**Chapter 1**

**INTRODUCTION**

1. **Introduction to CAR SHOWROOM MANAGEMENT**

Database applications are software programs designed to collect, manage and disseminate information efficiently. The current project deals with CAR SHOWROOM MANAGEMENT for the owner to deals with organization. Car showroom management system software is useful for handling distributors information, cars waiting list details, billing details, customer’s information, Customer help.

The car showroom management system is a standalone application which is based on selling cars also generating bill and also storing customer and employee details. The main principle behind the need of car showroom management system is easy supervision of car showroom. It has user friendly and modular approach. The modular approach of the software increases the flexibility and the software. Data storing is easier. It will be able to view the details a any time. Therefore, further the software can make all the system computerize, reduce time consumption, simple database is maintained, easy operator of the system, user interfaces are user friendly, it takes very less time to get use to with system. This project also maintains the record of sold car and available quantity of car.

1. **Introduction to Database Management System**

[1] A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyse data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, Enterprise DB, MongoDB, Maria DB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MySQL, SQLite and IBM.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Sometimes a DBMS is loosely referred to as a "database".

1. **Applications**

Databases are used to support internal operations of organizations and to underpin online interactions with customers and suppliers.

Databases are used to hold administrative information and more specialized data, such as engineering data or economic models. Examples of database applications include computerized library systems, flight reservation systems, computerized parts inventory systems, and many content management systems that store websites as collections of webpages in a database.

1. **Introduction to JAVA**

[2] Java is a general-purpose [computer programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation.[[17]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-design_goals-17) Java applications are typically compiled to [byte code](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture).

Swing is a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) [widget toolkit](https://en.wikipedia.org/wiki/Widget_toolkit) for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It is part of [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation)'s [Java Foundation Classes](https://en.wikipedia.org/wiki/Java_Foundation_Classes) (JFC) – an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for providing a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) for Java programs.

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

1. **Introduction to JDBC**

**Java Database Connectivity** (**JDBC**) is an [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), which defines how a client may access a [database](https://en.wikipedia.org/wiki/Database). It is Java based data access technology and used for Java database connectivity. It is part of the [Java Standard Edition](https://en.wikipedia.org/wiki/Java_Standard_Edition) platform, from [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). It provides methods to query and update data in a database, and is oriented towards [relational databases](https://en.wikipedia.org/wiki/Relational_database) and Java programming. It allows java program to execute SQL statement and retrieve result from database.

The JDBC provide the following advance feature

* Connection Management
* Auto loading of Driver Interface.
* Better exception handling
* Support for large object
* Annotation in SQL query.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

* Making a connection to a database.
* Creating SQL or MySQL statements.
* Executing SQL or MySQL queries in the database.
* Viewing & Modifying the resulting records.

1. **JDBC Architecture**

The JDBC API supports both two-tier and three-tier processing models for database access but in general, JDBC Architecture consists of two layers

* + JDBC API: This provides the application-to-JDBC Manager connection.
  + JDBC Driver API: This supports the JDBC Manager-to-Driver Connection.

The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases. The JDBC driver manager ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

1. **Introduction to SQL**

[3] Structure Query Language (SQL) is a programming language to request data from a database, to add, update, or remove data within a database, or to manipulate the metadata of the database. SQL was the first commercial language introduced for E.F Cod’s **Relational** model. Today almost all RDBMS (MySQL, Oracle, Informix, Sybase, MS Access) uses **SQL** as the standard database language. SQL is used to perform all type of data operations in RDBMS. Commonly used statements are grouped into the following categories:

* **Data Query Language (**[**DQL**](https://en.wikipedia.org/wiki/Data_Query_Language)**)**
  + SELECT - Used to retrieve certain records from one or more tables.
* **Data Manipulation Language (**[**DML**](https://en.wikipedia.org/wiki/Data_Manipulation_Language)**)**
  + INSERT - Used to create a record.
  + UPDATE - Used to change certain records.
  + DELETE - Used to delete certain records.
* **Data Definition Language (**[**DDL**](https://en.wikipedia.org/wiki/Data_Definition_Language)**)**
  + CREATE - Used to create a new table, a view of a table, or other object in database.
  + ALTER - Used to modify an existing database object, such as a table.
  + DROP - Used to delete an entire table, a view of a table or other object in the database
* **Data Control Language (**[**DCL**](https://en.wikipedia.org/wiki/Data_Control_Language)**)**
  + GRANT - Used to give a privilege to someone.
  + REVOKE - Used to take back privileges granted to someone.

1. **Introduction to MySQL**

**MySQL** is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system)(RDBMS). Its name is a combination of “My”, the name of co-founders Wideness’s daughter, and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). The MySQL development project has made its [source code](https://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-profit](https://en.wikipedia.org/wiki/Business) firm, the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), now owned by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). For proprietary use, several paid editions are available, and offer additional functionality.

1. **Features**

Car Showroom Management System gives you the power to:

* Manage Customer Information Efficiently.
* Define the car-sailed, available for sale, purchase car etc.
* Manage the type of service.
* Provide the mechanic for the services.

**Chapter 2**

**REQUIREMENTS SPECIFICATION**

**2.1 Hardware Requirements**

* Processor: Intel(R) Core(TM) i3 CPU @1.70Ghz
* Installed memory (RAM): 4.00GB
* System type: 64-bit Operating System, x64-based processor
* Graphics: NVIDIA (R) GEFORCE(R)
* Total size of Hard disk: 1TB

**2.2 Software Requirements**

* Operating System: Microsoft windows 7 and above.
* Integrated Development Environment:

1. Front-end tool: - Eclipse Neon2 4.6.2
2. Back-end tool: - MySQL

* Programming Language: JAVA

**Chapter 3**

**DESIGN IMPLEMENTATION**

**3.1 SYSTEM DESIGN**

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

System designing in terms of software engineering has its own value and importance in the system development process as a whole. To mention it may though seem as simple as anything or simply the design of systems, but in a broader sense it implies a systematic and rigorous approach to design such a system which fulfils all the practical aspects including flexibility, efficiency and security.

Before there is any further discussion of system design, it is important that some points be made clear. As it goes without saying that nothing is created that is not affected by the world in which it’s made. So, the systems are not created in a vacuum.

They are created in order to meet the needs of the users. They are not only intended to solve the existing problems, but they also come up with acceptable solutions to the problems that may arise in the future. The whole process of system development, from blueprint to the actual product, involves considering all the relevant factors and taking the required specifications and creating a useful system based on strong technical, analytical and development skills of the professionals.

Let’s get back to our discussion about what the system design phase is and the importance of system design in the process of system development. Being another important step in the system development process, system designing phase commences after the system analysis phase is completed. It’s appropriate to mention that the output or the specifications taken through the phase of system analysis become an input in the system design phase which in turn leads to workout based on the user defined estimations.

The importance of this phase may be understood by reason of the fact that it involves identifying data sources, the nature and type of data that is available. For example, in order to design a salary system, there is a need for using inputs, such as, attendance, leave details, additions or deductions etc. This facilitates understanding what kind of data is available and by whom it is supplied to the system so that the system may be designed considering all the relevant factors. In addition, system designing leads to ensure that the system is created in such a way that it fulfils the need of the users and keep them at ease being user-oriented. In terms of the flexibility, one of the main objectives of this phase is that it is intended to design such a system which can be dynamic in nature and responsive to the changes if required. Another important objective is that the phase of system designing is concerned with creating the system which can work efficiently providing the required output and being responsive to the time within a given time limit. The aspect of reliability and physical security of data cannot be ignored. With this respect, the system designing phase ensures security measures of the system effectively and efficiently.

**3.2 E-R Diagram**

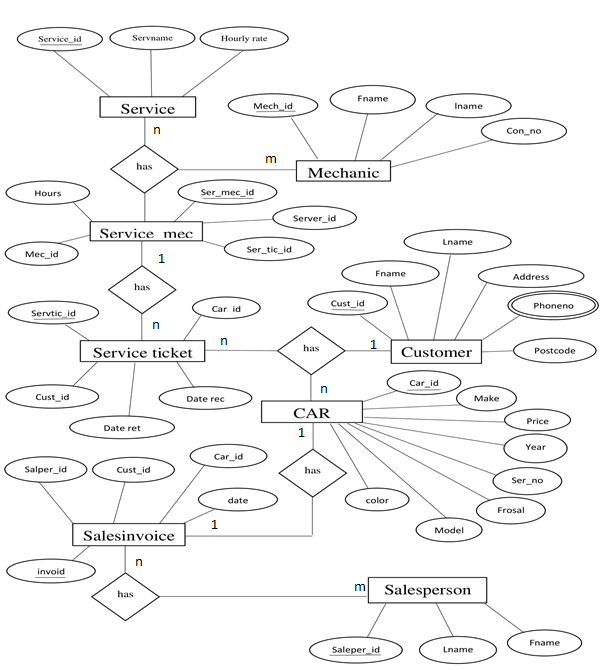
Fig:3.2.1 shows an entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

**3.3 Logical Schema**

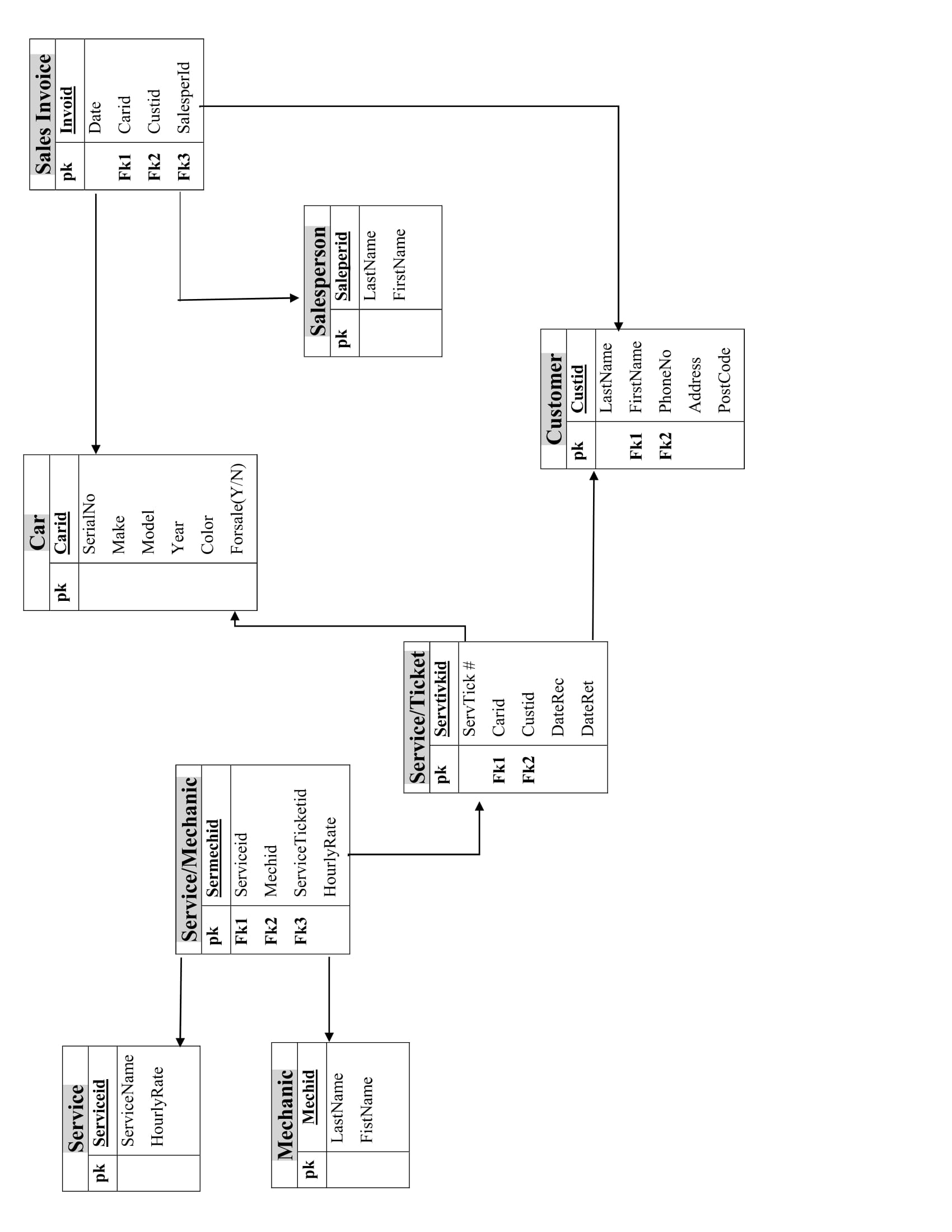
A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It’s the database designers who design the schema to help programmers understand the database and make it useful.

Logical Database Schema − This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

****

3.3.1 E-R Diagram

****

3.3.2 Logical Schema

**Chapter 4**

**IMPLEMENTATION**

The special methods used in the project are explained.

**4.1 Triggers**

[4] A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. Some uses for triggers are to perform checks of values to be inserted into a table or to perform calculations on values involved in an update.

A trigger is defined to activate when a statement inserts, updates, or deletes rows in the associated table. These row operations are trigger events. For example, rows can be inserted by [INSERT](https://dev.mysql.com/doc/refman/5.7/en/insert.html) or [LOAD DATA](https://dev.mysql.com/doc/refman/5.7/en/load-data.html) statements, and an insert trigger activates for each inserted row. A trigger can be set to activate either before or after the trigger event. For example, you can have a trigger activate before each row that is inserted into a table or after each row that is updated.

[5] The trigger used in the project is a after insert trigger. In this trigger we are inserting a new row in the customer table immediately after we insert a new row in customer table.

**4.2 Stored procedure**

[6] A procedure (often called a stored procedure) is a subroutine like a subprogram in a regular computing language, stored in database. A procedure has a name, a parameter list, and SQL statement(s). All most all relational database system supports stored procedure, MySQL 5 introduce stored procedure. In the following sections we have discussed MySQL procedure in details and used MySQL 5.6 under Windows 7. MySQL 5.6 supports "routines" and there are two kinds of routines: stored procedures which you call, or functions whose return values you use in other SQL statements the same way that you use pre-installed MySQL functions like pi(). The major difference is that UDFs can be used like any other expression within SQL statements, whereas stored procedures must be invoked using the CALL statement.

**4.3 Stored procedure and trigger used in the above application**

**delimiter $**

**create procedure smsdel(IN id int)**

**begin**

**delete from service mechanic where service\_mechanic\_id=id;**

**end $**

**delimiter /**

**create trigger pm**

**before insert on mechanic**

**for each row**

**begin**

**set NEW.ph\_no=CONCAT ("+91 “, NEW.ph\_no);**

**end;**

**$**

**delimiter $**

**create procedure smdel(IN id int)**

**begin**

**delete from service where service\_id=id;**

**end $**

**delimiter /**

**Chapter 5**

**TESTING**

**5.1 Introduction to testing**

Verification and validation is a generic name given to checking processes, which ensures that the software confirms to its specifications and meets the demands of users.

* **Validation**

Validation involves checking that the program has implanted meets the requirement.

* **Verification**

Verification involves checking that the program confirms to its specification.

**5.2 Stages in the Implementation of Testing**

* **Unit Testing**

Each individual unit is tested for correctness. These individual components will be tested to ensure that they operate correctly.

* **Module Testing**

A module is a collection of dependent components such as a function. A module encapsulates related components so can test without other system modules.

* **Sub-system Testing**

This phase involves testing collection of modules, which have been integrated into sub-systems. Sub-systems may be independently designed and implemented.

* **System testing**

The Sub-systems are integrated to make up the entire system. The errors that result from unanticipated interaction between sub-systems and system components are removed.

* **Acceptance testing**

This is the final stage in the testing process before the system is tested for

operational use. Any requirement problem or requirement definition problem

revealed from acceptance testing are considered and made error free.

* **Test plan**

Careful planning is needed to the most of testing and controlled testing cost.

**5.3 Results**

Several errors were detected and rectified and the whole project is working as it should with proper output and high efficiency. The several tests performed are as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test case** | **Input data** | **Steps to execute the test case** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Login screen. | Wrong username or password. | After entering the data click the login button. | A proper message indicating the error should appear and the user should be redirected to login screen. | A message  was displayed saying incorrect username or password. | Pass |
| 2 | customer  Screen. | inserting a user with existing user\_id. | Enter the user\_id and click the add button. | A message should be displayed saying incorrect user\_id. | A message was displayed saying incorrect user\_id. | Pass |
| 3 | Showroom  screen | Close logout | Click on logout | It will go back to the login page | Go back to login page | Pass |
| 4 | Car tab | Trying to delete car details. | Enter the Car number in dialog box and click ok button | The car with car\_id details has to be deleted | The car info is deleted | Pass |

**Chapter 6**

**SNAPSHOTS**

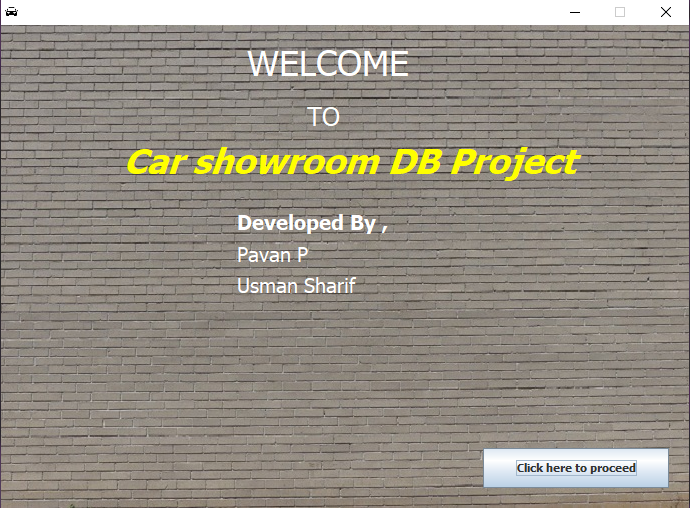


Figure 6.1:Welcome page

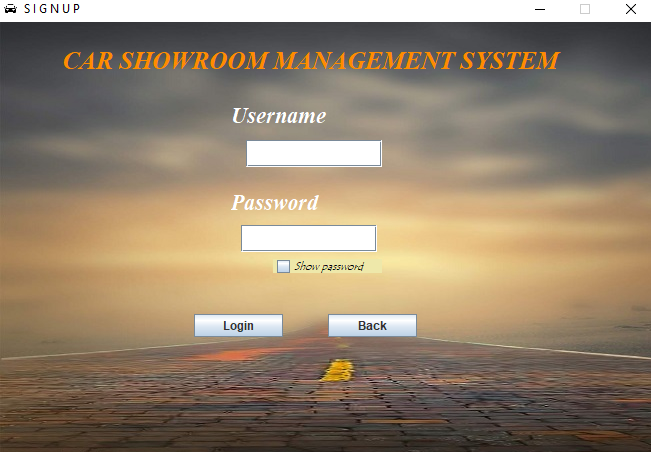


Fig 6.2: Login page

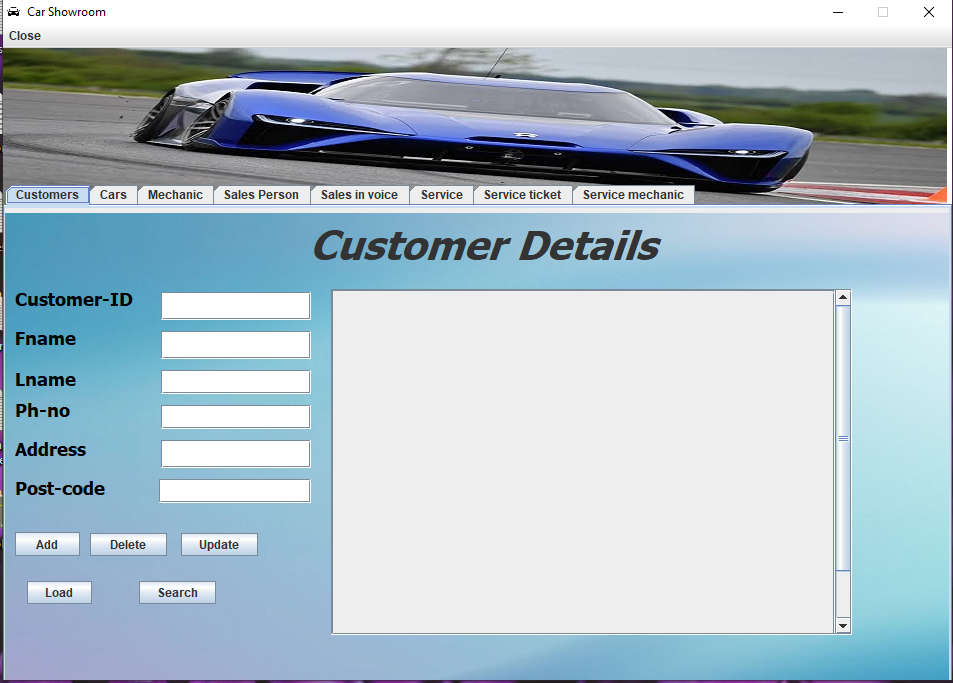


Fig 6.3: Customer details Screen.

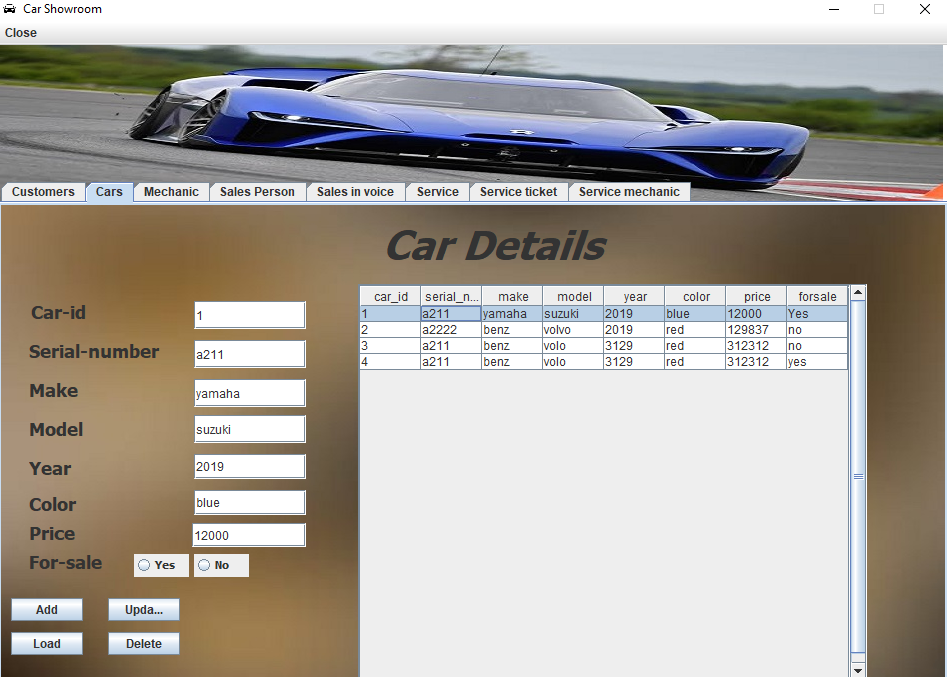
****

Fig 6.4: Car details screen.



Fig 6.5: Mechanic details screen.

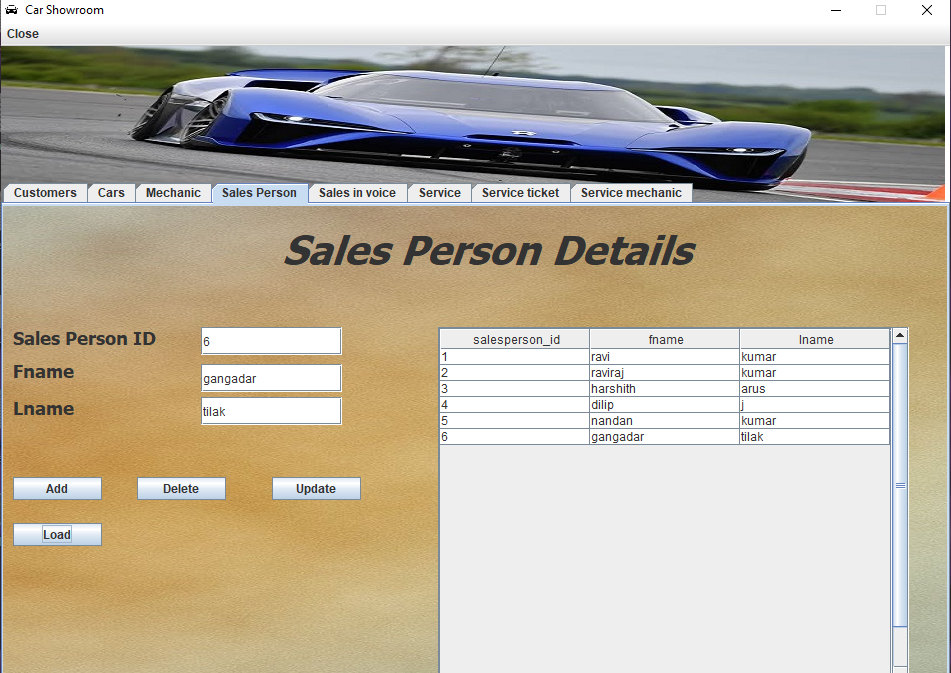


Fig 6.6: Sales person details screen.



Fig 6.7: Sales invoice screen.

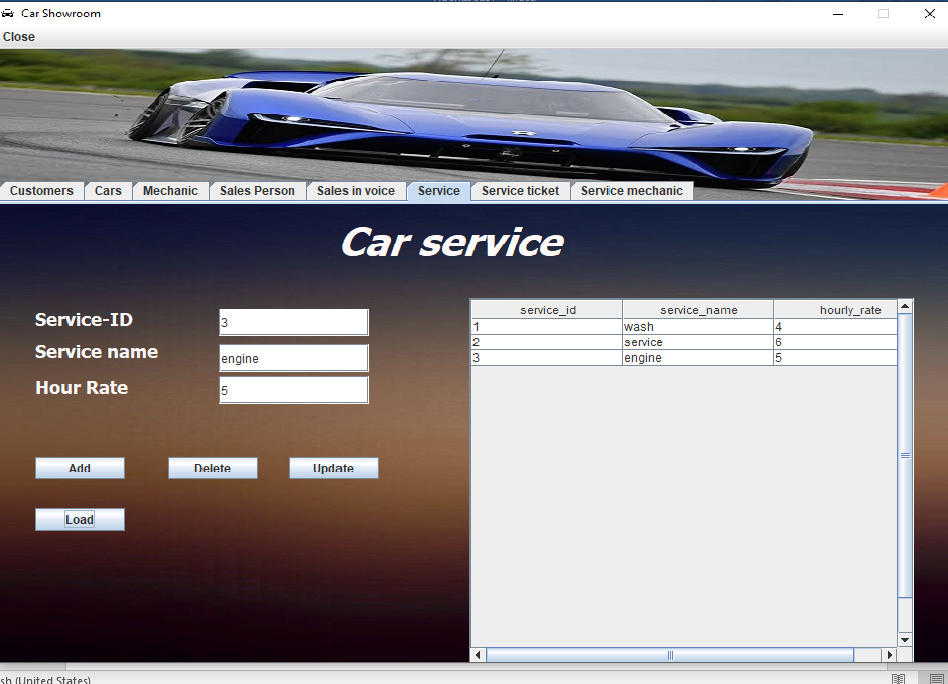


Fig 6.8: Car-service details.

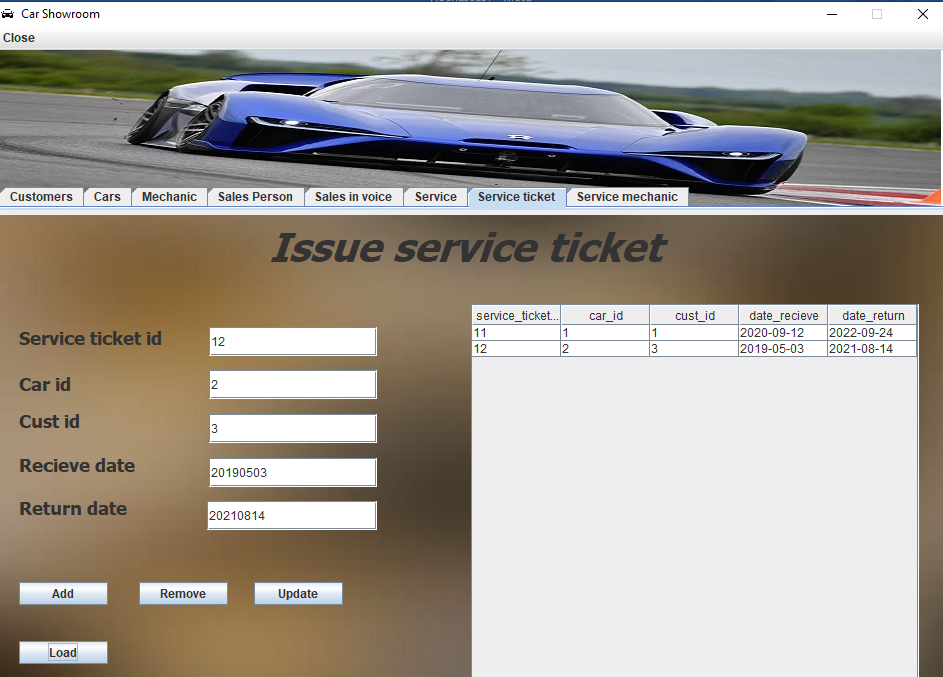


Fig 6.9: Service ticket details.

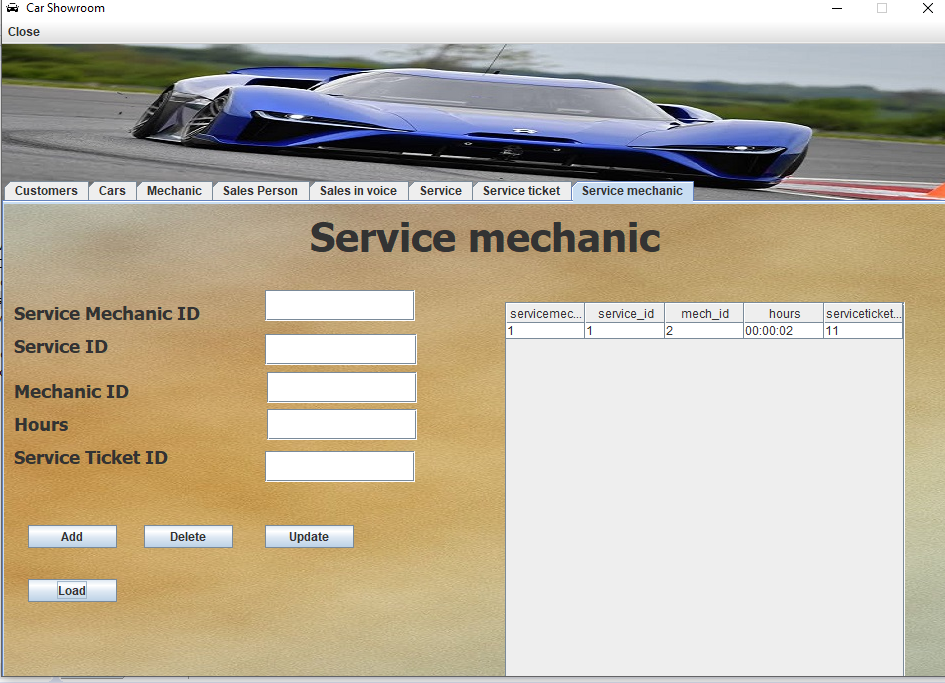


Fig 6.10: Service mechanic details.

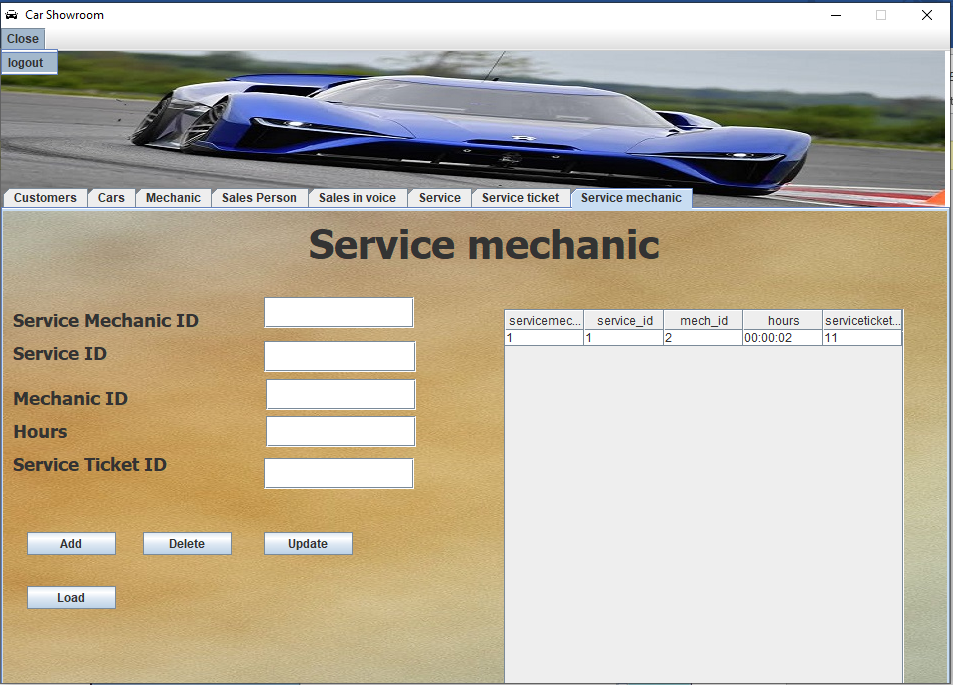
****

Fig 6.11: Logout screen

v

**Chapter 7**

**CONCLUSION**

This project presents the modelling, analysis and implementation of a car showroom. Especially, this paper also proposed the module of how a basic car showroom managing works which can assist in designing the extended models. The advantage of the proposed approach is the clear presentation of the showroom system behaviour and readiness for implementation.

**Chapter 8**

**FUTURE ENHANCEMENT**

This mini project was developed to fulfil user requirement; however, there is scope to improve performance of the car showroom in the area of user interface, database performance and query processing time etc.

The future enhancements that are possible in the project are as follows:

* Some enhancement in the code can be made for improved performance.
* Improvement in Graphics and Database handling can be made.
* In future reports can be generated.

**BIBLIOGRAPHY**

**Books:**

[1] Fundamental of DATABASE SYSTEMS – Ramez Elmasri, Shamkanth B Navathe,7th Edition ,Pearson Education,2017.

[2] The Complete Reference JAVA – Herbert Schildt, 7th Edition, Mc Graw Hill Education.

[3] Database management systems, Ramakrishnan, and Gehrke , 3rd Edition, 2014,McGraw Hill.

**Websites & links:**

[4] <https://www.tutorialspoint.com/plsql/plsql_triggers.htm>l

[5] <https://www.w3resource.com/mysql/mysql-triggers.php>

[6][https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-stored- procedures.html](https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-stored-%20%20%20%20procedures.html)