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|---------|
| ID | int | ▼ 11 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| NAME | varchar | ▼ 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| GENDER | varchar | ▼ 20 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ADDRESS | varchar | ▼ 60 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| DOB | varchar | ▼ 50 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| QUALIFICATION | varchar | ▼ 30 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| MOBILE | decimal | ▼ 10,0 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| USERNAME | varchar | ▼ 30 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| PASSWORD | varchar | ▼ 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| DOJ | varchar | ▼ 50 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| | | ▼ | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

5. DATA TABLE

REPORT ON VEHICLE RENTING SERVICE

loginadmin: -

| <u>Name Column</u> | <u>Data Type</u> | <u>Size</u> | <u>Constraints</u> |
|--------------------|------------------|-------------|--------------------|
| Username | Varchar | 20 | Primary key |
| Password | Varchar | 11 | Not Null |

carcompany: -

| <u>Name Column</u> | <u>Data Type</u> | <u>Size</u> | <u>Constraints</u> |
|--------------------|------------------|-------------|--------------------|
| CARNAME | Varchar | 30 | Primary key |

customers list: -

| <u>Name Column</u> | <u>Data Type</u> | <u>Size</u> | <u>Constraints</u> |
|--------------------|------------------|-------------|--------------------|
| C_ID | Varchar | 20 | Primary key |
| NAME | Varchar | 30 | Not Null |
| GENDER | Varchar | 10 | Not Null |
| ADDRESS | Varchar | 50 | Not Null |
| USE_AS | Varchar | 40 | Not Null |
| BOOK_DATE | Varchar | 20 | Not Null |
| RETURN_DATE | Varchar | 30 | Not Null |
| MOBILE | bigint | 20 | Not Null |
| STATUS | Varchar | 20 | Not Null |
| CAR_NUM | Varchar | 40 | Not Null |
| DRIVER_ID | int | 20 | Not Null |
| RENT | int | 20 | Not Null |
| ACTUAL_RETURN | Varchar | 40 | Not Null |

REPORT ON VEHICLE RENTING SERVICE

drivers list: -

| <u>Name Column</u> | <u>Data Type</u> | <u>Size</u> | <u>Constraints</u> |
|--------------------|------------------|-------------|--------------------|
| D_ID | Varchar | 40 | Primary key |
| NAME | Varchar | 30 | Not Null |
| GENDER | Varchar | 10 | Not Null |
| ADDRESS | Varchar | 60 | Not Null |
| EXPERIENCE | Varchar | 50 | Not Null |
| DOB | Varchar | 30 | Not Null |
| LICENSE_NUM | Varchar | 10 | Not Null |
| QUALIFICATION | Varchar | 30 | Not Null |
| MOBILE | Bigint | 40 | Not Null |

employee list: -

| <u>Name Column</u> | <u>Data Type</u> | <u>Size</u> | <u>Constraints</u> |
|--------------------|------------------|-------------|--------------------|
| ID | Varchar | 40 | Primary Key |
| NAME | Varchar | 30 | Not Null |
| GENDER | Varchar | 6 | Not Null |
| ADDRESS | Varchar | 50 | Not Null |
| DOB | Varchar | 20 | Not Null |
| QUALIFICATION | Varchar | 20 | Not Null |
| MOBILE | Decimal | 10,0 | Not Null |

REPORT ON VEHICLE RENTING SERVICE

| | | | |
|----------|---------|----|----------|
| USERNAME | Varchar | 40 | Not Null |
| PASSWORD | Varchar | 40 | Not Null |
| DOJ | Varchar | 20 | Not Null |

car list: -

| Name Column | Data Type | Size | Constraints |
|------------------|-----------|------|-------------|
| COMPANY | Varchar | 40 | Not Null |
| MODEL | Varchar | 20 | Not Null |
| NUMBER | Varchar | 20 | Primary key |
| MFG_YEAR | Year | 4 | Not Null |
| USEAS | Varchar | 30 | Not Null |
| SEATING_CAPACITY | Varchar | 30 | Not Null |
| TYPE | Varchar | 20 | Not Null |
| ENGINE_CAPACITY | Varchar | 30 | Not Null |
| RENT | Varchar | 20 | Not Null |
| STATUS | Varchar | 30 | Not Null |

6.CODING

Database Connectivity: -

Coding: -

```
package db;

import java.sql.*;

public class DBConnection {

    public Connection con;
    public Statement stmt;
    public ResultSet rs;
    public PreparedStatement pstmt;
    public DBConnection()

    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/car_renting",
"root", "root");
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

ANIMATION: -

CODING: -

```
package car_renting_system;  
  
public class Car_Renting_System  
{  
    public static void main(String[] args)  
    {  
        splash sp=new splash();  
        sp.setVisible(true);  
        login l=new login();  
        try  
        {  
            for(int z=0;z<=100;z++)  
            {  
                Thread.sleep(30);  
                sp.jProgressBar1.setValue(z);  
                if(z==100)  
                {  
                    sp.setVisible(false);  
                    l.setVisible(true);  
                }  
            }  
        }  
        catch(Exception e)
```

REPORT ON VEHICLE RENTING SERVICE

```
{  
    e.printStackTrace();  
}  
}  
}  
}
```

LOGIN: -

Coding On Login as Employee Button: -

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    LOGINASEMPLOYEE.setVisible(true);  
  
}
```

Coding On Login as Admin Button : -

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    LOGINASADMIN.setVisible(true);  
  
}
```

Coding On Admin Login Button: -

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
        DBConnection db = new DBConnection();  
        String username = jTextField1.getText();  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
int pass = Integer.parseInt(jTextField2.getText());
db.stmt = db.con.createStatement();
db.rs = db.stmt.executeQuery("select * from loginadmin where
USERNAME='"+username+"' and PASSWORD = '"+pass+"'");
if(db.rs.next())
{
    this.setVisible(false);

ADMIN_WINDOW aw = new ADMIN_WINDOW();
aw.setVisible(true);
}
else
{
    JOptionPane.showMessageDialog(this,"Sorry! Worong Username Or
Password.");
}
catch(Exception e)
{
    e.printStackTrace();
}
}
```

Coding On Employee Login Button: -

REPORT ON VEHICLE RENTING SERVICE

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        DBConnection db = new DBConnection();  
        String username = jTextField3.getText().toString();  
        int pass = Integer.parseInt(jTextField4.getText());  
        String query = "select * from employee_list where  
username='"+username+"' and password = '"+pass+"';"  
        db.stmt = db.con.createStatement();  
        db.rs = db.stmt.executeQuery(query);  
        if(db.rs.next())  
        {  
            this.setVisible(false);  
            EMPLOYEE_WINDOW ew = new EMPLOYEE_WINDOW();  
            ew.setVisible(true);  
        }  
        else  
        {  
            JOptionPane.showMessageDialog(this,"Sorry! wrong Username or  
Password ");  
        }  
    }  
    catch(Exception e)  
    {  
    }
```

```
        e.printStackTrace();  
    }  
}
```

Employee Main Window Coding: -

ADD DRIVER: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    try
```

```
{
```

```
    DBConnection db = new DBConnection();
```

```
    String sql = "select MAX(D_ID) from drivers_list";
```

```
    db.stmt = db.con.createStatement();
```

```
    db.rs = db.stmt.executeQuery(sql);
```

```
    while(db.rs.next())
```

```
{
```

```
    int id = db.rs.getInt(1);
```

```
    int id1=id+1;
```

```
    String d_id = Integer.toString(id1);
```

```
    jTextField36.setText(d_id);
```

```
    }  
  
    ADD_DRIVER.setVisible(true);  
  
}  
  
catch(Exception e)  
  
{  
  
}  
  
}  
  
}
```

Coding On Submit Button: -

```
private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {  
  
        DBConnection db = new DBConnection();  
  
        int id = Integer.parseInt(jTextField36.getText());  
  
        String name = jTextField22.getText().toString();  
  
        String gender = "";  
  
        if(jRadioButton5.isSelected())  
  
        {  
  
            gender="MALE";  
  
        }  
  
        if(jRadioButton6.isSelected())  
  
        {  
  
            gender="FEMALE";  
  
        }  
  
    }  
  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
}

String area = jTextArea3.getText();

String expe = jComboBox6.getSelectedItem().toString();

String dob = jTextField29.getText();

String license = jTextField24.getText();

String mob = jTextField25.getText();

String quali = jComboBox9.getSelectedItem().toString();

String sql= "insert into drivers_list
values("+id+","+name+","+gender+","+area+","+expe+","+dob+","+license+",
"+quali+","+mob+");"

db(stmt = db.con.createStatement());

int i = db(stmt.executeUpdate(sql);

if(i>0)

{

    JOptionPane.showMessageDialog(this,"Record Successfully Added !");

}

else

{

    JOptionPane.showMessageDialog(this,"Failed To Add Record !");

}

}

} catch (SQLException ex) {
```

REPORT ON VEHICLE RENTING SERVICE

```
    Logger.getLogger(EMPLOYEE_WINDOW.class.getName()).log(Level.SEVERE, null, ex);

}
```

}

Coding On Cancel Button: -

```
private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {

    ADD_DRIVER.setVisible(false);

}
```

Modify Driver: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

    MODIFY_DRIVER.setVisible(true);

}
```

Coding On Search By ID Button: -

```
private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {

    //MODIFY DRIVER

    try

    {
```

REPORT ON VEHICLE RENTING SERVICE

```
int id = Integer.parseInt(jTextField15.getText());  
DBConnection db = new DBConnection();  
String sql ="Select * from drivers_list where D_ID = '"+id+"'";  
db.stmt = db.con.createStatement();  
db.rs = db.stmt.executeQuery(sql);  
if(db.rs.next())  
{  
  
jTextField15.setText(db.rs.getString(1));  
jTextField16.setText(db.rs.getString(2));  
String gen = db.rs.getString(3);  
if("MALE".equals(gen))  
{  
    jRadioButton3.setSelected(true);  
}  
else  
{  
    jRadioButton4.setSelected(true);  
}  
jTextArea2.setText(db.rs.getString(4));  
jComboBox4.setSelectedItem(db.rs.getString(5));  
jTextField23.setText(db.rs.getString(6));  
jTextField20.setText(db.rs.getString(7));  
jComboBox5.setSelectedItem(db.rs.getString(8));
```

REPORT ON VEHICLE RENTING SERVICE

```
jTextField17.setText(db.rs.getString(9));  
  
}  
  
else  
  
{  
  
    JOptionPane.showMessageDialog(this,"No Driver Found !");  
  
}  
  
}  
  
catch(Exception e)  
  
{  
  
    e.printStackTrace();  
  
}  
  
}
```

Coding On Search By Name Button: -

```
private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// MODIFY DRIVER  
  
try  
  
{  
  
    String name = jTextField16.getText();  
  
    DBConnection db = new DBConnection();  
  
    String sql ="Select * from drivers_list where Name = '"+name+"'";  
  
    db.stmt = db.con.createStatement();
```

REPORT ON VEHICLE RENTING SERVICE

```
db.rs = db.stmt.executeQuery(sql);
if(db.rs.next())
{
    jTextField15.setText(db.rs.getString(1));
    jTextField16.setText(db.rs.getString(2));
    String gen = db.rs.getString(3);
    if("MALE".equals(gen))
    {
        jRadioButton3.setSelected(true);
    }
    else
    {
        jRadioButton4.setSelected(true);
    }
    jTextArea2.setText(db.rs.getString(4));
    jTextField17.setText(db.rs.getString(9));
    jComboBox4.setSelectedItem(db.rs.getString(5));
    jTextField23.setText(db.rs.getString(6));
    jComboBox5.setSelectedItem(db.rs.getString(8));
    jTextField20.setText(db.rs.getString(7));
}
else
```

REPORT ON VEHICLE RENTING SERVICE

```
{  
    JOptionPane.showMessageDialog(this,"No Driver Found !");  
}  
}  
  
catch(Exception e)  
{  
    e.printStackTrace();  
}  
}  
}
```

Coding On Update Button: -

```
private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
        DBConnection db = new DBConnection();  
        int id = Integer.parseInt(jTextField15.getText());  
        String name = jTextField16.getText();  
        String gender = "";  
        if(jRadioButton3.isSelected())  
        {  
            gender ="MALE";  
        }  
        if(jRadioButton4.isSelected())  
        {  
            gender = "FEMALE";  
        }  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
}

String address = jTextArea2.getText();

String mob = jTextField17.getText();

String expe = jComboBox4.getSelectedItem().toString();

String dob = jTextField23.getText();

String quali = jComboBox5.getSelectedItem().toString();

String license = jTextField20.getText();

String sql = "update drivers_list set Name = '"+name+"',Gender  
='"+gender+"',Address= '"+address+"', Mobile='"+mob+"',  
Experience='"+expe+"',DOB='"+dob+"',  
Qualification='"+quali+"',License_NUM='"+license+"' where D_ID='"+id+"'";

db(stmt = db.con.createStatement());

int i = db.stmt.executeUpdate(sql);

if(i>0)

{

    JOptionPane.showMessageDialog(this,"Record Successfuly Updated  
!");

}

else

{

    JOptionPane.showMessageDialog(this,"Record Failed To Update !");

}

}

catch(Exception e)
```

REPORT ON VEHICLE RENTING SERVICE

```
{  
    e.printStackTrace();  
}  
}
```

Coding On Delete Button: -

```
private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
  
    {  
  
        DBConnection db = new DBConnection();  
  
        int id = Integer.parseInt(jTextField15.getText());  
  
        String Name = jTextField16.getText();  
  
        String sql = "delete from drivers_list where D_ID = '"+id+"' and NAME  
        ='"+Name+"';  
  
        db.stmt = db.con.createStatement();  
  
        int i = db.stmt.executeUpdate(sql);  
  
        if(i>0)  
  
        {  
  
            JOptionPane.showMessageDialog(this,"Record Successfully Deleted !");  
  
        }  
  
        else  
  
        {  
  
            JOptionPane.showMessageDialog(this,"Failed To Delete Record !");  
  
        }  
    }catch(Exception e)
```

REPORT ON VEHICLE RENTING SERVICE

```
{  
    e.printStackTrace();  
}  
  
}
```

Driver List: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {  
    DRIVERS_LIST.setVisible(true);  
}  
  
public void displaydriverslist()  
{  
    try  
    {  
        DBConnection db = new DBConnection();  
        String sql = "select * from drivers_list";  
        db.pstmt = db.con.prepareStatement(sql);  
        db.rs = db.pstmt.executeQuery();  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
jTable2.setModel(DbUtils.resultSetToTableModel(db.rs));  
}  
  
catch(Exception e)  
{  
  
    e.printStackTrace();  
}  
  
}
```

List Of Customer: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
public void displaycustomerslist()  
{  
  
    try  
{  
  
        DBConnection db = new DBConnection();  
  
        String sql = "select * from customers_list";  
  
        db.pstmt = db.con.prepareStatement(sql);  
  
        db.rs = db.pstmt.executeQuery();  
  
        jTable3.setModel(DbUtils.resultSetToTableModel(db.rs));  
    }  
  
    catch(Exception e)
```

```
{  
    e.printStackTrace();  
}  
}
```

Add Customer: -

Event Performed On Menu Item :-

Action Event performed(actionPerformed)

Coding: -

```
private void  
jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
        DBConnection db = new DBConnection();  
        String sql = "select MAX(C_ID) from customers_list";  
        db.stmt = db.con.createStatement();  
        db.rs = db.stmt.executeQuery(sql);  
        while(db.rs.next())  
        {  
            int id =db.rs.getInt(1);  
            int id1 = id+1;  
            String c_id = Integer.toString(id1);  
            jTextField27.setText(c_id);  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    ADD_CUSTOMER.setVisible(true);
}
```

Coding On Car Available Button:-

```
private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        DBConnection db = new DBConnection();
        String company = jComboBox11.getSelectedItem().toString();
        int capacity =
        Integer.parseInt(jComboBox12.getSelectedItem().toString());
        String sql="select * from car_list where COMPANY = '"+company+"' and
SEATING_CAPACITY ='"+capacity+"' and STATUS = 'AVAILABLE'";
        db.pstmt = db.con.prepareStatement(sql);
        db.rs = db.pstmt.executeQuery();
        jTable5.setModel(DbUtils.resultSetToTableModel(db.rs));
    }
    catch(Exception e)
    {
```

REPORT ON VEHICLE RENTING SERVICE

```
    e.printStackTrace();  
}  
  
}
```

Coding On Submit Button: -

```
private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
        DBConnection db = new DBConnection();  
        int id = Integer.parseInt(jTextField27.getText());  
        String name = jTextField28.getText();  
        String gender = "";  
        if(jRadioButton7.isSelected())  
        {  
            gender="MALE";  
        }  
        if(jRadioButton8.isSelected())  
        {  
            gender="FEMALE";  
        }  
        String address = jTextArea4.getText();  
        String use = jComboBox10.getSelectedItem().toString();  
        String bookd = jTextField8.getText();  
        String returnd = jTextField9.getText();  
    }
```

REPORT ON VEHICLE RENTING SERVICE

```
String mob = jTextField31.getText();
String carnum = jTextField26.getText();
int did = Integer.parseInt(jComboBox13.getSelectedItem().toString());
int rent = Integer.parseInt(jTextField34.getText());
String actualdate = jTextField9.getText();
String status = "ACTIVE";
String sql = "insert into customers_list
values(""+id+"','"+name+"','"+gender+"','"+address+"','"+use+"','"+bookd+"','"+r
eturnd+"','"+mob+"','"+carnum+"','"+did+"','"+rent+"','"+
actualdate+"','"+status+"')";
db(stmt = db.con.createStatement());
int i = db(stmt.executeUpdate(sql);
String sql8 ="update car_list set STATUS='NOT AVAILABLE' where
CAR_NUM='"+carnum+"'";
db(stmt = db.con.createStatement());
int k = db(stmt.executeUpdate(sql8);
if(i>0)
{
    JOptionPane.showMessageDialog(this,"Customer Added Succesfully!");
}
else
{
    JOptionPane.showMessageDialog(this,"Something Went Wrong!");
}
```

```
        }  
    }  
    catch(Exception e)  
    {  
        e.printStackTrace();  
    }  
  
}
```

Coding On Cancel Button: -

```
private void jButton21ActionPerformed(java.awt.event.ActionEvent evt) {  
    ADD_CUSTOMER.setVisible(false);  
}  
  
}
```

Modify Customer: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {  
    MODIFY_CUSTOMER.setVisible(true);  
  
}  
  
}
```

Coding On Search By ID Button: -

REPORT ON VEHICLE RENTING SERVICE

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {  
    //MODIFY CUSTOMER  
  
    try  
  
    {  
  
        int id = Integer.parseInt(jTextField12.getText());  
  
        DBConnection db = new DBConnection();  
  
        String sql1="select * from customers_list where C_id = '"+id+"'";  
  
        db.stmt = db.con.createStatement();  
  
        db.rs = db.stmt.executeQuery(sql1);  
  
        if(db.rs.next())  
  
        {  
  
            jTextField12.setText(db.rs.getString(1));  
  
            jTextField13.setText(db.rs.getString(2));  
  
            String gen = db.rs.getString(3);  
  
            if("MALE".equals(gen))  
  
            {  
  
                jRadioButton1.setSelected(true);  
  
            }  
  
            else  
  
            {  
  
                jRadioButton2.setSelected(true);  
  
            }  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
jTextArea1.setText(db.rs.getString(4));  
jTextField14.setText(db.rs.getString(8));  
jTextField10.setText(db.rs.getString(6));  
jTextField11.setText(db.rs.getString(7));  
jTextField37.setText(db.rs.getString(9));  
jComboBox14.setSelectedItem(db.rs.getString(9));  
  
}  
  
else  
{  
    JOptionPane.showMessageDialog(this, "No Record Found!");  
}  
  
}  
  
catch(Exception e)  
{  
    e.printStackTrace();  
}  
  
}
```

Coding On Search By Name: -

REPORT ON VEHICLE RENTING SERVICE

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        String name = jTextField13.getText().toString();  
        DBConnection db = new DBConnection();  
        String sql1="select * from customers_list where Name = '"+name+"'";  
        db.stmt = db.con.createStatement();  
        db.rs = db.stmt.executeQuery(sql1);  
        if(db.rs.next())  
        {  
            jTextField12.setText(db.rs.getString(1));  
            jTextField13.setText(db.rs.getString(2));  
            String gen = db.rs.getString(3);  
            if(gen=="MALE")  
            {  
                jRadioButton1.setSelected(true);  
            }  
  
            if(gen=="FEMALE")  
            {  
                jRadioButton2.setSelected(true);  
            }  
            jTextArea1.setText(db.rs.getString(4));  
            jTextField14.setText(db.rs.getString(5));  
        }  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
jTextField10.setText(db.rs.getString(6));
jTextField11.setText(db.rs.getString(7));
jTextField37.setText(db.rs.getString(8));
jComboBox14.setSelectedItem(db.rs.getString(9));

}

else
{
    JOptionPane.showMessageDialog(this, "No Record Found!");
}

}

catch(Exception e)
{
    e.printStackTrace();
}

}
```

Coding On Update Button: -

```
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
```

REPORT ON VEHICLE RENTING SERVICE

```
DBConnection db = new DBConnection();

int id = Integer.parseInt(jTextField12.getText());

String name = jTextField13.getText();

String gender="";

if(jRadioButton1.isSelected())

{

    gender="MALE";

}

if(jRadioButton2.isSelected())

{

    gender="FEMALE";

}

String address = jTextArea1.getText();

String mob = jTextField14.getText();

String bdate = jTextField10.getText();

String rdate = jTextField11.getText();

String carnum = jTextField37.getText();

int did = Integer.parseInt(jComboBox14.getSelectedItem().toString());

String sql = "update customers_list set Name='"+name+"',

Gender='"+gender+"', Address='"+address+"', Mobile ='"+mob+"',

Book_Date='"+bdate+"', Return_Date='"+rdate+"', Car_num='"+carnum+"'";

db.stmt = db.con.createStatement();
```

REPORT ON VEHICLE RENTING SERVICE

```
int i = db.stmt.executeUpdate(sql);
if(i>0)
{
    JOptionPane.showMessageDialog(this,"Record Successfully Updated!");
}
else
{
    JOptionPane.showMessageDialog(this,"Records Updation Failed!");
}

}
catch(Exception e)
{
    e.printStackTrace();
}
}
```

Coding On Delete Button :-

```
private void
jButton11ActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        DBConnection db = new DBConnection();
```

REPORT ON VEHICLE RENTING SERVICE

```
int id = Integer.parseInt(jTextField12.getText());

String sql = "delete from customers_list where c_id ='" + id + "'";

db(stmt = db.con.createStatement());

int i = db(stmt.executeUpdate(sql);

if(i>0)

{

    JOptionPane.showMessageDialog(this,"Record Successfully Deleted !");

}

else

{

    JOptionPane.showMessageDialog(this,"Failed To Delete Record !");

}

catch(Exception e)

{

    e.printStackTrace();

}

}
```

Return Car :-

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    RETURN_CAR.setVisible(true);  
}
```

Coding On Search By ID Button: -

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
  
        int id = Integer.parseInt(jTextField5.getText());  
  
        DBConnection db = new DBConnection();  
  
        String sql = "select * from customers_list where C_ID = '"+id+"'";  
  
        db.stmt = db.con.createStatement();  
  
        db.rs = db.stmt.executeQuery(sql);  
  
        if(db.rs.next())  
        {  
  
            jTextField5.setText(db.rs.getString(1));  
  
            jTextField6.setText(db.rs.getString(2));  
  
            jTextArea5.setText(db.rs.getString(4));  
  
            jTextField18.setText(db.rs.getString(6));  
  
            jTextField19.setText(db.rs.getString(7));  
  
            jTextField39.setText(db.rs.getString(9));  
  
            jTextField38.setText(db.rs.getString(11));  
  
            jTextField7.setText(db.rs.getString(8));  
        }  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
        }  
  
    else  
  
    {  
  
        JOptionPane.showMessageDialog(this,"No Record Found!");  
  
    }  
  
}  
  
catch(Exception e)  
  
{  
  
    e.printStackTrace();  
  
}  
  
}
```

Coding On Search By Name Button: -

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
  
    {  
  
        String name = jTextField6.getText();  
  
        DBConnection db = new DBConnection();  
  
        String sql = "select * from customers_list where Name = '"+name+"'";  
  
        db.stmt = db.con.createStatement();  
  
        db.rs = db.stmt.executeQuery(sql);  
  
        if(db.rs.next())  
  
        {  
  
            jTextField5.setText(db.rs.getString(1));  
  
            jTextField6.setText(db.rs.getString(2));  
  
        }  
  
    }  
  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
jTextArea5.setText(db.rs.getString(4));  
jTextField18.setText(db.rs.getString(5));  
jTextField19.setText(db.rs.getString(6));  
jTextField39.setText(db.rs.getString(9));  
jTextField38.setText(db.rs.getString(11));  
jTextField7.setText(db.rs.getString(8));  
}  
  
else  
{  
    JOptionPane.showMessageDialog(this,"No Record Found!");  
}  
}  
  
catch(Exception e)  
{  
    e.printStackTrace();  
}  
}
```

Coding On Returned Button: -

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
        DBConnection db = new DBConnection();  
        int id = Integer.parseInt(jTextField5.getText());  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
SimpleDateFormat formdate = new SimpleDateFormat("dd-MM-yyyy");
String date = formdate.format(System.currentTimeMillis());
String sql1 = "select CAR_NUM from customers_list where C_ID =
""+id+"";
String cnumber = "";
db.pstmt = db.con.prepareStatement(sql1);
db.rs = db.pstmt.executeQuery();
while(db.rs.next())
{
    cnumber = db.rs.getString("CAR_NUM");
}
String sql2 = "select RENT from car_list where NUMBER =
""+cnumber+"";
db.pstmt = db.con.prepareStatement(sql2);
db.rs = db.pstmt.executeQuery();
int carrent = 0;
while(db.rs.next())
{
    carrent = Integer.parseInt(db.rs.getString("RENT"));
}
String sql4="SELECT DATEDIFF(RETURN_DATE,BOOK_DATE) AS
RESULT FROM customers_list WHERE C_ID = "+id+"";
db.pstmt = db.con.prepareStatement(sql4);
```

REPORT ON VEHICLE RENTING SERVICE

```
db.rs = db.pstmt.executeQuery();

int crent=0;

int fine=0;

while(db.rs.next())

{

    int datedif = Integer.parseInt(db.rs.getString("RESULT"));

    crent = datedif * carrent;

}

String sql5 = "SELECT DATEDIFF(CURDATE(),RETURN_DATE) AS
RST FROM customers_list WHERE C_ID='"+id+"'";

db.pstmt = db.con.prepareStatement(sql5);

db.rs = db.pstmt.executeQuery();

while(db.rs.next())

{

    int ddif = Integer.parseInt(db.rs.getString("RST"));

    fine = ddif * 3000;

    jTextField33.setText(Integer.toString(fine));

}

int totalrent = fine+crent;
```

REPORT ON VEHICLE RENTING SERVICE

```
jTextField32.setText(Integer.toString(totalrent));  
  
String sql3 = "update customers_list set STATUS='NOT ACTIVE',  
ACTUAL_RETURN=CURDATE() , RENT='"+totalrent+"' where  
C_ID='"+id+"'";  
  
db(stmt=db.con.createStatement());  
  
int i = db.stmt.executeUpdate(sql3);  
  
String sql8 ="update car_list set STATUS='AVAILABLE' where  
NUMBER='"+cnumber+"'";  
  
db.stmt = db.con.createStatement();  
  
int k = db.stmt.executeUpdate(sql8);  
  
if(i>0)  
{  
    JOptionPane.showMessageDialog(this,"CAR IS RETURNED  
SUCCESSFULLY !");  
}  
  
else  
{  
    JOptionPane.showMessageDialog(this,"SOMETHING WENT  
WRONG");  
}  
  
}  
  
}  
  
catch(Exception e)  
{  
    e.printStackTrace();
```

```
    }
```

```
}
```

Coding On Cancel Button :-

```
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {  
    RETURN_CAR.setVisible(false);  
}
```

Coding Print Invoice Button:-

```
private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
        INVOICE.setVisible(true);  
        int id = Integer.parseInt(jTextField5.getText());  
        DBConnection db = new DBConnection();  
        String sql = "select * from customers_list where C_ID='"+id+"'";  
        db.pstmt = db.con.prepareStatement(sql);  
        db.rs = db.pstmt.executeQuery();  
        while(db.rs.next())  
        {  
            jLabel104.setText(db.rs.getString(1));  
            jLabel103.setText(db.rs.getString(2));  
            jLabel105.setText(db.rs.getString(6));  
        }  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
jLabel106.setText(db.rs.getString(7));  
jLabel107.setText(db.rs.getString(9));  
jLabel108.setText(db.rs.getString(10));  
jLabel110.setText(db.rs.getString(11));  
  
}  
  
}  
  
catch(Exception e)  
{  
    e.printStackTrace();  
}  
  
}
```

View Cars: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent evt) {  
    VIEWCAR.setVisible(true);  
}
```

Coding On View Available Cars: -

REPORT ON VEHICLE RENTING SERVICE

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        DBConnection db = new DBConnection();  
        String Company = jComboBox2.getSelectedItem().toString();  
        String Status = jComboBox1.getSelectedItem().toString();  
        int Capacity = Integer.parseInt(jComboBox3.getSelectedItem().toString());  
        String sql = "Select * from car_list where COMPANY ="+Company+"  
and STATUS ="+Status+" and SEATING_CAPACITY = "+Capacity+"";  
        db.pstmt = db.con.prepareStatement(sql);  
        db.rs = db.pstmt.executeQuery();  
        jTable1.setModel(DbUtils.resultSetToTableModel(db.rs));  
    }  
    catch(Exception e)  
    {  
        e.printStackTrace();  
    }  
}
```

Change Password: -

Event Performed On Menu Item :-

Action Event performed(actionPerformed)

Coding: -

```
private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
```

REPORT ON VEHICLE RENTING SERVICE

```
    CHANGEPASSWORD.setVisible(true);  
}
```

Coding On Submit Button: -

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        DBConnection db = new DBConnection();  
  
        String username = jTextField1.getText().toString();  
  
        int opass = Integer.parseInt(jTextField2.getText());  
  
        int npass = Integer.parseInt(jTextField3.getText());  
  
        int cpass = Integer.parseInt(jTextField4.getText());  
  
        String sql= "select * from employee_list where Username =  
        '"+username+"' and Password ='"+opass+"'";  
  
        db.stmt = db.con.createStatement();  
  
        db.rs = db.stmt.executeQuery(sql);  
  
        if(db.rs.next())  
        {  
            if(npass==cpass)  
            {  
                String sql1 = "update employee_list set Password='"+cpass+"' w  
Username ='"+username+"'";  
  
                int j = db.stmt.executeUpdate(sql1);  
            }  
        }  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
if(j>0)
{
    JOptionPane.showMessageDialog(this,"Pasword Successfully
Changed!");
}
else
{
    JOptionPane.showMessageDialog(this,"Failed To Change
Password!");
}

}

else
{
    JOptionPane.showMessageDialog(this,"Password Does Not Match!");
}

}

else
{
    JOptionPane.showMessageDialog(this,"Username and Password not
Found!");
}

}

catch(Exception e)
{
```

REPORT ON VEHICLE RENTING SERVICE

```
    e.printStackTrace();  
}  
  
}
```

Coding On Cancel Button: -

```
private void jButton13ActionPerformed(java.awt.event.ActionEvent evt)  
{  
    Change_Password.setVisible(false);  
}
```

Admin Main Window: -

Add Employee: -

Event Performed On Menu Item:-

```
Action Event performed(actionPerformed)
```

Coding: -

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    ADD_EMPLOYEE.setVisible(true);  
}
```

Coding On Submit Button: -

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        DBConnection db = new DBConnection();  
        String name = jTextField9.getText();
```

REPORT ON VEHICLE RENTING SERVICE

```
int id = Integer.parseInt(jTextField23.getText());
String address = jTextArea1.getText();
String dob = jTextField8.getText();
String quali = jComboBox1.getSelectedItem().toString();
String mob = jTextField11.getText();
String doj = jTextField10.getText();
String username = jTextField5.getText();
int npass = Integer.parseInt(jTextField6.getText());
int cpass = Integer.parseInt(jTextField7.getText());
```

//CODE FOR RADIO BUTTON ON GENDER OPTION.

```
String gender = "";
if(jRadioButton1.isSelected())
{
    gender = "MALE";
}
else if(jRadioButton2.isSelected())
{
    gender = "FEMALE";
}
```

//CODE TO MATCH NEW PASSWORD AND CONFIRM PASSWORD.

REPORT ON VEHICLE RENTING SERVICE

```
if(cpass==npass)
{
    db(stmt = db.con.createStatement();
    String query = "insert into employee_list
values (" + id + "," + name + "," + gender + "," + address + "," + dob + "," + quali + "," +
mob + "," + username + "," + cpass + "," + doj + ")";
    int i = db(stmt.executeUpdate(query);
    if(i>0)
    {
        JOptionPane.showMessageDialog(this,"Employee Successfully Added
!");
    }
    else
    {
        JOptionPane.showMessageDialog(this,"Something Went Wrong !");
    }
}
else
{
    JOptionPane.showMessageDialog(this, "Password does not Match !");
}
}
```

REPORT ON VEHICLE RENTING SERVICE

```
        catch(Exception e)
        {
            e.printStackTrace();
        }

    }
```

Coding On Cancel Button: -

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    Add_Employee.setVisible(false);
}
```

Modify Employee: -

Event Performed On Menu Item:-

```
Action Event performed(actionPerformed)
```

Coding:- -

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    MODIFY_EMPLOYEE.setVisible(true);
}
```

Coding On Search Button: -

```
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
```

REPORT ON VEHICLE RENTING SERVICE

```
DBConnection db = new DBConnection();

int id = Integer.parseInt(jTextField12.getText());

String sql = "select * from employee_list where id='"+id+"'";

db.stmt=db.con.createStatement();

db.rs = db.stmt.executeQuery(sql);

if(db.rs.next())

{

    jTextField13.setText(db.rs.getString(2));

    String gen = db.rs.getString(3);

    if("MALE".equals(gen))

    {

        jRadioButton3.setSelected(true);

    }

    else

    {

        jRadioButton4.setSelected(true);

    }

    jTextArea2.setText(db.rs.getString(4));

    String datevalue = db.rs.getString(5);

    jComboBox2.setSelectedItem(db.rs.getString(6));

    jTextField15.setText(db.rs.getString(7));

    String datevalue1 = db.rs.getString(10);

    jTextField14.setText(db.rs.getString(5));

    jTextField16.setText(db.rs.getString(10));

}
```

REPORT ON VEHICLE RENTING SERVICE

```
        }  
  
    else  
  
    {  
  
        JOptionPane.showMessageDialog(this,"Data Not Found !");  
  
    }  
  
}  
  
catch(Exception e)  
  
{  
  
    e.printStackTrace();  
  
}  
  
}
```

Coding On Submit Button: -

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
  
    {  
  
        DBConnection db = new DBConnection();  
  
        String name = jTextField13.getText().toString();  
  
        int id = Integer.parseInt(jTextField12.getText());  
  
        String address = jTextArea2.getText().toString();  
  
        String dob = jTextField14.getText();  
  
        String quali = jComboBox2.getSelectedItem().toString();  
  
        String mob = jTextField15.getText();  
  
        String doj = jTextField16.getText();  
  
    }  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
String gender="";  
  
if(jRadioButton3.isSelected())  
{  
    gender ="MALE";  
}  
  
else if(jRadioButton4.isSelected())  
{  
    gender ="FEMALE";  
}  
  
String sql ="update employee_list set NAME ="+name+", GENDER =  
"+gender+", ADDRESS = "+address+", DOB='"+dob+"',QUALIFICATION  
='"+quali+"',MOBILE =" +mob+",DOJ='"+doj+"' where ID='"+id+"'";  
  
db.stmt = db.con.createStatement();  
  
int i = db.stmt.executeUpdate(sql);  
  
  
if(i>0)  
{  
    JOptionPane.showMessageDialog(this,"Employee Details Successfully  
Updated ! ");  
}  
  
else  
{  
    JOptionPane.showMessageDialog(this,"Something Went Wrong !");  
}
```

REPORT ON VEHICLE RENTING SERVICE

```
        }  
    catch(Exception e)  
    {  
        e.printStackTrace();  
    }  
  
}
```

Coding On Cancel Button: -

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {  
    MODIFY_EMPLOYEE.setVisible(false);  
}
```

Employee List: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
public void displayemployeelist()  
{  
    try  
    {  
        DBConnection db = new DBConnection();
```

REPORT ON VEHICLE RENTING SERVICE

```
String sql = "select * from employee_list";
db.pstmt = db.con.prepareStatement(sql);
db.rs = db.pstmt.executeQuery();
jTable1.setModel(DbUtils.resultSetToTableModel(db.rs));

}

catch(Exception e)

{
    e.printStackTrace();
}
```

Driver List: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding: -

```
public void displayDriverlist()

{
    try
    {
        DBConnection db = new DBConnection();
        String sql = "select * from Drivers_list ";
        db.pstmt = db.con.prepareStatement(sql);
        db.rs = db.pstmt.executeQuery();
        jTable2.setModel(DbUtils.resultSetToTableModel(db.rs));
    }
}
```

REPORT ON VEHICLE RENTING SERVICE

```
    }  
  
    catch(Exception e)  
  
    {  
  
        e.printStackTrace();  
  
    }  
  
}
```

View Cars: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding On View Available Cars Button: -

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
  
    {  
  
        DBConnection db = new DBConnection();  
  
        String comp = jComboBox4.getSelectedItem().toString();  
  
        int capacity = Integer.parseInt(jComboBox5.getSelectedItem().toString());  
  
        String status = jComboBox3.getSelectedItem().toString();  
  
        String sql = "select * from car_list where COMPANY='"+comp+"' AND  
SEATING_CAPACITY='"+capacity+"' AND STATUS='"+status+"'";  
  
        db.pstmt = db.con.prepareStatement(sql);  
  
        db.rs = db.pstmt.executeQuery();  
  
        jTable4.setModel(DbUtils.resultSetToTableModel(db.rs));  
  
    }  
}
```

```
        catch(Exception e)  
    }
```

Add Company: -

Event Performed On Menu Item:-

Action Event performed(actionPerformed)

Coding On Submit Button: -

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
        DBConnection db = new DBConnection();  
  
        String comp = jTextField17.getText().toString();  
  
        String sql = "insert into carcompany values ('"+ comp +"')";  
  
        db.stmt = db.con.createStatement();  
  
        int i = db.stmt.executeUpdate(sql);  
  
        if(i>0)  
        {  
            JOptionPane.showMessageDialog(this,"Company Added Successfully  
!");  
        }  
        else  
        {  
            JOptionPane.showMessageDialog(this,"Failed To Add Company!");  
        }  
    }
```

REPORT ON VEHICLE RENTING SERVICE

```
        }  
    catch(Exception e)  
    {  
        e.printStackTrace();  
    }  
  
}
```

Coding On Cancel Button: -

```
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {  
    ADDCOMPANY.setVisible(false);  
}  
}
```

Add Car: -

Event Performed On Menu Item:-

```
Action Event performed(actionPerformed)
```

Coding: -

```
private void  
jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    ADD_CAR.setVisible(true);  
}  
}
```

Coding On Submit Button: -

```
private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
    {  
    }
```

REPORT ON VEHICLE RENTING SERVICE

```
DBConnection db = new DBConnection();

String model = jTextField18.getText();

String company = jComboBox9.getSelectedItem().toString();

String Number = jTextField19.getText();

String mfg = jTextField20.getText();

String useas = jComboBox10.getSelectedItem().toString();

String capacity = jComboBox11.getSelectedItem().toString();

String type = jComboBox7.getSelectedItem().toString();

String engine = jTextField2.getText();

String rent = jTextField22.getText();

String status = "AVAILABLE";

String sql = "insert into car_list
values("+company+","+model+","+Number+","+mfg+","+"
useas+","+capacity+","+type+","+engine+","+rent+","+status+");"

db(stmt = db.con.createStatement());

int i = db(stmt.executeUpdate(sql);

if(i>0)

{

    JOptionPane.showMessageDialog(this,"New Car Is Added Successfully

!");

}

else

{

    JOptionPane.showMessageDialog(this,"Car Registration Failed !");

}
```

```
    }  
  
    catch(Exception e)  
  
    {  
  
        e.printStackTrace();  
  
    }  

```

Coding On Cancel Button: -

```
private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    ADD_CAR.setVisible(false);  

```

Change Password: -

Event Performed On Menu Item:-

```
Action Event performed(actionPerformed)
```

Coding: -

```
private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    CHANGE_PASS.setVisible(true);  
  
}
```

Coding On Submit Button: -

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try  
  
    {  
  
        DBConnection db = new DBConnection();  
  
        String username = jTextField1.getText();  
  
        int opass = Integer.parseInt(jTextField2.getText());  

```

REPORT ON VEHICLE RENTING SERVICE

```
int npass = Integer.parseInt(jTextField3.getText());
int cpass = Integer.parseInt(jTextField4.getText());

String sql = "Select COUNT(*) from loginadmin where username=
""+username+" and password ="+opass+"";

db(stmt = db.con.createStatement());
db.rs = db.stmt.executeQuery(sql);

if(db.rs.next())
{
    if(cpass == npass)
    {
        String sql1 = "update loginadmin set password = "+cpass+" where
username = "+username+"";

        int j = db.stmt.executeUpdate(sql1);

        if(j>0)
        {
            JOptionPane.showMessageDialog(this,"Password Changed
Successfully !");
        }
        else
        {
            JOptionPane.showMessageDialog(this,"Unable To Change
Password !");
        }
    }
}
```

REPORT ON VEHICLE RENTING SERVICE

```
        else
        {
            JOptionPane.showMessageDialog(this,"Password Confirmation Does
Not Match");
        }
    }
else
{
    JOptionPane.showMessageDialog(this,"You Entered Wrong Username
and Password!");
}
}
}
catch(Exception e)
{
    e.printStackTrace()
}
}
```

Coding On Cancel Button: -

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
    CHANGE_PASS.setVisible(false);
}
```

7.DESIGN

SCREENSHOTS

Splash Screen: -

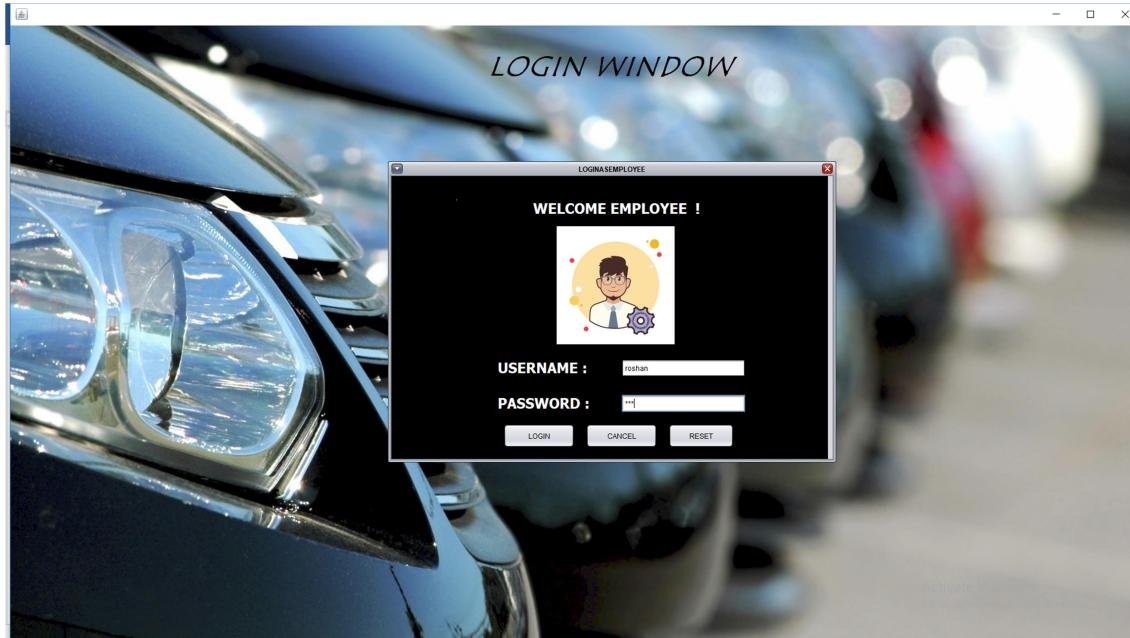


Login Window: -

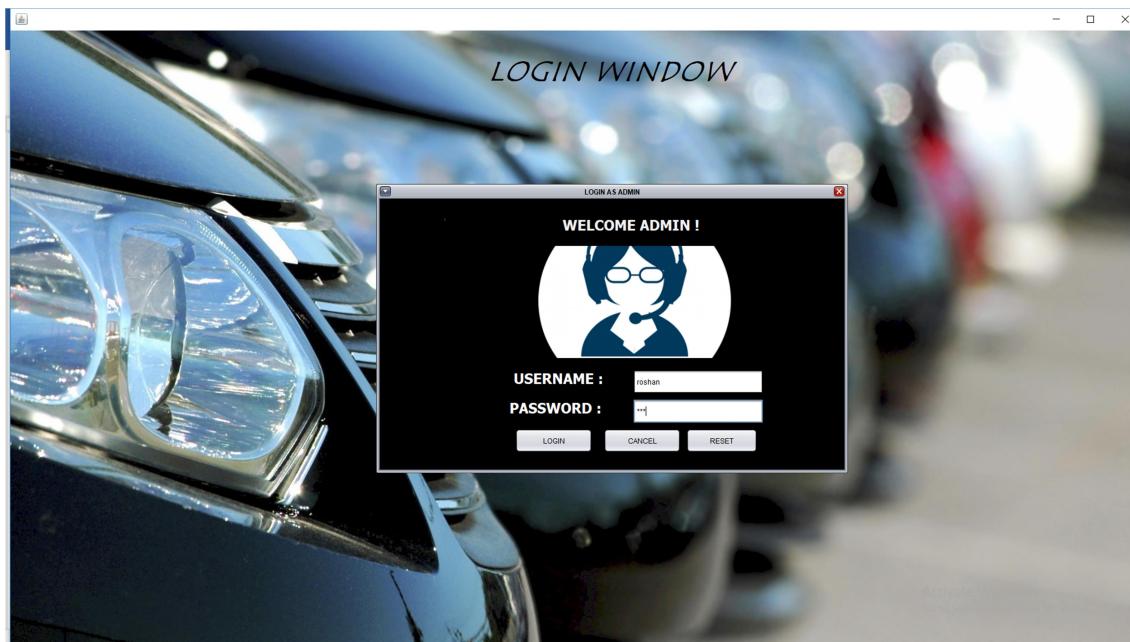


REPORT ON VEHICLE RENTING SERVICE

Employee Login: -

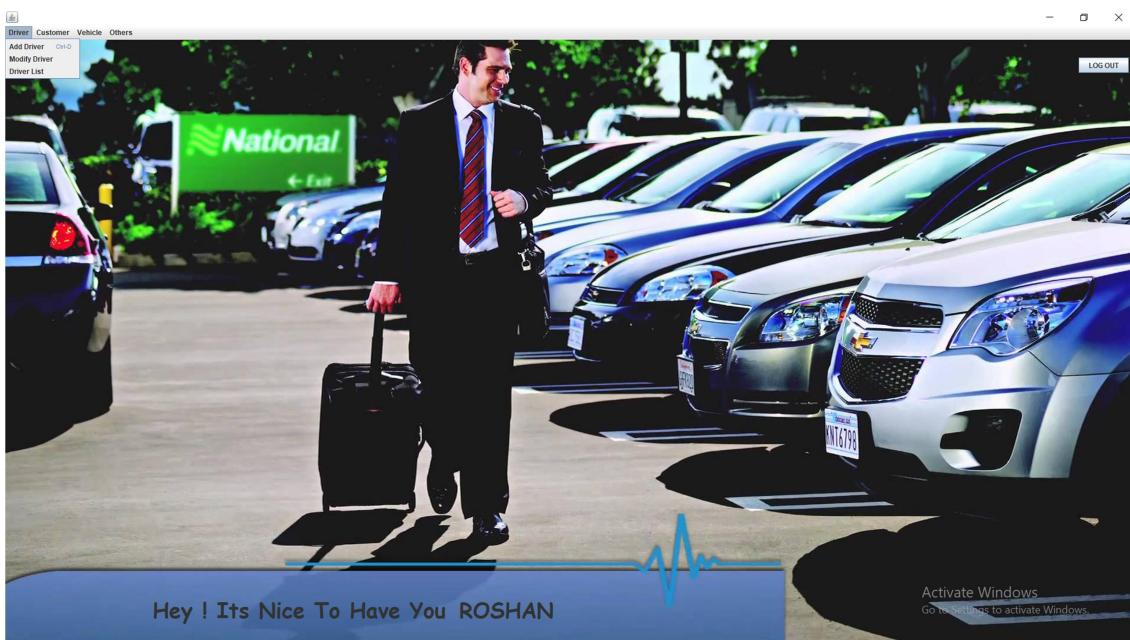
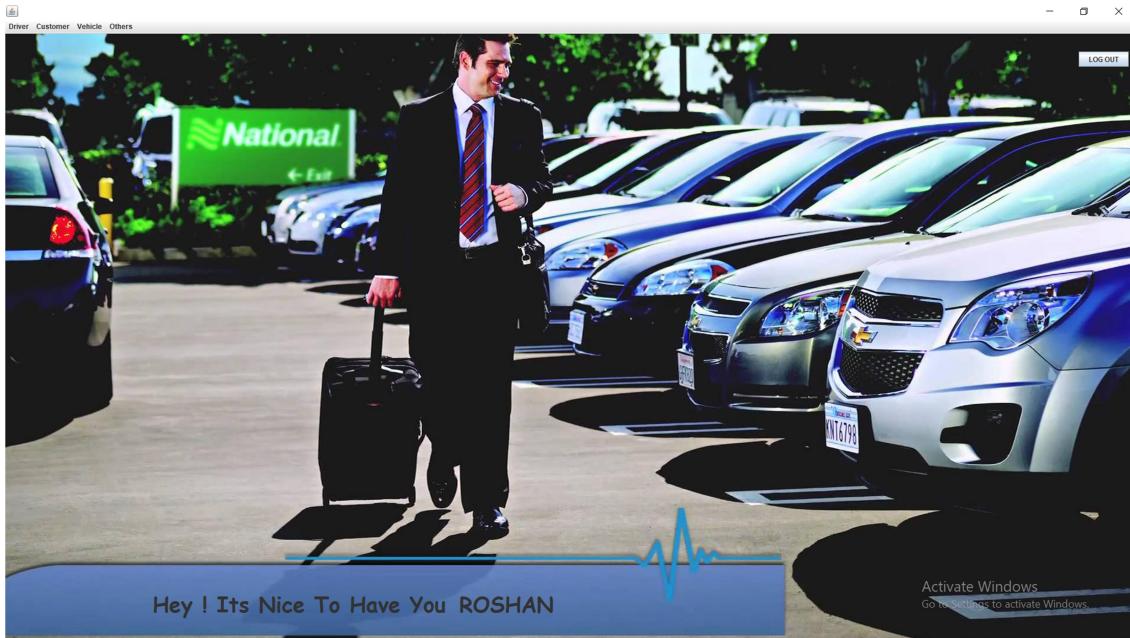


Admin Login: -

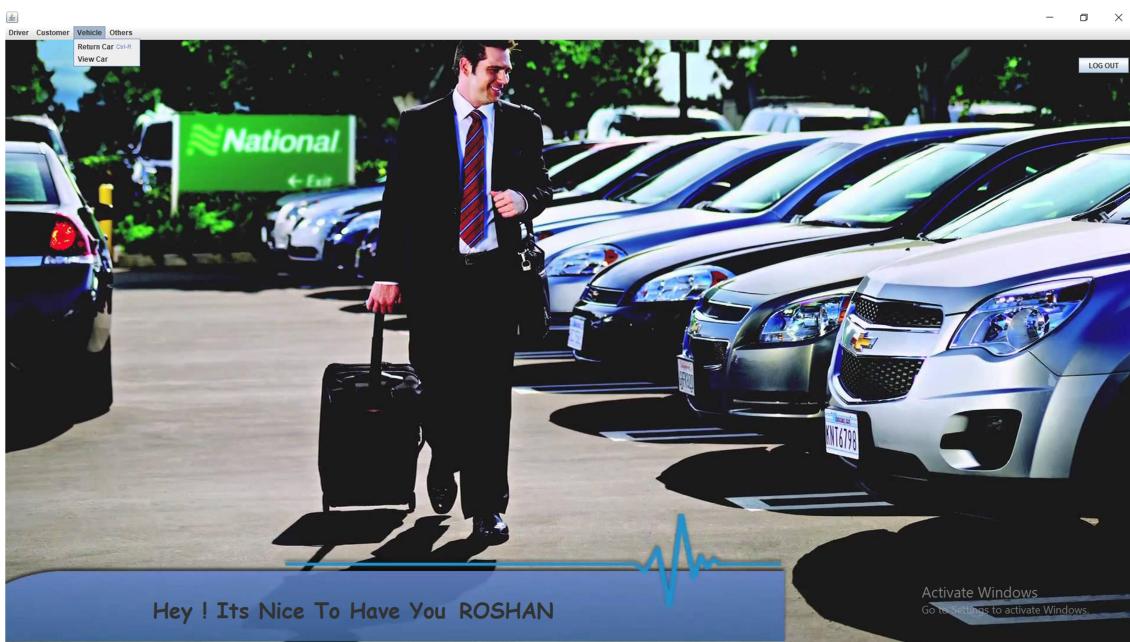


REPORT ON VEHICLE RENTING SERVICE

Employee Window: -



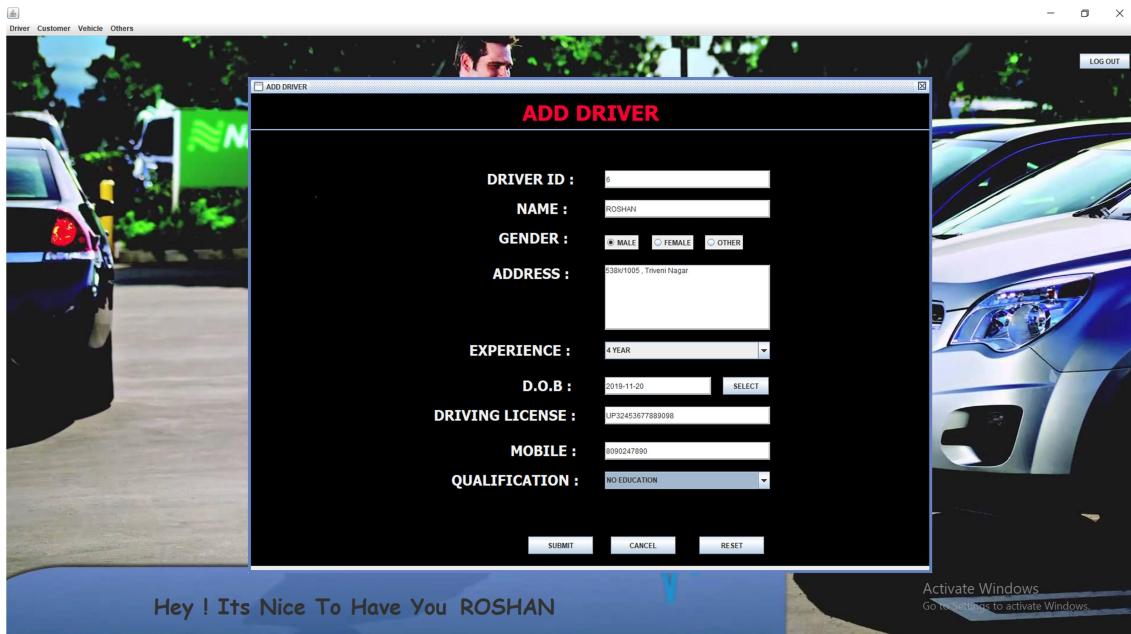
REPORT ON VEHICLE RENTING SERVICE



REPORT ON VEHICLE RENTING SERVICE

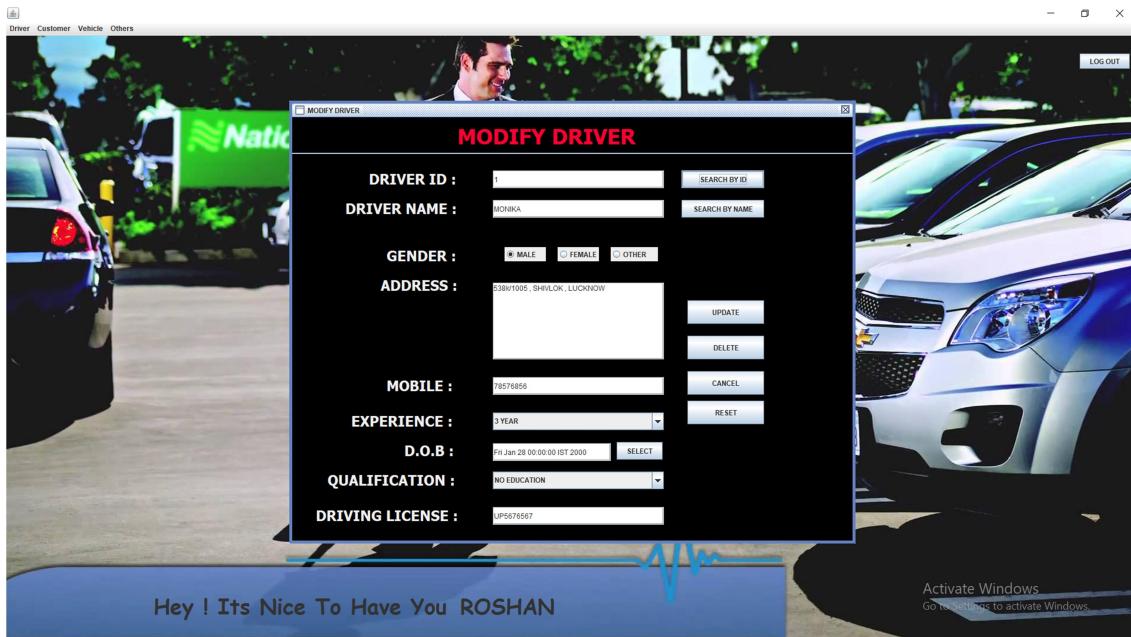


Add Driver: -



REPORT ON VEHICLE RENTING SERVICE

Modify Driver: -

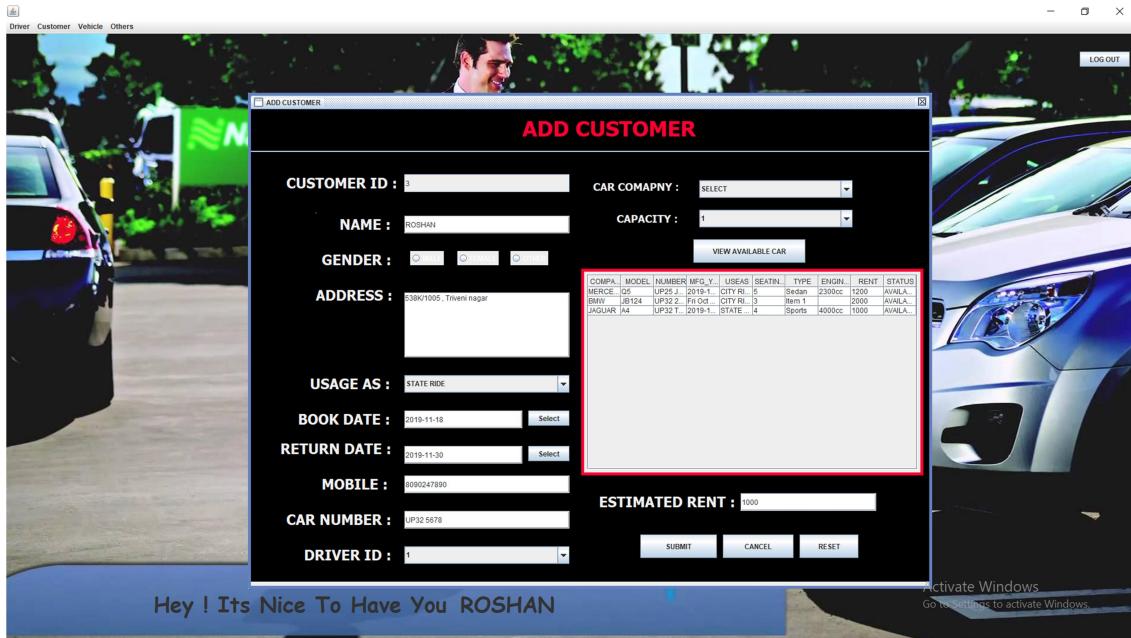


Driver List: -

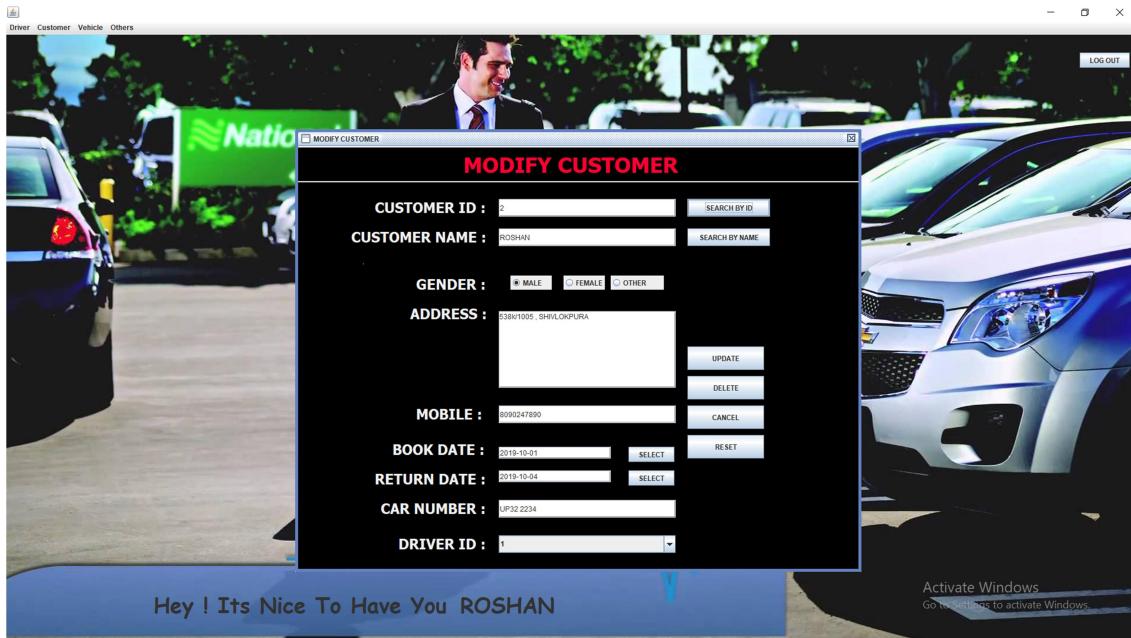
| ID_ID | NAME | GENDER | ADDRESS | EXPERIENCE | D.O.B. | LICENSE_NUM | QUALIFICATION | MOBILE |
|-------|---------|--------|-------------------|------------|---------------------|---------------|---------------|------------|
| 1 | MONIKA | MALE | 53891005 SHIVL | 3 YEAR | Fri Jan 28 00:00:00 | UP9676667 | NO EDUCATION | 78578856 |
| 2 | ROSHAN | MALE | 53891005 SHIVL | 5 YEAR | Sun Feb 07 00:00:00 | UP5689 | INTERMEDIATE | 90908990 |
| 3 | PRABHAV | MALE | 53891005 SHIVL | 3 YEAR | Mon Mar 08 00:00:00 | UP98765432 | HIGH SCHOOL | 808012345 |
| 4 | GAURAV | MALE | Thaneur Ganj, Ch. | 4 YEAR | 2019-05-14 | UP7201980054 | INTERMEDIATE | 9935191235 |
| 5 | YOGI | MALE | (fthjdhgfhfhfy) | 4 YEAR | 2019-11-06 | UP32 u9987690 | HIGH SCHOOL | 8090247890 |

REPORT ON VEHICLE RENTING SERVICE

Add Customer: -

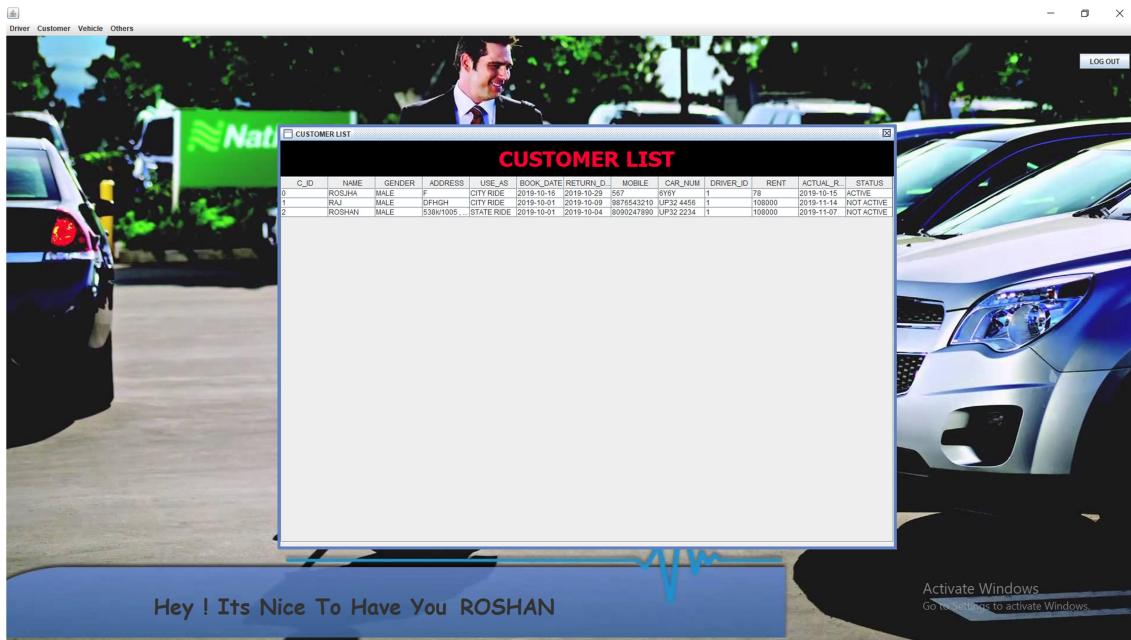


Modify Customer: -

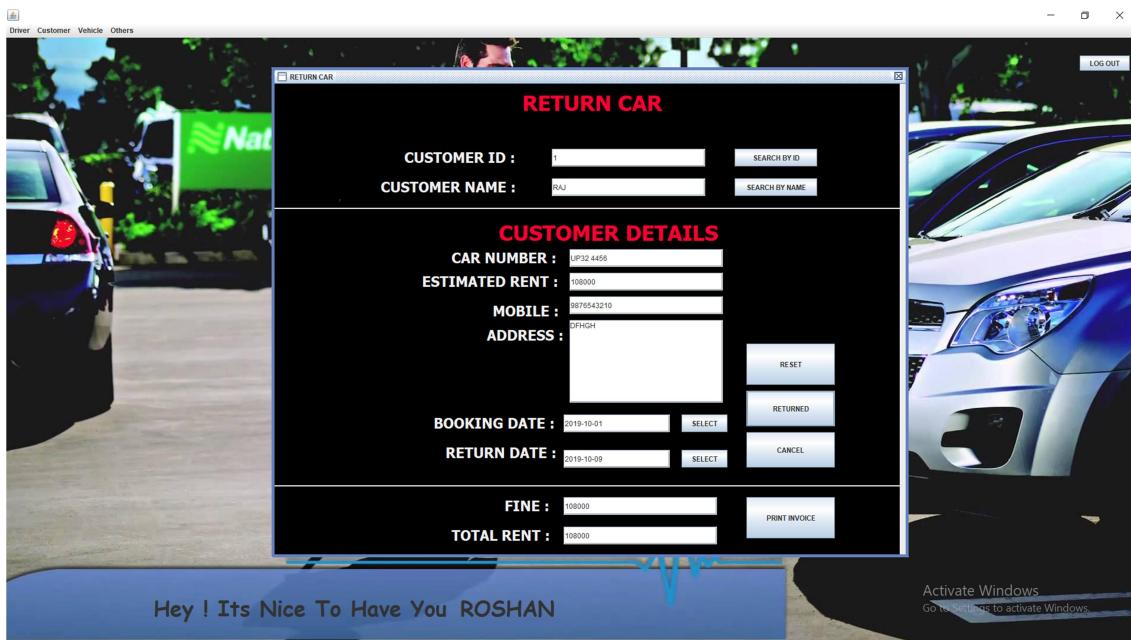


REPORT ON VEHICLE RENTING SERVICE

Customer List: -

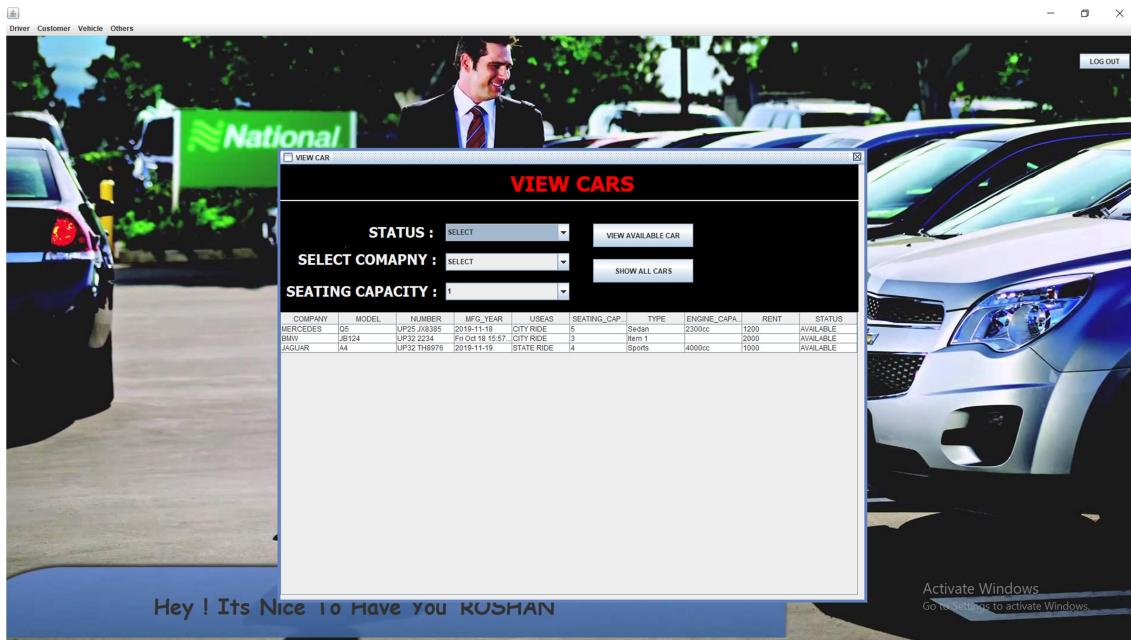


Return Car: -

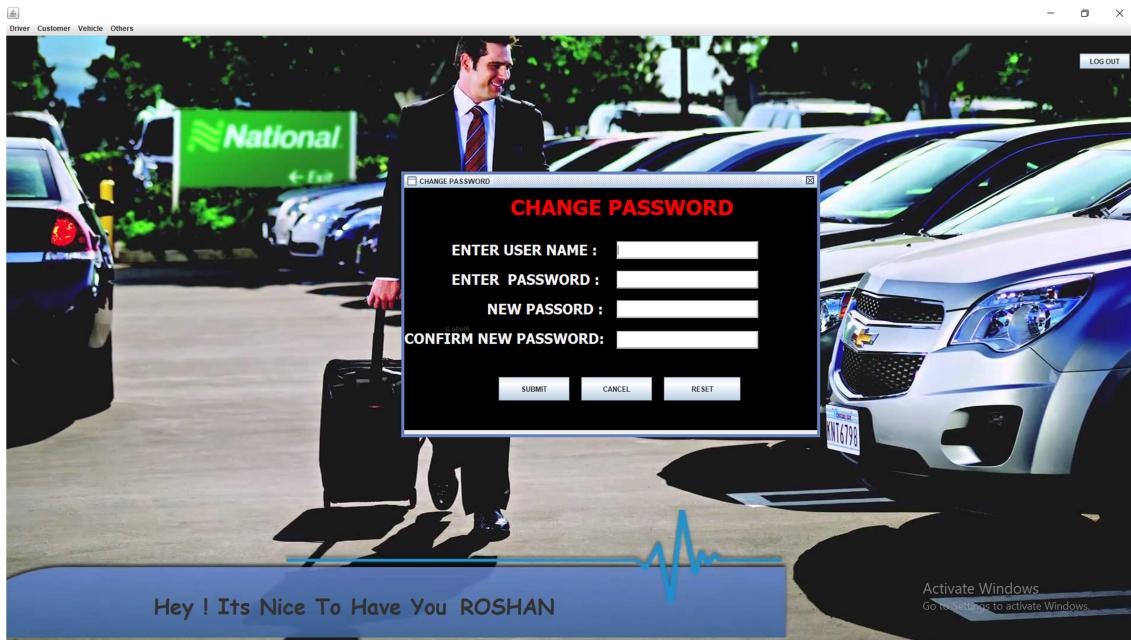


REPORT ON VEHICLE RENTING SERVICE

View Cars: -

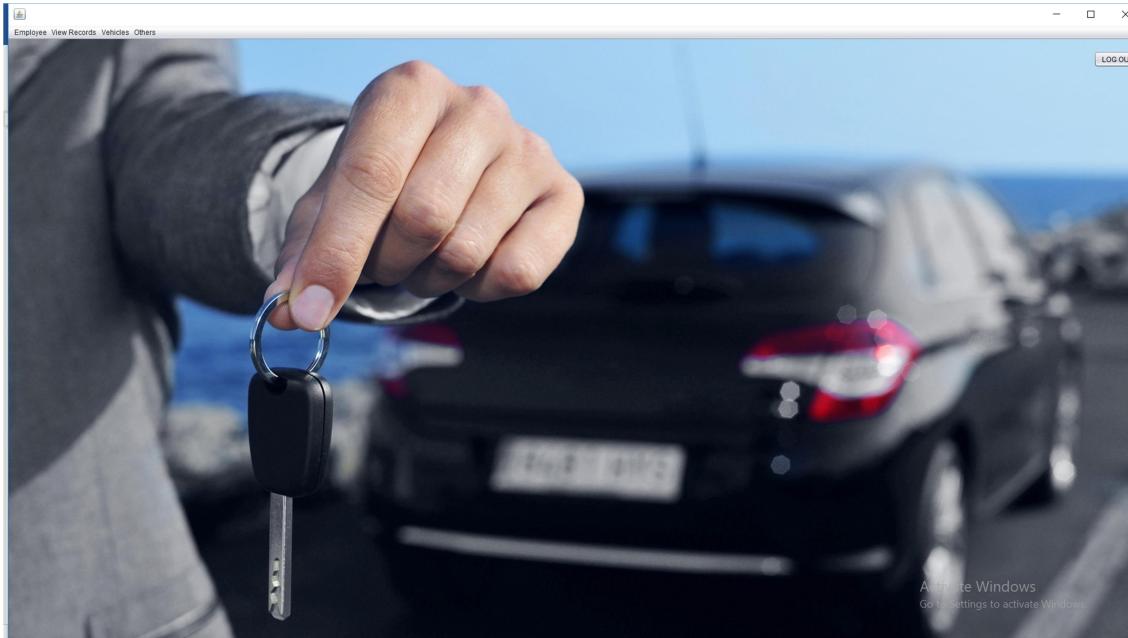


Change Password: -

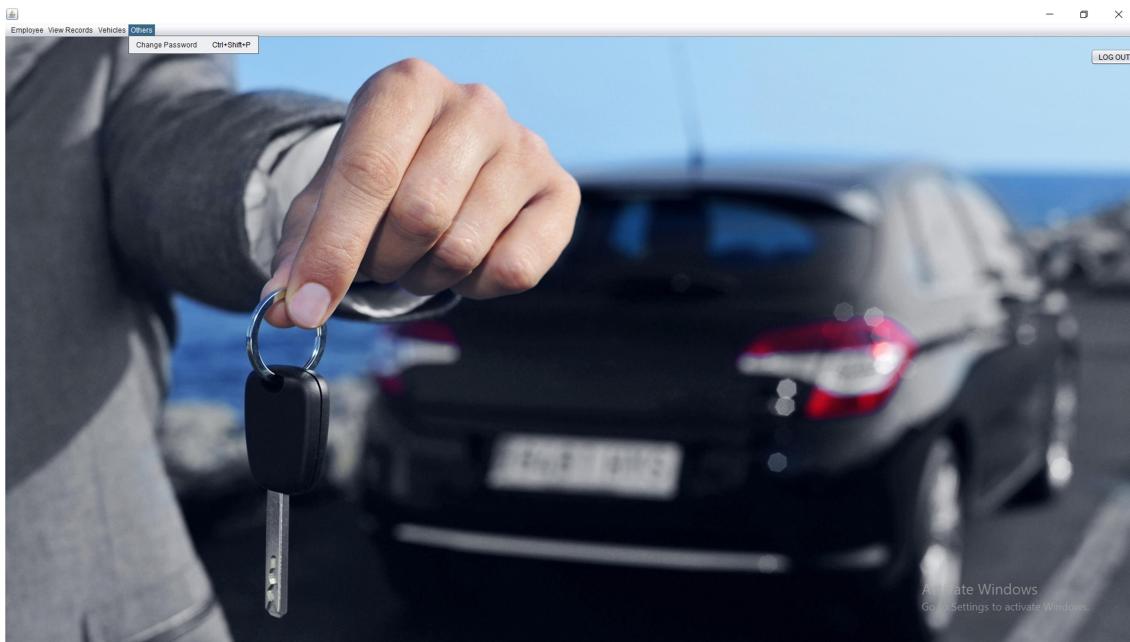


REPORT ON VEHICLE RENTING SERVICE

Admin Window: -



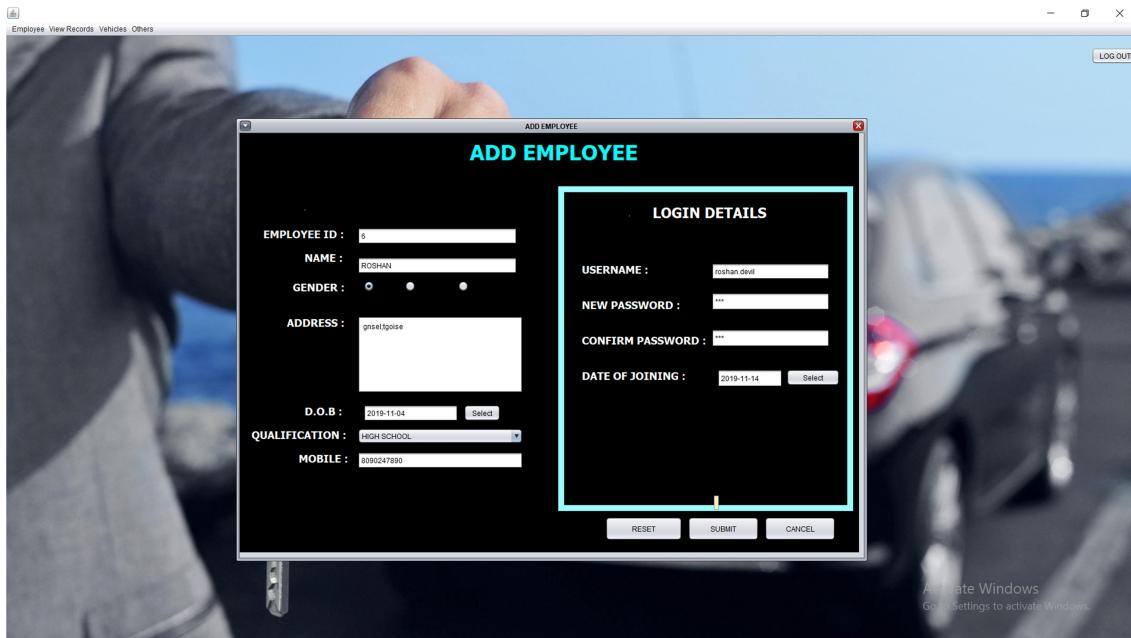
REPORT ON VEHICLE RENTING SERVICE



REPORT ON VEHICLE RENTING SERVICE

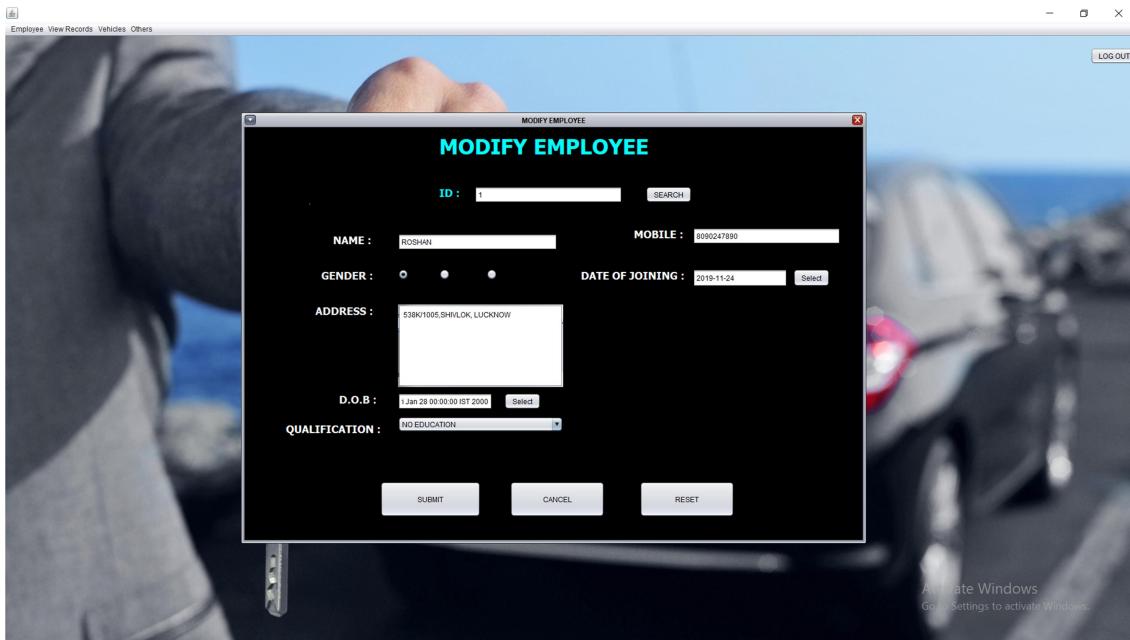


Add Employee: -



REPORT ON VEHICLE RENTING SERVICE

Modify Employee: -



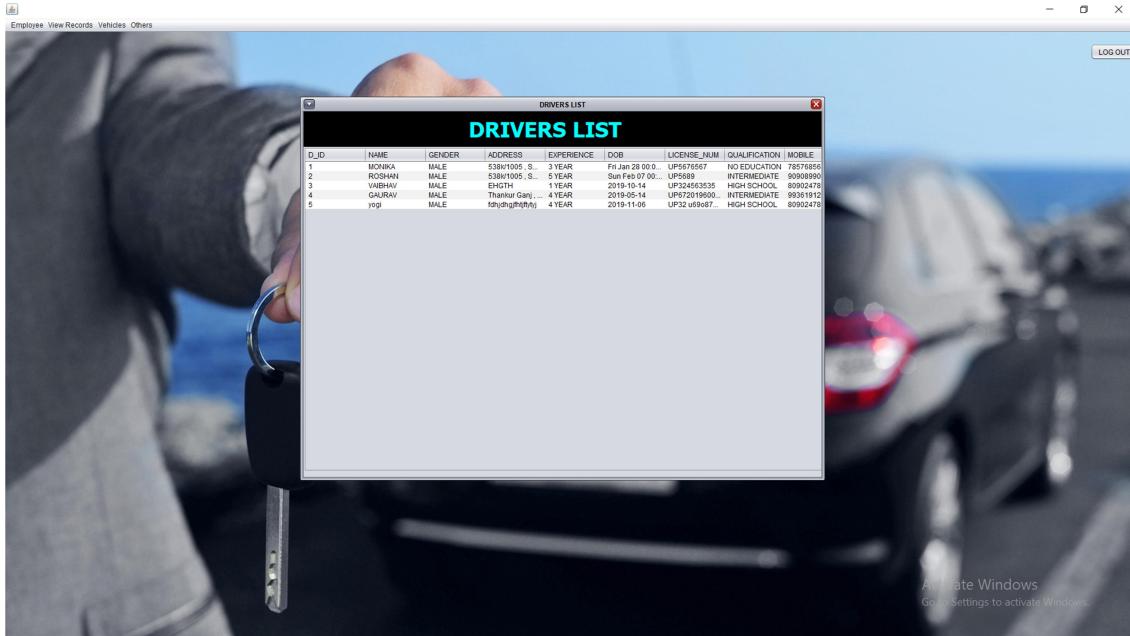
Employee List: -

A screenshot of a Windows application window titled "EMPLOYEE LIST". The window displays a table of employee records:

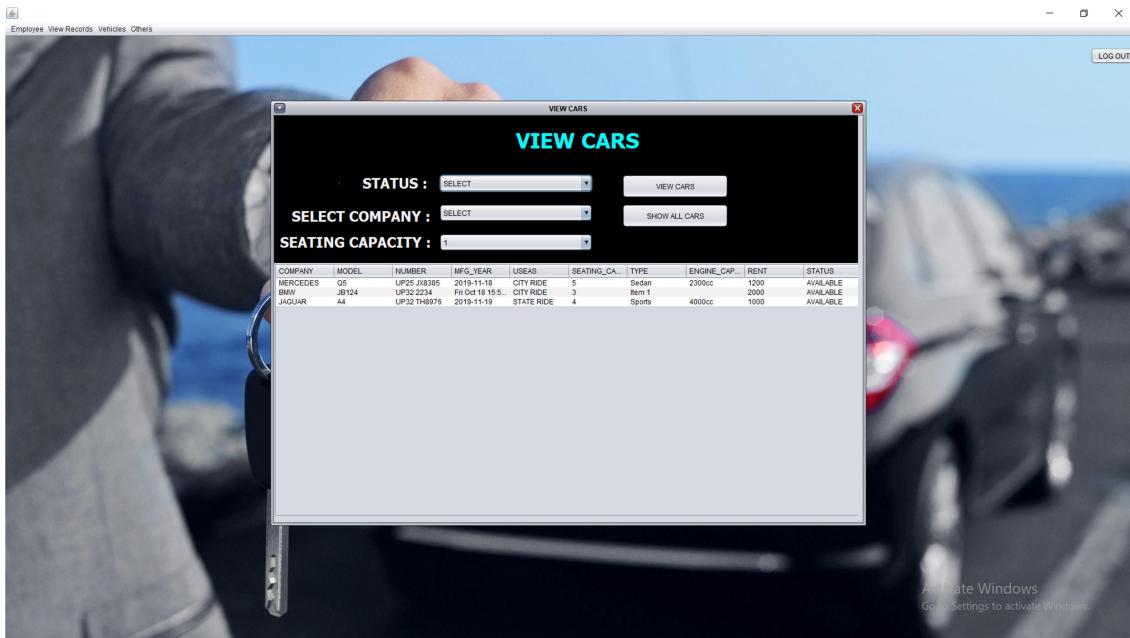
| ID | NAME | GENGER | ADDRESS | DOB | QUALIFICATI. | MOBILE | USERNAME | PASSWORD | DOJ |
|----|---------|--------|----------------|-----------------------------------|--------------|------------|-------------|----------|------------|
| 1 | ROSHAN | MALE | 538K1005,S. | Fr Jan 28 00:00:00 IST 2019-10-01 | ND EDUCATI. | 8990247890 | roshan.dell | 123 | 2019-11-24 |
| 2 | ROSHAN | MALE | CFHOTYH | SELECT | 453453452 | 2019-10-24 | SUNIL_SK | 1234 | |
| 3 | suniti | MALE | GOVT NAGA | 1990-01-01 | HIGH SCHOOL | 9938191235 | KOMAL_69 | 8099 | 2019-10-24 |
| 4 | SHAALU | FEMALE | GOMITI NAGA | 2019-10-21 | HIGH SCHOOL | 9938191235 | KOMAL_69 | 8099 | 2019-10-24 |
| 5 | rishabh | MALE | 538K1005.ra... | 2019-11-04 | INTERMEDIA | 8990247890 | rishab.ak | 8099 | 2019-11-07 |

REPORT ON VEHICLE RENTING SERVICE

Driver List: -

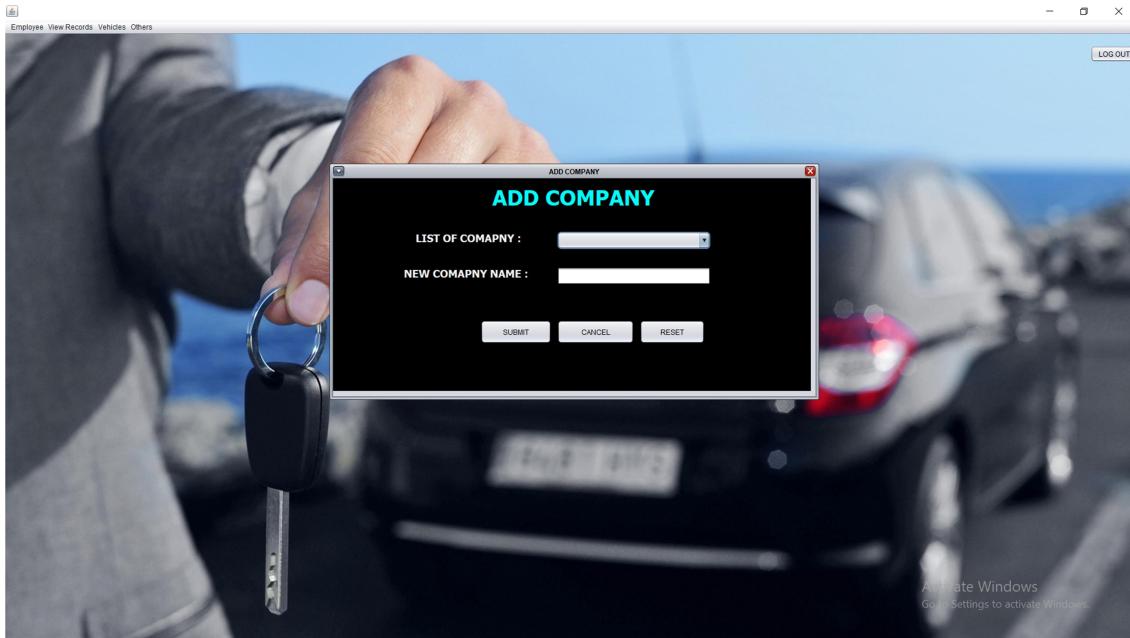


View Cars: -

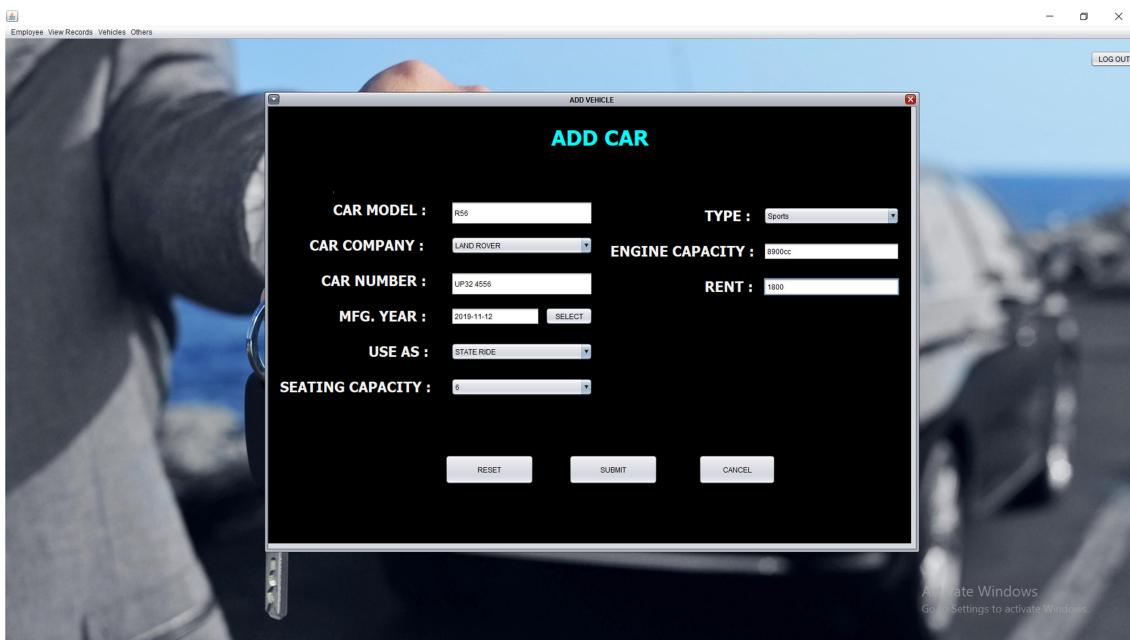


REPORT ON VEHICLE RENTING SERVICE

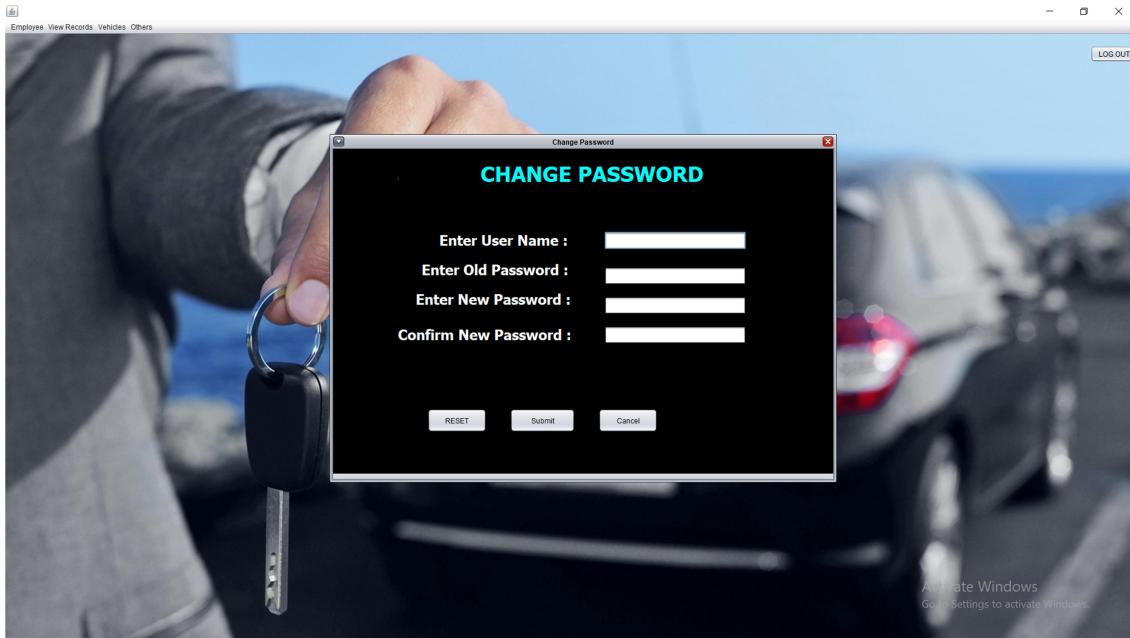
Add Company: -



Add Car: -



Change Password: -



8. TESTING

System Testing: -

Black box testing method was used for system testing. The black box testing usually demonstrates that software functions are operational; that the input is properly accepted and the output is correctly produced; and that integrity of external information (databases) is maintained.

Why testing is done

- Testing is the process of running a system with the intention of finding errors.
- Testing enhances the integrity of a system by detecting deviations in design and errors in the system.
- Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system.
- Testing also adds value to the product by confirming to the user requirements.

Causes of Errors

- The most common causes of errors in a software system are:
- **Communication gap between the developer and the business decision maker:** A communication gap between the developer and the business decision maker is normally due to subtle differences between them. The differences can be classified into five broad areas: Thought process, Background and Experience, Interest, Priorities, Language
- **Time provided to a developer to complete the project:** A common source of errors in projects comes from time constraints in delivering a product. To keep to the schedule, features can be cut. To keep the features, the schedule can be slipped. Failing to adjust the feature set or schedule when problems are discovered can lead to rushed work and flawed systems.

REPORT ON VEHICLE RENTING SERVICE

- **Over Commitment by the developer:** High enthusiasm can lead to over commitment by the developer. In these situations, developers are usually unable to adhere to deadlines or quality due to lack of resources or required skills on the team.
- **Insufficient testing and quality control:** Insufficient testing is also a major source of breakdown of e-commerce systems during operations, as testing must be done during all phases of development.
- **Inadequate requirements gathering:** A short time to market results in developers starting work on the Web site development without truly understanding the business and technical requirements. Also, developers may create client-side scripts using language that may not work on some client browsers.
- **Keeping pace with the fast changing Technology:** New technologies are constantly introduced. There may not be adequate time to develop expertise in the new technologies. This is a problem for two reasons. First, the technology may not be properly implemented. Second, the technology may not integrate well with the existing environment.

Testing Principles

- To discover as yet undiscovered errors.
- All tests should be traceable to customer's requirement.
- Tests should be planned long before the testing actually begins.
- Testing should begin "in the small" & progress towards "testing in the large".
- Exhaustive Testing is not possible.

REPORT ON VEHICLE RENTING SERVICE

- To be most effective training should be conducted by an Independent Third Party

Testing Objectives

- Testing is a process of executing a program with the intent of finding errors.
- A good test case is one that has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

Kinds of Testing:

Black Box Testing- Not based on any knowledge of internal designs or code. Tests are based on requirements and functionality.

White Box Testing- Based on the knowledge of the internal logic of an application's code. Tests are based on coverage of code statements, branches, paths and statements.

Unit Testing- The most 'micro' scale of testing; to test particular functions and code modules. Typically done by the programmer and not by the testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code; may require developing test driver modules or test harnesses.

Integration Testing- Testing of combined parts of an application to determine if they function together correctly. The 'parts' can be code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/ server and distributed systems.

REPORT ON VEHICLE RENTING SERVICE

Functional Testing- Black-box type testing geared to functional requirements of an application; testers should do this type of testing. This doesn't mean that the programmers shouldn't check that their code works before releasing it.

Regression Testing- Re-testing after fixes or modifications of the software or its environment. It is difficult to determine how much re testing is needed, especially near the end of the development cycle. Automated testing tools can be especially useful for this type of testing.

Acceptance Testing- Final testing based on the specifications of the end user or customer or based on use by end-users/ customers over some limited period of time.

User Acceptance Testing- Determining if software is satisfactory to an end user customer

Testing Technique Used

We will continuously test our project to ensure that it is fully functional. In order to perform testing test cases are designed with the intent of finding the errors in the project and help in removing those errors. Testing begins at the module level and is conducted systematically. It is generally conducted by independent test groups or third party.

Testing is done in our project Car Renting System with help of black box testing that exercise all the functional requirement of the project test cases are designed using this approach by providing set of input conditions to get the expected output.

9. IMPLEMENTATION

9.1 SYSTEM IMPLEMENTATION:-

During this stage, the software design is released as a set of programs. Unit testing involves verifying that each unit is working according to the specification of the customer.

The individual programs or units are integrated and tested as a complete system to ensure that the software requirements have been met as specified by the end users. After testing has been completed, the software system is delivered to the customer.

The last stage of SDLC is maintenance. The developer is required to maintain the proposed system from time-to-time after it has been tested and implemented by the users. The maintenance that may involve are correction of coding, rapid training and managing the user groups for the proposed system.

Password Encryption: Whatever security measures taken, in any case if anyone with destructive or malicious mind gets access to the database he can temper the critical data. To restrict it we've used encrypting algorithm facilitated to us by Microsoft ASP.Net .

Two algorithms' are there-

SHA1- It takes the password and encrypt it in a string of 40 characters.

MD5- It takes the password and encrypt it in a string of 32 characters.

The beauty of these Algorithms are that whatever be the length of password the length of converted string remains same.

However a question arises here that if this thing is that good then why didn't we use it for storing each and every data????

The answer lies in the fact that not all data are needed to be encrypted as long as we keep the critical data(like password encrypted and safe). It will

REPORT ON VEHICLE RENTING SERVICE

only increase the load on server and network with no additional advantage. So we've not used this encryption Algorithm on all data blindly.

We have implemented different approach related to security issue for our project "Car Rental System".

Authorization:

Super Admin has authorized to give permission to admin for the verification of the employees and customers.

Information Integrity:

Only admin has right in the modification of the information of employees. And only employee can modify the data of the customer .There is no possibility by the end user or unauthorized user of hacking the information.

Detection:

Super Admin will have a bank up of the project. Also there is a complex and alphanumeric password for admin and student. Super admin itself has a complex and alphanumeric password. There is a option for changing the password by all the admin and itself after every month.

10.FUTURE SCOPE

Purpose: -

The purpose of this document is to specify requirements and to give guidelines for the development of above said project. In particular it gives guidelines on how to prepare the above said project.

The advancement in Information Technology and internet penetration has greatly enhanced various business processes and communication between companies (services provider) and their customers of which car rental industry is not left out.

This Car Rental System is developed to provide the following services:

- **Enhance Business Processes:** To be able to use internet technology to project the rental company to the global world instead of limiting their services to their local domain alone, thus increase their return on investment (ROI).
- **Car Reservation:** Tools through which customers can reserve available cars prior to their expected pick-up date or time.
- **Customer's registration:** A registration portal to hold customer's details, monitor their transaction and used same to offer better and improve services to them.
- **Group bookings:** Allows the customer to book space for a group in the case of weddings or corporate meetings (Event management).

Scope: -

This project traverses a lot of areas ranging from business concept to computing field and required to perform several researches to be able to achieve the project objectives.

The area covers include:

- **Car rental industry:** This includes study on how the car rental business is being done, process involved and opportunity that exist for improvement.
- Java Technology used for the development of the application.

REPORT ON VEHICLE RENTING SERVICE

- General customers as well as the company's staff will be able to use the system effectively.
- Web-platform means that the system will be available for access 24/7 except when there is a temporary server issue which is expected to be minimal
- Eco-friendly: The monitoring of the car activity and the overall business becomes easy and includes the least of paper work.
- The software acts as an office that is open 24/7. It increases the efficiency of the management at offering quality services to the customers.
- It provides custom features development and support with the software.

11. Limitations

Limitations :-

Customers can't directly access this software only by taking the assistance of employee they can use the service of this system although this system is very comfy for the employees in the organization.

- 1) The car renting system is only limited to the home country which means it only rents car inside the country border it only gives three option city ride , state ride and country ride .
- 2) Car renting system does not deal with the situations in which if any accident or damage occurs it only deals with the fine for delays in return date , the system does not calculate for the damage done to the car.
- 3) Special characters like @, #, *, & etc.. in text fields are allowed.
- 4) There is no specified length restriction given to the username and password field and both fields accept alphanumeric value.

12. Bibliography

References

The following books were referred during the analysis and execution phase of the project

Mastering Netbeans

By David Salter

SOFTWARE ENGINEERING

By Roger S. Pressman

Java - The Complete Reference

By Herbert Schlidt

COMPLETE REFERENCE .NET

By David S Platt

MSDN 2003

By Microsoft

IMAGES

Google Search

Pinterest

Bing Search

HTML PUBLISHING BIBL

- Alan Simpson.

C# 2008

Andrew Troelson

REPORT ON VEHICLE RENTING SERVICE

WEBSITES:

www.google.com
www.bing.com
www.msn.com
www.wikipedia.com
www.imagegallery.com
www.sql.com'
www.learnjava.com

REPORT ON VEHICLE RENTING SERVICE

Bio data of each group member:

1. VIKAS MISHRA

ENROLLMENT NO. =1700101199

COURSE =MCA

SEMESTER =4th

2. ROSHAN KUMAR SHARMA

ENROLLMENT NO. =2000102066

COURSE =MCA

SEMESTER =4th