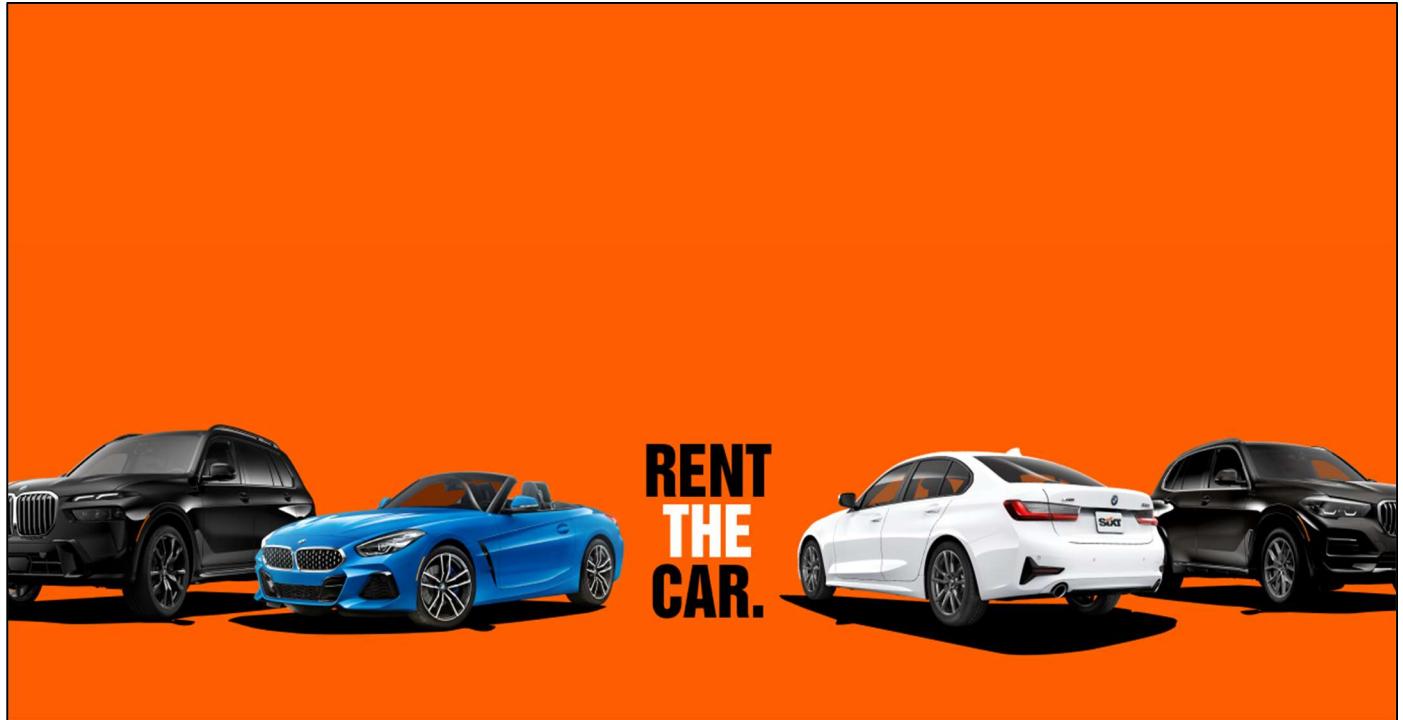


PROJECT REPORT

Car Rental System



Developed by

Name : MD. RIZWAN FATMI

Address : Patna, Bihar-800 006

Mobile : + 91 8210006010

PREFACE

This project is for practice purposes, which requires a combination of practical work and detailed work.

The title of the project is “Car Rental System”. To make such a project, I have tried and attempted to deal every aspect with sufficient knowledge and complete the report as per requirement & guidelines.

The approach adopted here to solve the project is not unique. Problems could be solved through different techniques but I have adopted the simplest one, which is easy to understand. There may be the possibility of more flexibility in the problem but I have attempted to consider all necessary aspects for solving the project.

Lastly, your comments, suggestions and criticism for further improvement in project exercise will be appreciated.

ACKNOWLEDGEMENT

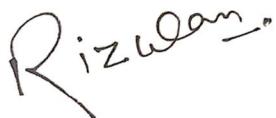
My acknowledgment extends to all those who teach me to code and prepare projects.

I am thankful to all of my teachers who give me valuable and very important guidance and information to make this Project Report Successful.

I am grateful to other members of my study center and regional center, Patna for imparting to me the required & necessary training, information, guidelines, sympathy, and criticism which upgraded my programming skills and troubleshooting while doing this project very much.

Lastly, I would like to thank my family, my friends, and my respected seniors for their moral support and other precious co-operations.

Place: Patna



Signature

PROJECT SUMMARY

SUBJECT / TITLE : "CAR RENTAL SYSTEM".

PROJECT CATEGORIES : JAVA BASED DBMS.

PREPARED BY : MD.RIZWAN FATMI.

ADDRESS : PATNA, BIHAR

MOBILE NO. : + 91 8210006010.

EMAIL : rizwanfatmi@gmail.com.

CONTENTS

1. INTRODUCTION OF THE PROJECT	001
2. OBJECTIVES	003
3. TOOLS/ENVIRONMENT USED	005
4. ANALYSIS OF THE SYSTEM	007
• SRS	
• FEASIBILITY STUDY	
• DATA FLOW DIAGRAMS	
• E-R DIAGRAMS	
• USE CASE DIAGRAM	
• PROJECT SCHEDULING	
• SDLC	
5. SYSTEM DESIGN	024
• MODULARIZATION DETAILS	
• DATABASE DESIGN	
• INPUT SCREENS	
7. TESTING IMPLEMENTATION & MAINTENANCE	033
• TYPE OF TESTING	
• IMPLEMENTATIONS	
• CODE OPTIMIZATION	
• MAINTENANCE	
8. IMPLEMENTATION OF SECURITY	042
9. OUTPUT SCREENS	044
10. LIMITATION / FUTURE APPLICATION OF THE PROJECT	046

INTRODUCTION



INTRODUCTION :

The title of this project is "Car Rental System". This project is being implemented as an Educational project only for practice purposes. Through this project, we simply know about all the Car renting system-related common operations which are mentioned in the given modules.

This project has the following Sections/Modules: -

- Login Details
- Customer Details
- Car's Details
- Car Issue Details
- Car Return Details

OBJECTIVES



OBJECTIVES :

In the present scenario, the computer becomes an essential component of our life. We are using computers in most fields, especially in big Organizations to smoothly, quickly, and effectively run that Organization.

For Example, in the Car renting system is being used for recording Customer information, car information, car issue information, and car return information as well as controlling the management of the Car rental system. This maintenance makes the work easier & fast and the pressure on work employee get reduced.

This project implements an easy-to-use interface for the staff so that it can be handled without much training. A user-friendly interface helps and easier the work of data updating and consequently generating the updated reports or records.

My project has exactly tried to reduce the workload of the staff.

TOOLS /ENVIRONMENT/PLATFORM USED



Tools Used:

Front End GUI Tools	:	Java,Java_Swing
RDBMS Back End	:	Sqlite 3
Operating System	:	Windows 10 Home Basic

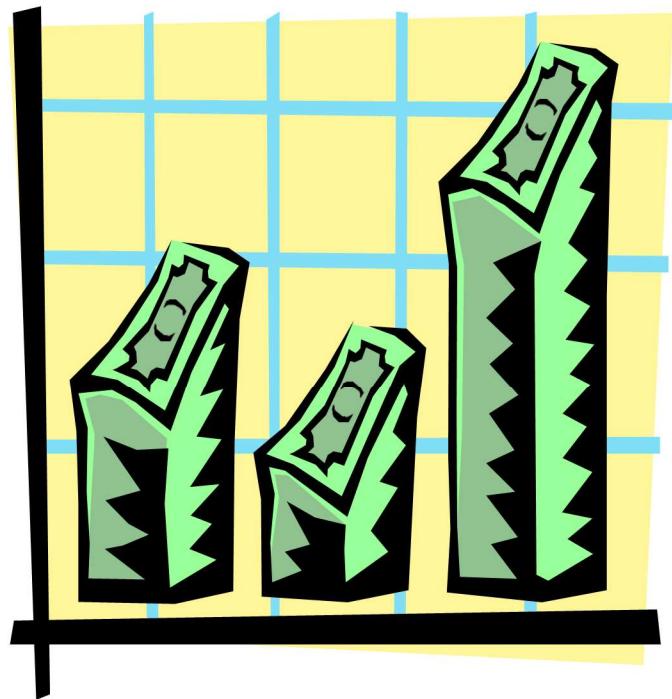
Software Requirement:

- Front End Software : JDK 18, J2EE1.4, Net Beans IDE 8.0.2
- Backend Software : Sqlite 3

Hardware Requirement:

- CPU : Intel core2duo processor,3.2GHz
- RAM : 2 GB/More
- HARD DISK : 160 GB/More

ANALYSIS OF THE SYSTEM



INTRODUCTION: THE EXISTING SYSTEM :

Before developing a new system anywhere, it is imperative for the developers to acquaint themselves with the environment in which the proposed system would be installed. Also, it is necessary to interact with the user regarding their demands and the resources available to them. This gives a clear overview of the improvements wanted and to be provided.

The study and analysis of the mechanism of the Library Management System were carried out diligently.

SYSTEM REQUIREMENT & SPECIFICATION :

- In the Car Rental System Company has to record details of car issues and car return of every customer, maintaining that record normally or through a manual way is very complex, thus arises the need for computerization.
- Maintenance of huge records of customers and cars. This requires an efficient and accurate Management System for the data.

FEASIBILITY STUDY :

Feasibility is the determination of whether or not a project is worth doing/possible or not in various senses of the parameter. The process followed in making this determination is called a feasibility study.

In the conduct of the feasibility study, the following types of feasibility are studied-

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility
- Management Feasibility
- Legal Feasibility
- Time Feasibility

Technical Feasibility

This is concerned with specifying equipment and software that will successfully satisfy the user requirements.

Hardware: Hardware selected has been examined against the processing capacity, and the memory requirement and found satisfactory for current as well as near future workload.

Software: Software selected has been examined against the processing, reliability, flexibility and accuracy are satisfactory.

System is developed under backend or frontend tool. For backend RDBMS's are the best for database handling these days and Sqlite3 is RDBMS'S package. Again for the frontend, after selection of Sqlite3 as backend, the feasible options left were java from these.

Operational Feasibility

It is mainly related to human organizational and political aspects. it asks if the system will work when it is developed and installed.

After considering the following point that the system is behaviorally feasible: -

- The project is carried out on the request of the users.
- To a greater extent, the proposed system aims at maximizing user friendliness. This is intended to overcome resistance to change by the existing staff.
- Since users are not experienced in handling a computerized system, it is planned to provide phase wise training to different levels of the existing staff, by the trainer.
- Users have been made aware of the power of the software and hardware environment.

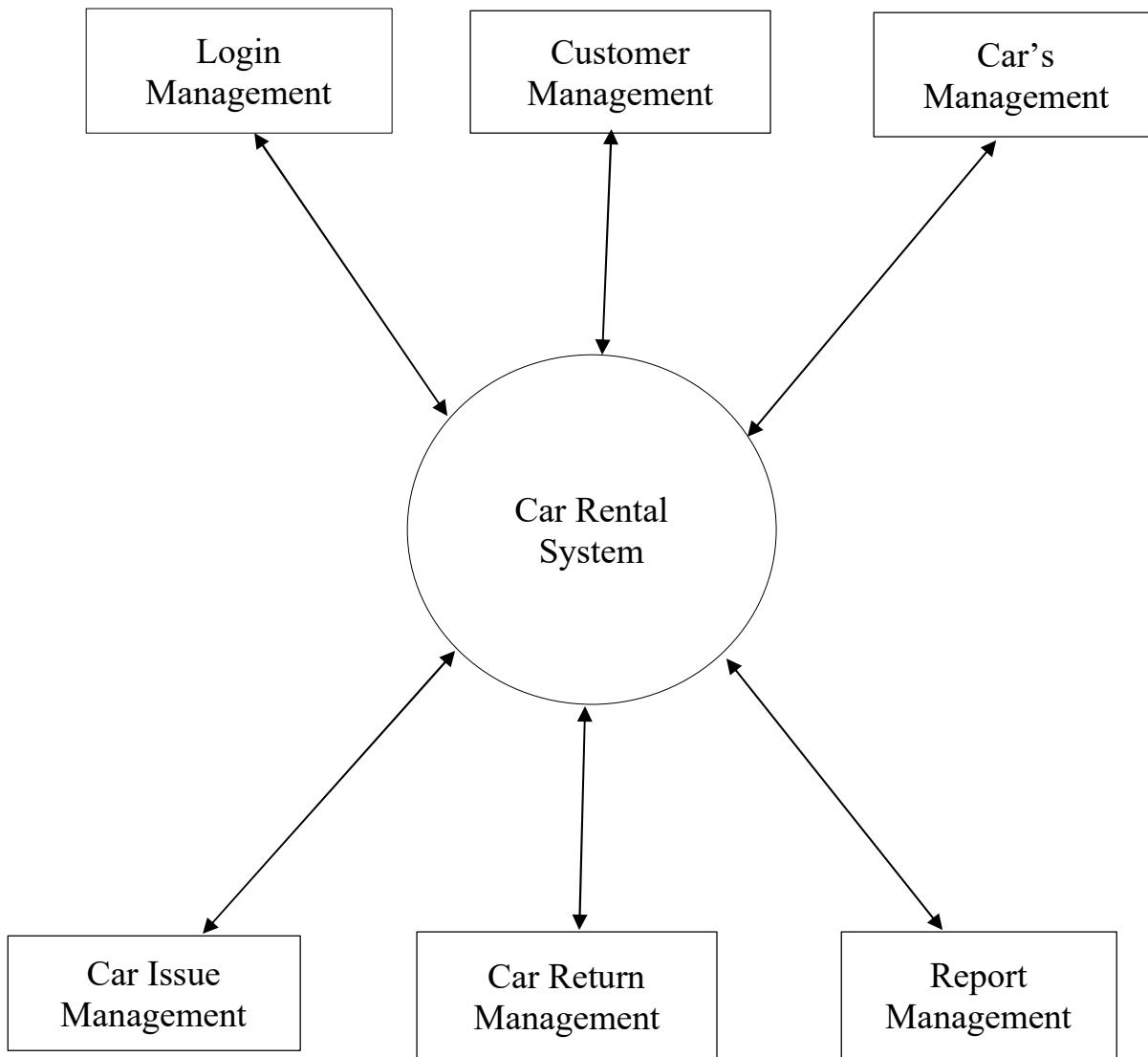
Economical Feasibility

It is the most frequently used technique for evaluating the effectiveness of a proposed system. Major cost at this point of time would be incurred towards acquiring an RDBMS package only. Car Rental System is fully capable of arranging all the basic software and hardware according to the requirement. Major benefits expected from this system are: -

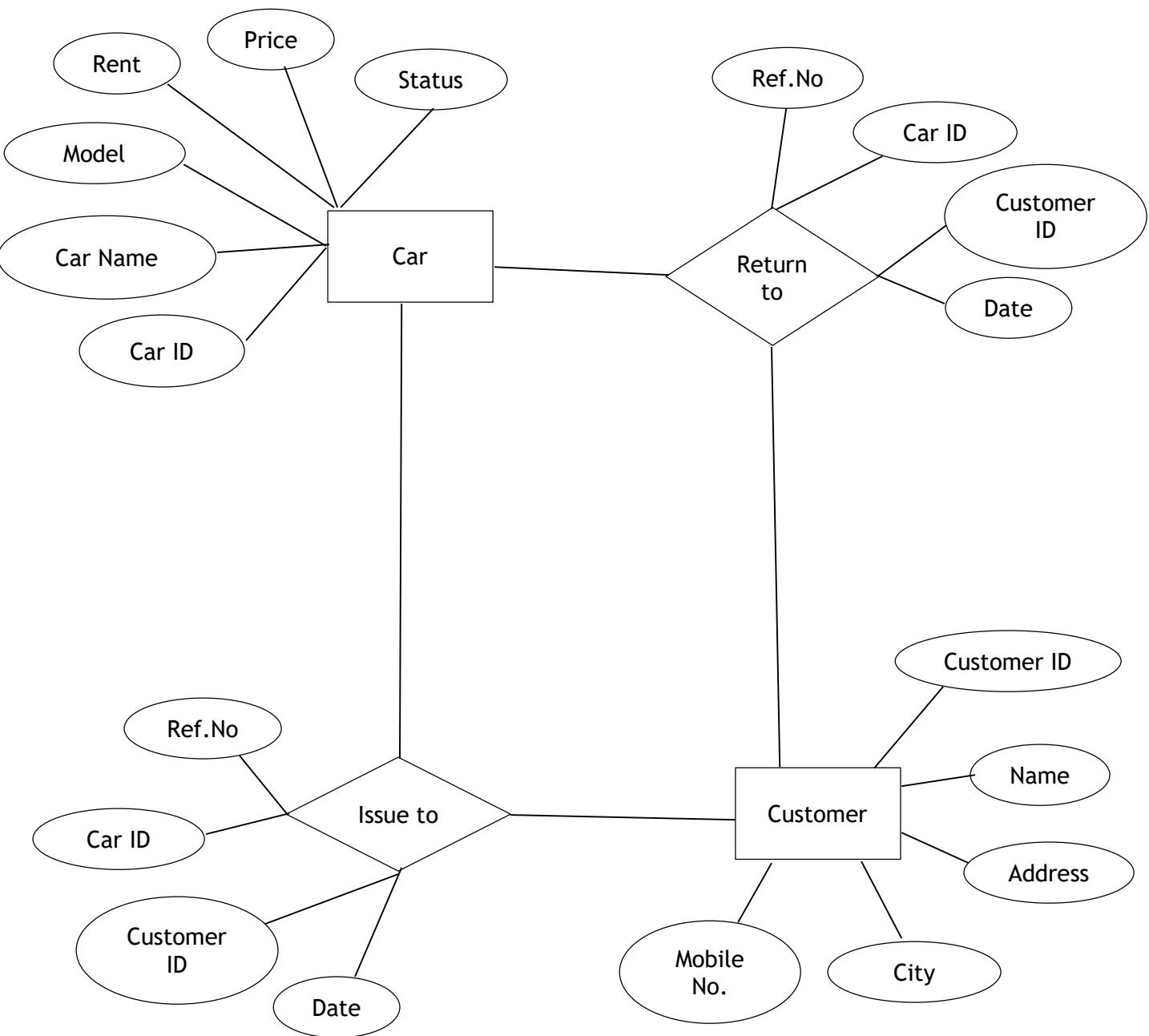
- Increased speed of work
- Less time consuming
- More Accurate work
- More flexible processing

Overall, we can see this project or software is economical feasible.

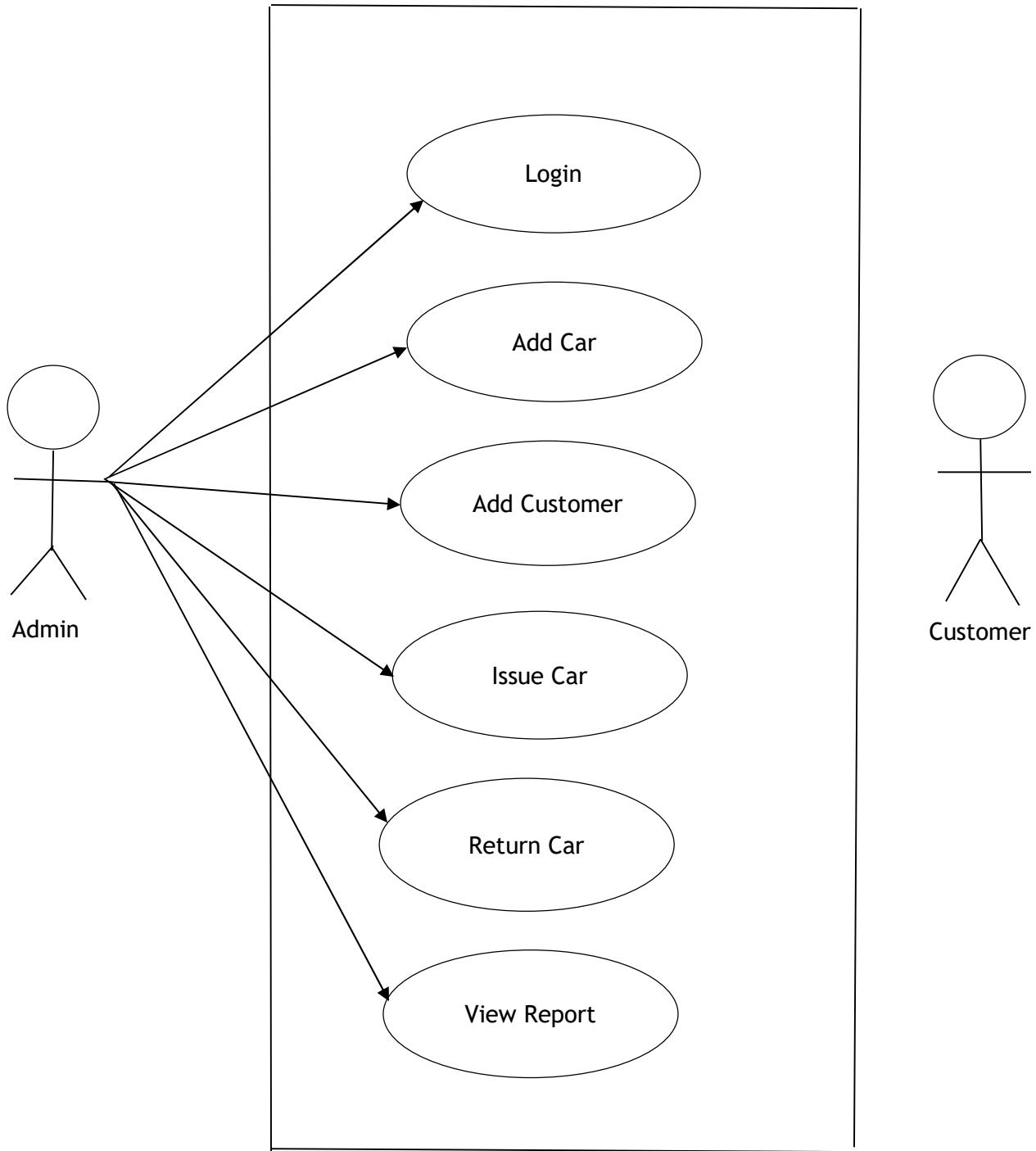
DATA FLOW DIAGRAM (DFD)



ER-DIAGRAM



USE CASE DIAGRAM



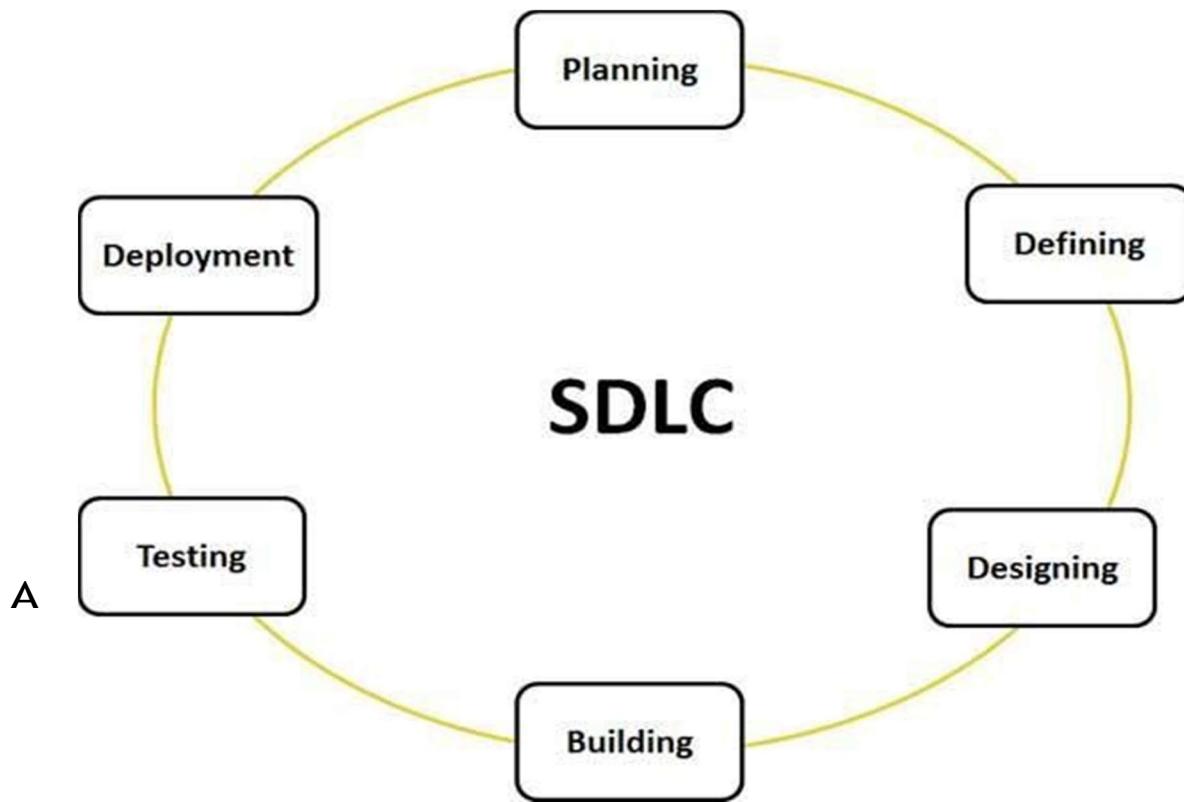
SDLC

SDLC is the acronym of Software Development Life Cycle. It is also called as Software development process. The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development life cycle consists of the following stages:

Stage 1: Planning and Requirement Analysis: Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements: Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through 'SRS' - Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the product architecture: SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in DDS - Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity , budget and time constraints , the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product: In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc are used to generate the code. Different high level programming languages such as C, C++, Pascal, **Java**, and JSP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product: This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However this stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance: Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometime product deployment happens in stages as per the organizations' business strategy. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during software development process. These models are also referred as "Software Development Process Models". Each process model follows

a Series of steps unique to its type, in order to ensure success in process of software development. Following are the most important and popular SDLC models followed in the industry:

- □ Waterfall Model
- □ Iterative Model
- □ Spiral Model
- □ V-Model
- □ Big Bang Model

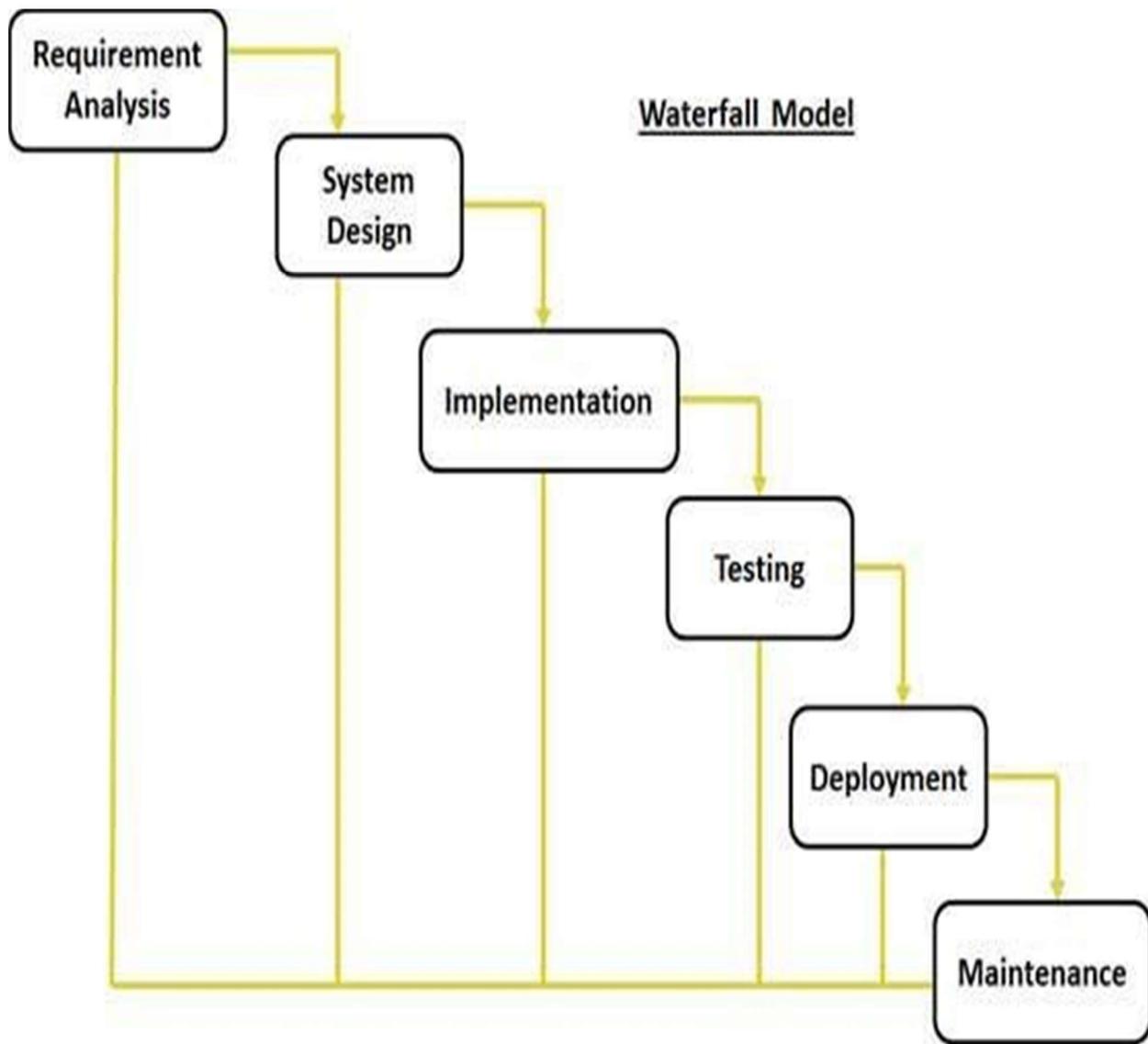
The other related methodologies are Agile Model, RAD Model – Rapid Application Development and Prototyping Models.

Waterfall model is the earliest SDLC approach that was used for software development .The waterfall Model illustrates the software development processin a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins onlyif the previous phase is complete. In waterfall model phases do not overlap.

Waterfall Model design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, thewhole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for thenext phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

- Requirement Gathering and analysis** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

□ □ **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

□ □ **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

□ **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

□ **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

□ **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

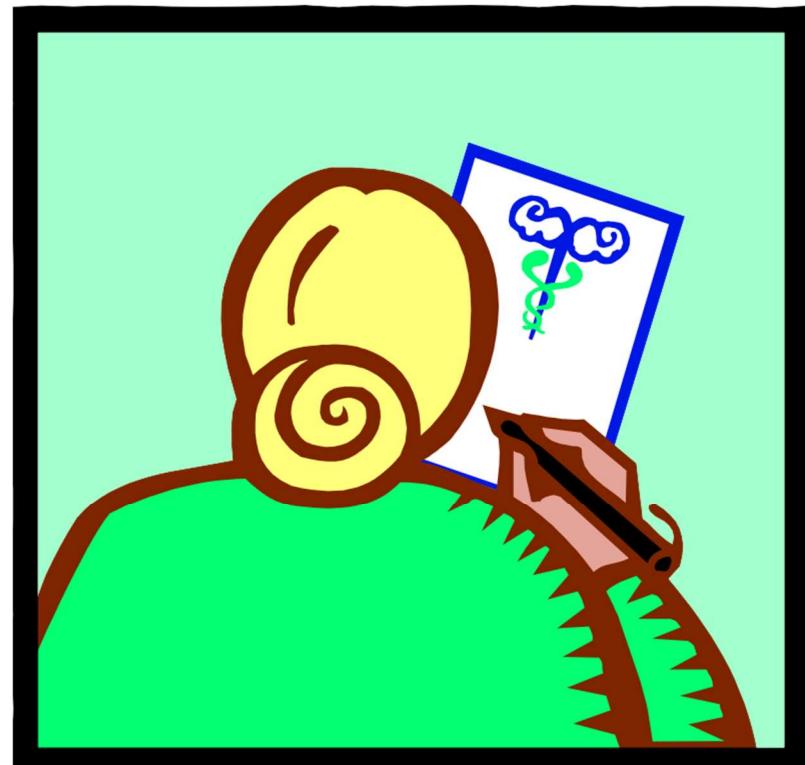
All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- □ Requirements are very well documented, clear and fixed
- □ Product definition is stable
- □ Technology is understood and is not dynamic
- □ There are no ambiguous requirements
- □ Ample resources with required expertise are available to support the product
- □ The project is short

THE SYSTEM DESIGN



NUMBER OF MODULES AND ITS DESCRIPTION

INPUT OF THE PROJECT

The Input of a software project is most important because it is the raw data of it. without Input, one cannot achieve the required output. The Input of this project is very simple. The Data in the following sections are directly inserted by the admin.

These car rental system entities, are involved in the functionality of its. The following are the main input entities:

- **Add Customer:** The admin is one who adds information about the customer and get customer details as a report.
- **Add Car:** The admin can also add information about the car and get car details as a report just like the customer.
- **Issue Car:** The admin can issue the car to the customer and can get car issue report.
- **Return Car:** the car returned by the customer is entered into the system by the admin and the admin can get the report of the returned car also.

OUTPUT TO THE PROJECT

While doing a project output is an essential and necessary part, because if we are doing some work there must be some output so that we can know about the conclusion of our work and whether it is successful or not. The output of the project is to view all the records stored in different tables in a very logical manner so that the information from that tables/register can be easily understandable and highly illustrative even represented in a brief format. The outputs of the records are arranged in the following ways:

1. Customer List : This report gives the details of the Customer.
2. Car's List : This report gives the details and status of Car's.
3. Car Issue List : This report gives the details of issued car's.
4. Car Returned List : This report gives the details of returned car's which was issued to the customer.

DATABASE DESIGN/TABLE OR DATA STRUCTURE OF EACH MODULE

1. Table Name : - Customer

ATTRIBUTE	DATATYPE	CONSTRAINTS	PURPOSE
Customer ID	INT	Primary Key	For identification of customer's.
Name	VARCHAR(45)	Not Null	For saving the customer's details
Address	VARCHAR(100)	Not Null	For saving the customer's details
City	VARCHAR(50)	Not Null	For saving the customer's details
Mobile Number	VARCHAR(10)	Not Null	For saving the customer's details

2. Table Name :- Car

ATTRIBUTE	DATATYPE	CONSTRAINTS	PURPOSE
Car ID	INT	Primary Key	For identification of car's
Name	VARCHAR(45)	Not Null	For saving the car's details
Model	VARCHAR(45)	Not Null	For saving the car's details
Rent	VARCHAR(15)	Not Null	For saving the car's details
Prince	VARCHAR(15)	Not Null	For saving the car's details
Status	VARCHAR(45)	Not Null	For saving status of the car's

3. Table Name :- Car Issue

ATTRIBUTE	DATATYPE	CONSTRAINTS	PURPOSE
Ref.No	INT	Primary Key	For Uniquely identify issue of car
Car_ID	INT	Not Null	For identification of car's
Customer_ID	INT	Not Null	For identification of customer
Issue_Date	VARCHAR(200)	Not Null	For saving date when the car issued

4. Table Name :- Car Return

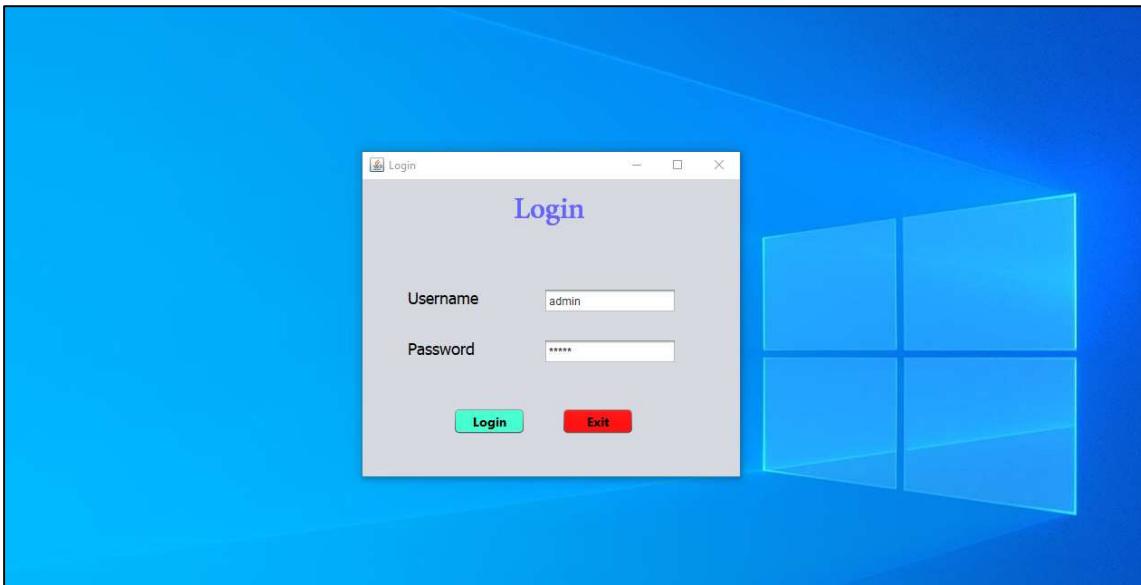
ATTRIBUTE	DATATYPE	CONSTRAINTS	PURPOSE
Ref.No	INT	Primary Key	For Uniquely identify return of car
Car_ID	INT	Not Null	For identification of car
Customer_ID	INT	Not Null	For identification of customer
Return_Date	VARCHAR(200)	Not Null	For saving date when the car returned

INPUT SCREEN/ FORMS DESIGN

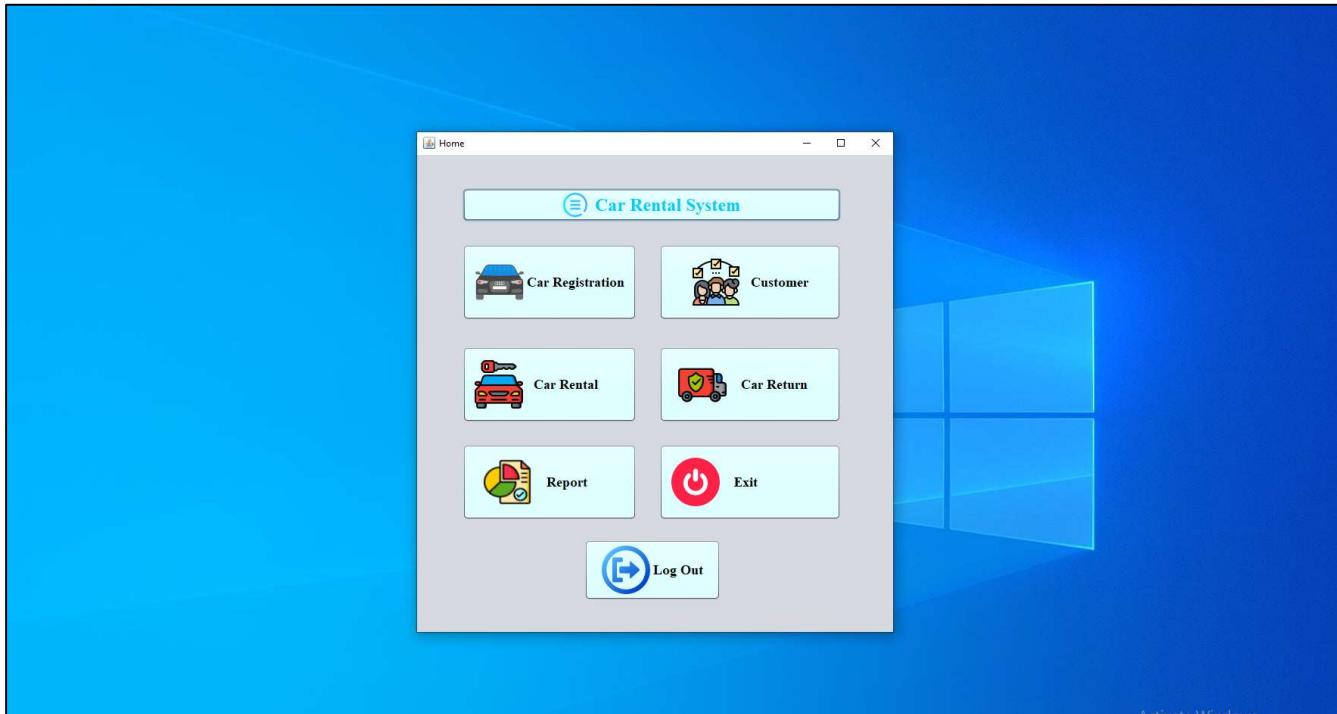


SCREENS:

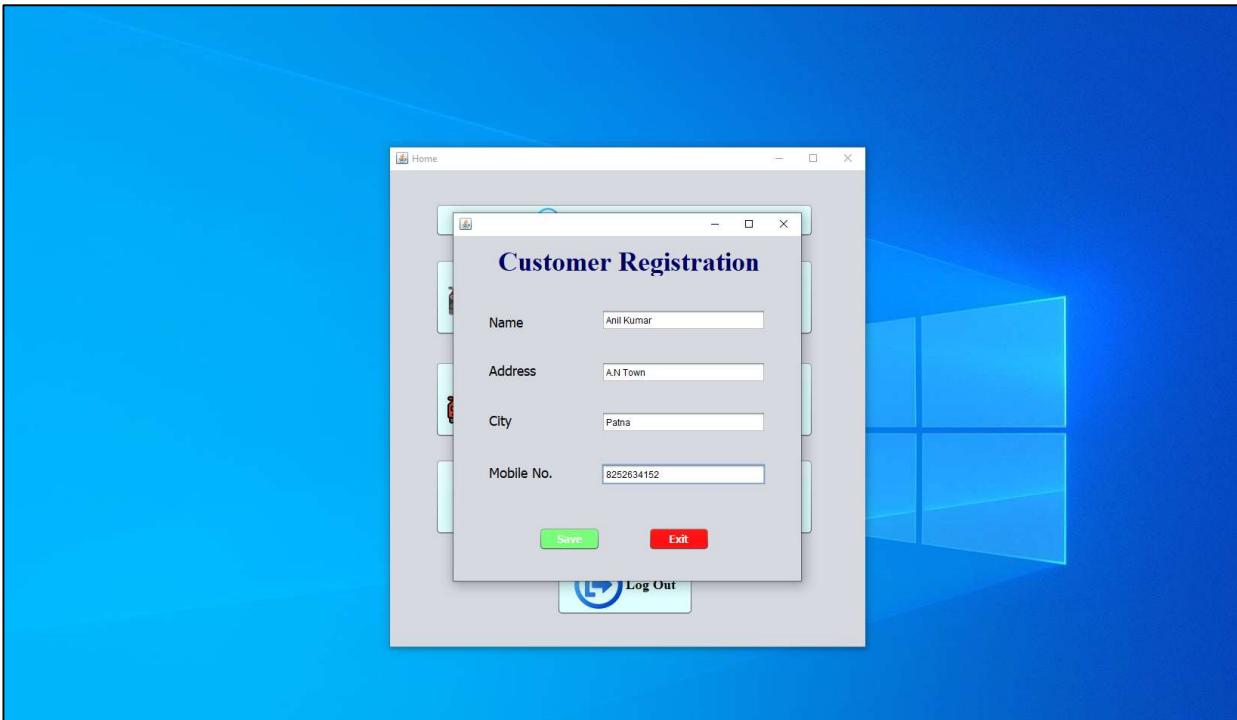
LOGIN



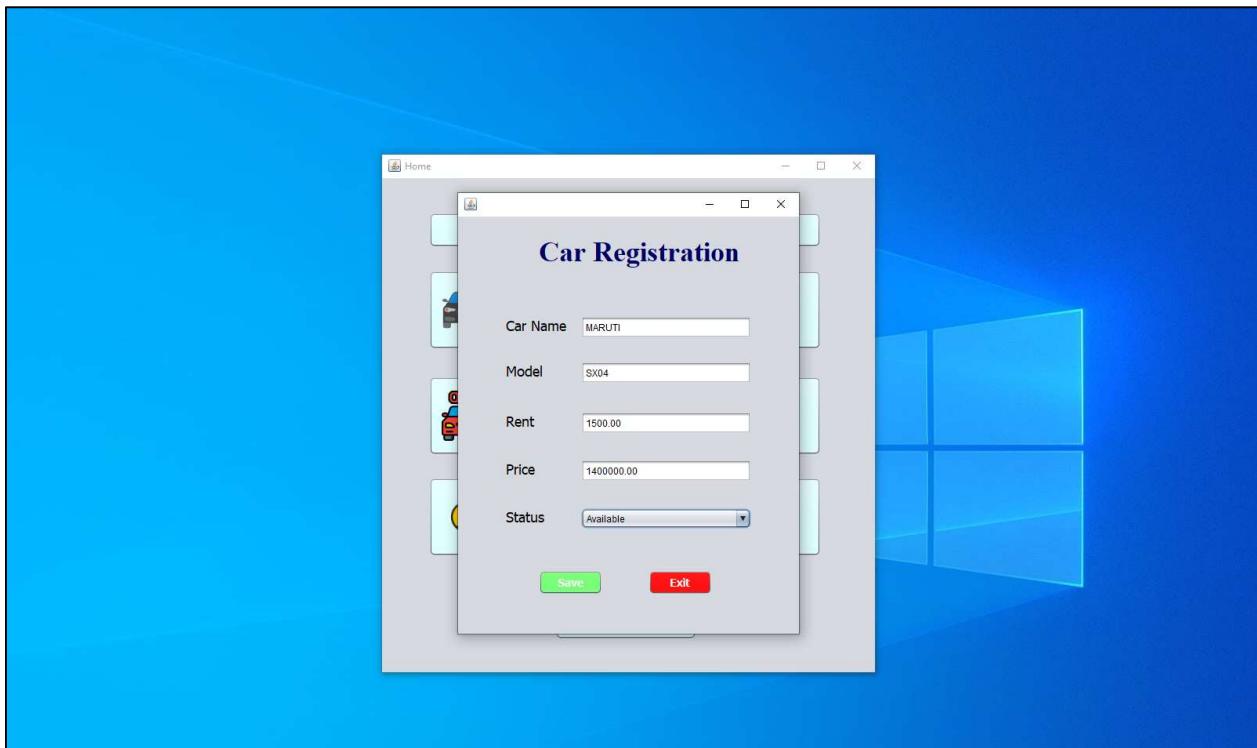
HOME PAGE



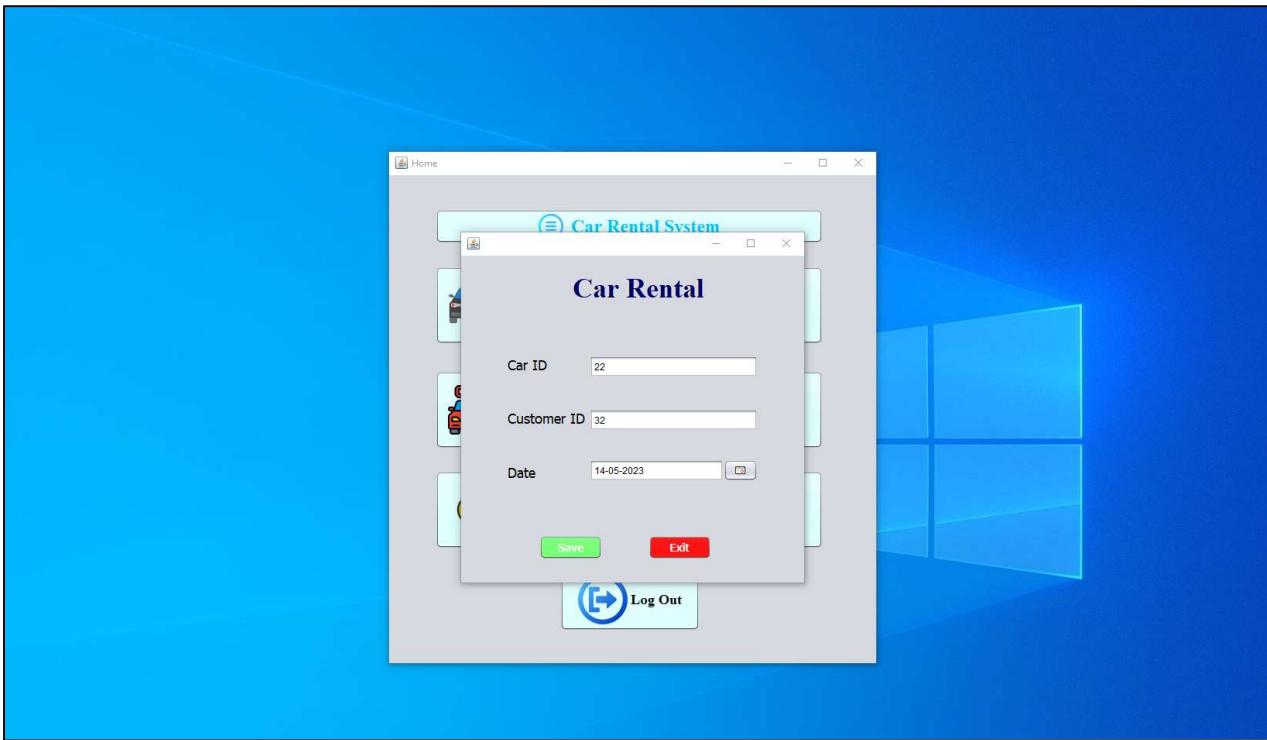
ADD CUSTOMER



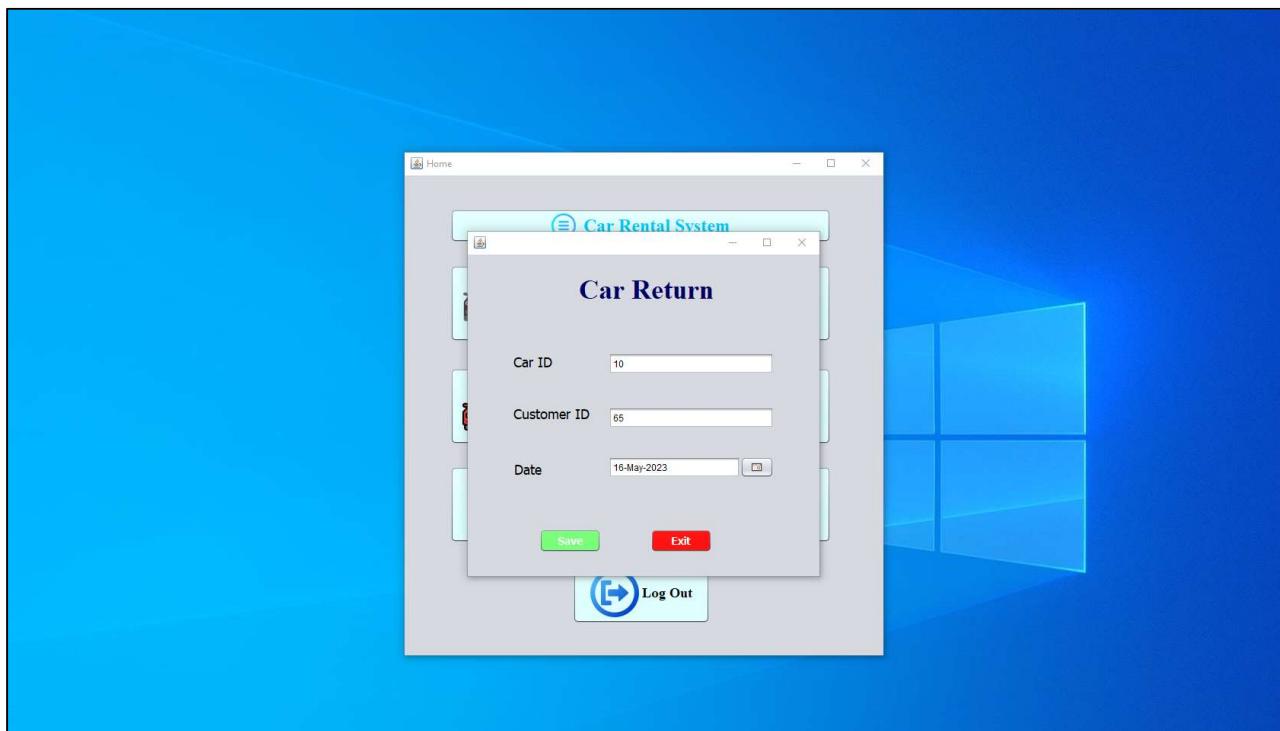
ADD CAR



CAR ISSUE



CAR RETURN

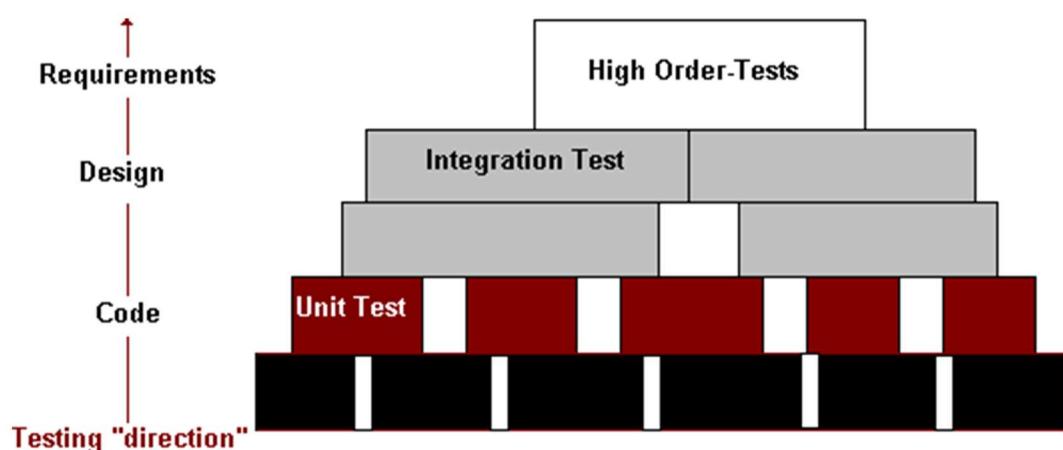


SYSTEM TESTING IMPLEMENTATION & MAINTENANCE



SYSTEM TESTING :-

One of the most important function before actual implementation of system is its testing. Testing is a set of activities that can be planned in advance and conducted system actually. For this reason, a template for software testing should be defined for the software process. Testing is vital to the success of the system. System testing makes a logical assumption that all parts of the system are correct and the goal will be successively achieved. Inadequate testing or no testing leads to error that may not appear until month later.



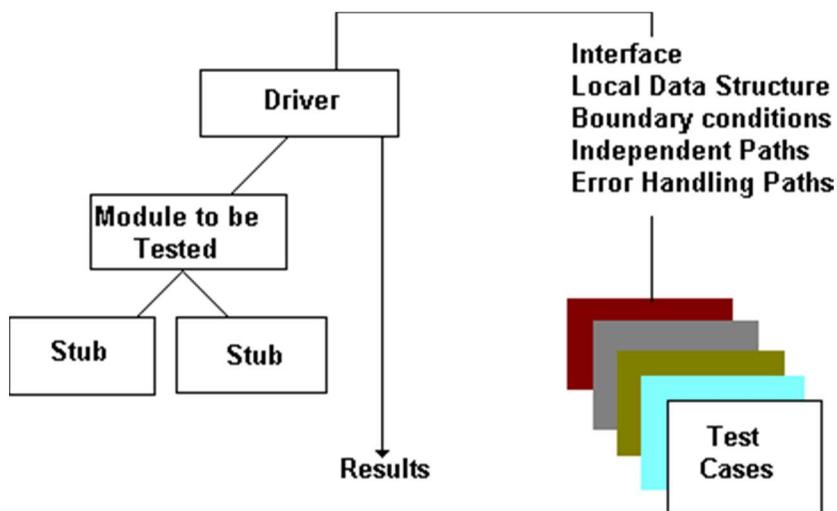
There are several types of testing to check /remove different types of errors as far as possible from the programs. Some very important are-

UNIT TESTING :-

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. The unit test is white-box oriented and the step can be conducted in parallel for multiple components.

Unit testing is normally considered as an adjunct to the coding step. After source level code has been developed, reviewed and verified for correspondence to component level design, unit test case designs begin. A review of design information provider guidance for establishing test cases that is likely to uncover errors in each of the categories. Each test case should be coupled with a set of expected results. Because a component is not a stand-alone program, driver software must be developed for each unit test. Driver accepts test case date, passes such data to the component and prints relevant results. After passing test case data

to the component, expected results are obtained.



INTEGRATION AND SYSTEM TESTING : -

When the individual program modules were working properly, we combined the module into a working system. This integration is planned and co-ordinated so that when an error occurs, we have an idea of what caused it. Integration is the process of verifying that the components of the system work together. For testing the entire system was viewed as a hierarchy of modules. We began with the highest level of design and worked down. The next modules to be tested are those that call the

previously tested modules. For example, variables used for record sets and connections that are frequently called across forms from modules.

Function Test

Once we were sure that information is passed among modules according to the design prescription we tested the system to assure whether the function described the requirement specification were actually performed by the integrated system.

Acceptance Test

When the function test was completed, we involved the user to make sure that the system worked according to user's expectation. Thus the user did the final acceptance test.

Installation Test

When acceptance test was completed, the acceptance system was installed in the environment in which it would be used and a final installation test was performed to make sure that the system function as it should.

WHITE BOX TESTING :-

Also called 'Structural Testing / Glass Box Testing' is used for testing the

code keeping the system specs in mind. For this Developers Test following test

- **Mutation Testing**

A kind of testing in which, the application is tested for the code that was modified after fixing a particular bug/defect. It also helps in finding out which code and which strategy of coding can help in developing the functionality effectively.

- **Control Structure Testing**

The Flow of control execution path is considered for testing. It does also checks
Conditional Testing , Branch Testing, Domain Testing, Data Flow Testing ,Loop testing such as Simple, Nested, Conditional, Unstructured Loops.

BLACK BOX TESTING :-

Also called 'Functional Testing' as it concentrates on testing of the functionality rather than the internal details of code.

Test cases are designed based on the task descriptions

- **Comparison Testing**

Test cases results are compared with the results of the test Oracle.

- **Graph Based Testing**

Cause and effect graphs are generated and cyclometric complexity considered in using the test cases.

- **Boundary Value Testing**

Boundary values of the Equivalence classes are considered and tested as they generally fail in Equivalence class testing.

IMPLEMENTATION : -

Once the system was tested satisfactorily, and then comes the implementation of the system. Implementation is the process of changing from old system to new system. But before implementing the new system, all the data should be transferred from the old system to the new system. In our case as the existing system was a manual one without a homogeneous data source, it was thought prudent to enter the master table thorough various formatted forms. Efforts are on to complete the master table creation and implement the system in full.

MAINTENANCE : -

It has been estimated that maintenance of any Software product usually requires much more effort than the effort necessary to develop the product. Many studies indicate that the relative effort of development of a typical system to its maintenance effort is roughly in the 40:60 ratios. Maintenance involves performing any one or more of the following three kinds of activities:

Correcting errors that were not discovered during the product development phase. This is called corrective maintenance. Improving the implementation of the system and enhancing the functionality of the system according to the customer's requirements. This is called perfect maintenance. Porting the Software to a new environment, e.g. to a new Computer or to a new Operating System. This is called adaptive maintenance. In the proposed project we had done corrective maintenance and adaptive maintenance.

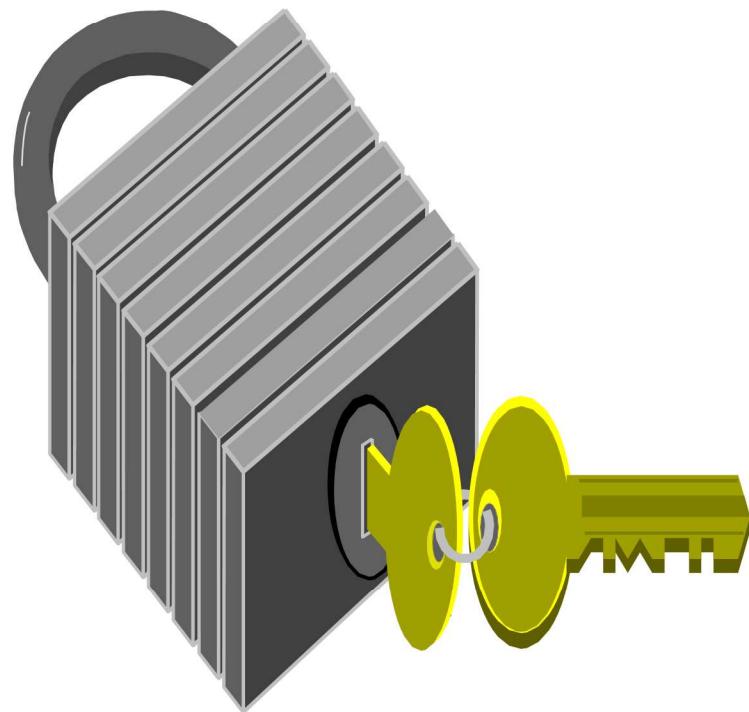
CODE OPTIMIZATION : -

When I was creating this application in Java then I expect to give the best results using minimal computer resources such as memory and hardware space. In order to ensure that my application uses the minimal resources I tried time to

time it for improving its performance. I started my process of optimization from the designing stage itself and continued till the deployment and distribution stage. I kept in my mind this theory that "A well-designed application is the first step towards an optimized application". Following are the most important factors, which support me to develop this application according to the requirement of users.

The speed with which an application executes is one of the most important parameters to measure its performance.

SECURITY CONCERN/MEASURES



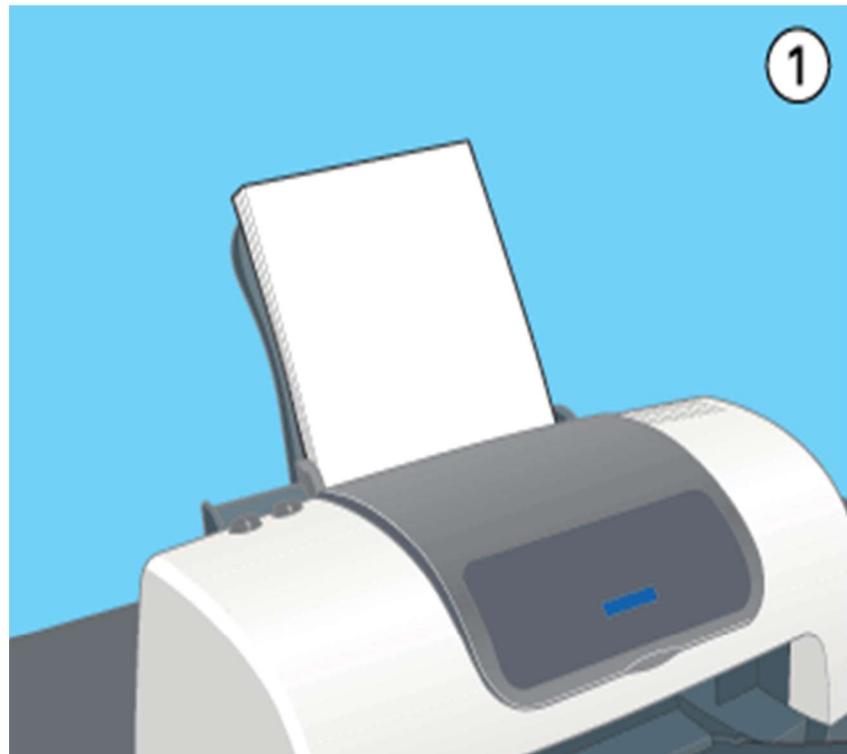
NEED OF SECURE :

A database must have solid security system to control the activities that can be performed and the information that can be viewed and modified. A solid security system ensures the protection of data stored, regardless of how users gain access to the database.

SECURITY LEVEL :

A password is assigned to limit access to a project. Password prevents access to the protected item/project by unauthorized users. As type a password, typing character displays in the form of asterisk (*). Password is also case sensitive

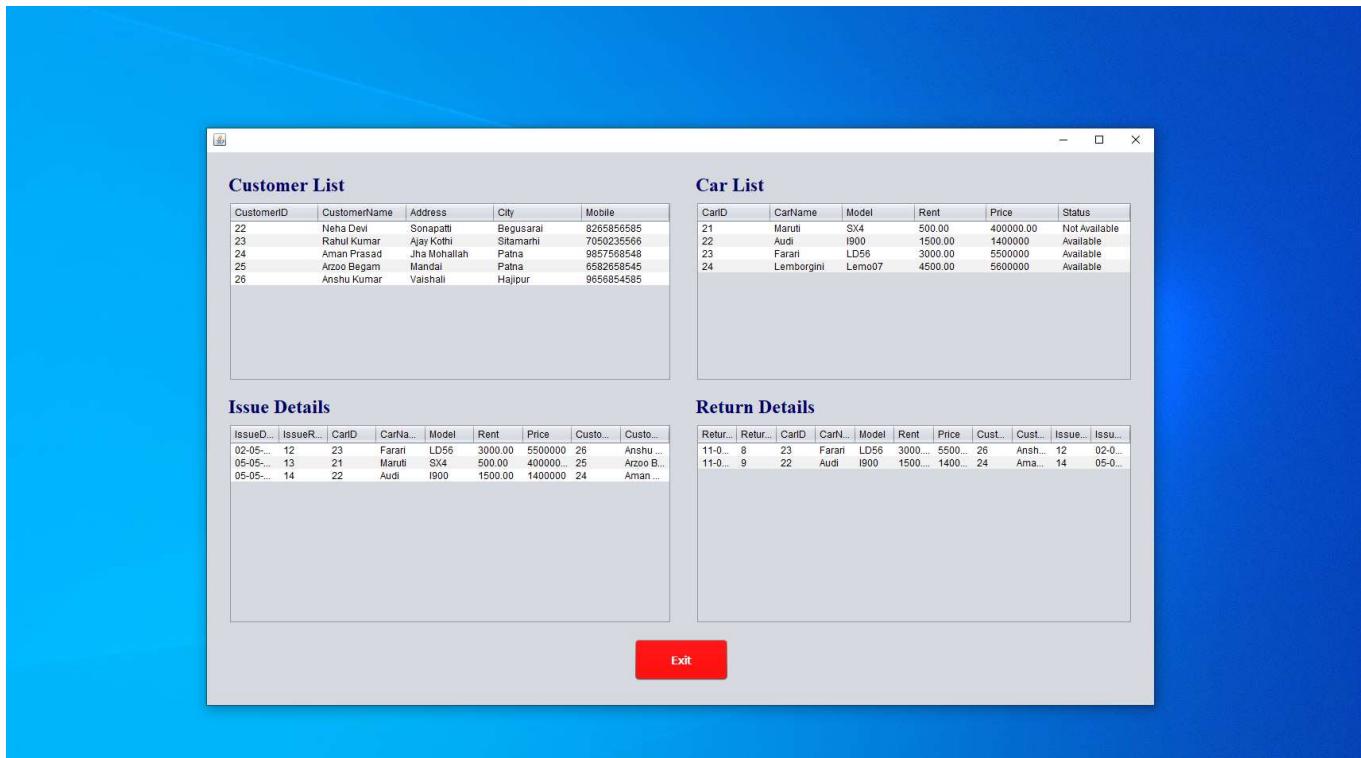
OUTPUT SCREEN / REPORT DESIGN



The following types of reports will be generated, thus fulfilling the need of Management -

- Customer List
- Car List
- Car Issue Details
- Car Return Details

REPORT



LIMITATIONS AND FUTURE SCOPE



LIMITATIONS AND FUTURE SCOPE :

The development of this software is being done keeping in mind the future scope of this application. I find that it has good prospects in the future also. I find it is getting solved almost all the related work of car rental system. Keeping in mind the needs of the organization, which may crop up in the near future, I have tried and introduced certain features, which may be required by the organization, so at that time implementation can be done without any problems. Right now, I am developing this software keeping in mind just of a general current working mechanism of the car renting companies. I am sure It will run successfully, and it can be distributed to other similar working mechanism organizations too. The flexibility and the number of reports, that this software would be providing, also enhance the scope of the software too much.

This educational project is designed to display almost all the related functions of the car rental system. The java system will computerize the activity of the car renting system and maintenance and various other related work.

This project is being developed according to the present time requirement or demand of the car rental system which will be more useful to manage, and regular work of the car renting companies properly. In it, admin has total picture related to all the activities of the company, which will help to take any correct and perfect decisions and further improvements about the system.

In other words, this project is being developed to fulfill requirement of car rental companies.