***CHAPTER***

***– 1 – INTRODUCTION***

**1. Introduction**

Transportation facility is a matter of headache for those people who do not have any personal transport in city. On occasions like Wedding, Vacation, home shifting, tour outside the city and on many other situations they feel the necessity of a vehicle to sort out the problems. So if it is possible to design or develop a web based application for availing transport whenever and wherever possible, then it will be beneficial for both customer and transport provider. Now a days, by some clicks only, we can get whatever you want at home. We already know about the online shopping, e-banking etc. Similarly, The Car Renting Service is the online facility to book cars online within few clicks only. Some people can not afford to have a car, for those people this system becomes very helpful. This system includes various cars, as per the customer order and comfort, it place the order and deliver the car as per the location within the area. For travelling a long distance, booking can be done via internet service only.

This Car Renting Service is very user friendly. It will simplifies the task and reduce the paper work and it is anticipated that functions of the system will be easily accessed by admin, customers and drivers. Admin who has full rights who can perform any type of operations in the project such as adding a new car, change the car rent or change the car details etc. and the customer can book the car online according to their expectation and hire the cars from their specific location anytime and anywhere. And the third user is driver who take a payment by the customer as the generated bill from the website and upload the payment status on the website.

This project also have to facility to check their customers and their payment mode and status details along with date and time. First time customers will have to create a profile if they are taking a car on rent and select the appropriate payment mode. However customers are taking this service by visiting the office, they will get their id and password. Customers will have the facility to select any type of car, search car by their brand name. Every client has been given unique ID and password. This system work 24×7 because of it’s an online existence and this is also helpful for those customers who want to see our newsletter without signing.

This project Car Renting Service has been developed in Java, JSP and database is MySQL which runs over the Apache Server. This is the most complete web portal for managing car renting business. It is a system designed specifically for medium and small vehicle rental businesses. Because of the variety of sizes of their vehicles, car rental agencies may also serve the self-moving industry needs, by renting vans or trucks, and in certain markets other types of vehicles such as motorcycles or scooters may also be offered.

**1.1 Purpose for the system**

The advancement in Information Technology and internet penetration has greatly enhanced various business processes and communication between companies (services provider) and their customers. Car Renting Service is developed to provide the following services:

**1.1.1 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).

**1.1.2 Online Vehicle Reservation:**

A tools through which customers can reserve available cars online prior to their expected pick-up date or time.

**1.1.3 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.

**1.1.4 Group bookings:**

Allows the customer to book space for a group in the case of weddings or corporate meetings (Event management).

**1.1 Overview and issues involved :-**

The main aim to develop this website to get a powerful online car renting software to raise the business profitability.

**1.4 Aims & Objectives**

**Specific goals are –**

• To produce a web-based system that allow customer to register and reserve car online and for the company to effectively manage their car renting business.

• To ease customer’s task whenever they need to rent a car.

**2.1.2 Product functionality**:

Car Renting Service provides the features for booking a car online. It includes several functionalities describes as below:

**2.1.2.1 Car Rental Management:**

It provides car reservation facility online. Customer can visit the website and check for various cars. If they are feasible with requirement, then booking can be done.

**2.1.2.2 Checking For Availability:**

Administrator can check for the availability of the car. He maintains the database of car. If no any car is available it is the responsibility of the administrator to provide alternative options.

**2.1.2.3 Payment system:**

Administrator/owner of the application or the driver responsible for payment from the customer. Booking cancellation, booking confirm, these all activities are done by the administrator of the application.

**2.1.2.4 Maintenance Manager:**

If any car requires maintenance like repair or replacement of any parts, then maintenance manager maintain the data about that. Payment of maintenance are done by the the owner of application in offline mode.

**2.2 Benefits of Online Car Rental** **Services**

• This online car rental solution is fully functional and flexible.

• It is very easy to use and eco-friendly.

• This online car rental system helps in back office administration.

• It saves a lot of time, money and labor.

• The monitoring of the vehicle activity and the overall business becomes easy and includes the least of paper work.

• It increases the efficiency of the management at offering quality services to the customers.

It provides custom features development and support with the application.

**2.1 How Car Rental Services Work**

A car rental is a vehicle that can be used temporarily for a period of time with a fee. Renting a car assists people to get around even when they do not have access to their own personal vehicle or don't own a vehicle at all. The individual who want to rent a car must first contact the car rental company for the desire vehicle. This can be done online. At this point, this person has to supply some information such as; dates of rental, and type of car. After these details are worked out, the individual renting the car must present a valid Identification. Most companies throughout the industry make a profit based of the type of cars that are rented. The rental cars are categorized into economy, compact, compact premium, premium and luxury. And customers are free to choose any car of their choice based ontheir purse and availability of such car at the time of reservation.

**Objective**

The general objective of this project is to develop a web service that will automate the renting process of the company and be able to store information in a single database.

**1.2 Problem definition :-**

Now-a-days it is too difficult to maintain a user records and to share the data from multiple system in multi user environment, there is a lot of duplicate work, and more chances of mistakes. This are some common problems which are as follows :-

1. There are many customers who want to take their own drive across the city as they are uncomfortable of having a driver.

2. Today customers are very busy and they do not really like the idea of visiting the vehicle rental offices to book their cars.

3. Sometimes customer needs space in the car that is not possible in taxi or bus.

4. Customer can't make his own choice in a taxi or bus.

5. The customer does not have their own car for travel.

**1.3 Proposed solution :-**

There is a solution of those problems which are discussed above. Our website solves all the above mentioned problems :-

1. A registration form is provided to users who want to hire a car on rent using our website.

2. Customers fill up the form on the website by providing essential information and register themselves and after registration a unique user id is provided to them.

3. Using this unique user id, customer can login to their account.

4. Using the provided account customer can view various cars and book the car as per their requirements.

5. The additional functionalities of updating user information, password recovery, posting feedbacks, password updation etc are also provided to users.

6. The guest users of the website can view the car details and subscribe our newsletter.

7. Customers can reserve available cars online prior to their expected pick-up date or time.

8. Administrator/Owner of the website and driver can monitor their transactions and use the same to offer better and improve services to them.

9. Allows the customer to book space for a group in the case of weddings or corporate meetings (Event management).

**1.6 Organization of Report:-**

This thesis consists of six (6) chapters. The whole website has designed after a careful study was done on the following points:

**Chapter 1: Introduction.**

This chapter is the introduction about the project that had been developed. It consists of background, problem statement, objective, scope, and report organization.

**Chapter 2: Literature Review.**

This chapter explained the case study of the project. These are two general structures of this study, the technique that has been used and the former system that already created. This chapter provided the literature review that is related with that will be developed later. This chapter comprises two sections: The first section reviews about the existence of other systems. The second section describes the review on method, equipment, and

technology.

**Chapter 3: Analysis:-**

A preliminary investigation was done through which we were able to decide how and for whom the system has to be designed? In this few questions like need for such system, problem in existing system, to whom this system will be useful and in what conditions the system will be useful, all these questions were answered. In this we studied about the feasibility of the system which was further divided into three categories which are economical, technical and operational. We also did the requirement analysis and the specifications were noted and worked upon to find an optimal solution. In our report a proper Entity-relationship diagram and class diagram has been shown to understand the working of the system.

**Chapter 4: Design:-**

The outer as well as inner layout of the system is covered under this heading. The architectural design, system design and the interface designs are key components which are nicely presented in pictorial way.

**Chapter 5: Implementation and Testing.**

This chapter discuss on how the system Car Renting Service had been developed in development environment structurally and logically. In the report, under this heading we have only two main features they are testing and result set. Here we have mentioned what kind of testing is being used and how we have tested the designed system. This chapter discussed on the results or output produced as expected and the result is further discussed.

**Chapter 6: Conclusion.**

This chapter concludes about the entire system.

**1.7 Implementation Language:-**

**Technology Stack :-**

|  |  |  |
| --- | --- | --- |
| 1. | Front End | HTML5,CSS3 and Javascript 1.8(Bootstrap 4.0). |
| 2. | Client-side validation | Javacsript 1.8 |
| 3. | Core Technology | JAVA SE 8 |
| 4. | Presentational Tier | Java Server Pages(JSP) 2.2 & Servlet 3.0 |
| 5. | Database tools | MySQL 5.6 & SQLyog 12.4 |
| 6. | Web Server | Apache Tomcat |
| 7. | Other tools | Netbeans IDE 7.1 |

**1.7.1 Brief Overview of the technology :-**

**1.7.1.1 Front end**:- HTML, CSS, JavaScript, Bootstrap.

1. **HTML**:- HTML is the program that is used to create and generate html documents.
2. **CSS**:- It is a style sheet language, that is used to describe the look and format of a web document written in any markup language.
3. **Java Script**:- It is dynamic computer programming language. JavaScript is mostly used as a part of web browser.

**1.7.1.2 Back end**:- JSP, Servlet, Mysql, SQLyog.

1. **JSP**:- Hypertext Preprocessor, PHP is a general-purpose scripting language that is especially suited for [web development](http://en.wikipedia.org/wiki/Web_development). PHP generally runs on a [web server](http://en.wikipedia.org/wiki/Web_server).

2. **Servlet**:- It is basically a software which provides a virtual environment for dynamic server side web applications.

3. **MySQL**:- MySQL is a Database, widely used for accessing query & managing data in databases.

Other tools :- Netbeans, Apache Tomcat.

1. Netbeans (1-2 lines ka description)

2. Apache Tomcat

***CHAPTER***

***– 2 – LITERATURE SURVEY***

**2.1 Literature Survey:-**

2.1.1 Existing system function:

Most companies throughout the industry make a profit based on the type of cars that are rented. The rental cars are categorized into economy, compact, compact premium, premium and luxury. And customers are free to choose any car of their choice based on their purse and availability of such car at the time of reservation.

Car Rental System gives car rental service for both foreign and local customers. This organization carries out its daily work by providing; their service to the customers using manually system. The organization uses a manual system for reserving, renting, register and to keep record of all the rental activities and customer information. The detailed existing system functions are listed as follows –

• During car reservation the customers book a car by making a phone call to the company otherwise they are expected to go to the company to make booking.

• During renting a car the customer personal information, payments status and rent agreements are filled in the car rent agreement form in order to hold legal contract between the customer and company for renting the car.

• The company normal work time schedule is from 10:30am – 6:00pm; therefore the company gives services for ten and half hours a day.

• The company makes a general report about the rented cars once at the end of the month and generates a report.

**An existing system can provide manually paper work.**

**The user has to go in the office where user can get the car on rent and book their car.**

**In the existing system you can not provide feedback of the user to the admin online.**

**The existing system only provides text-based interface, which is not as userfriendly as Graphical user Interface.**

**Since the system is implemented in Manual, so the response is very slow.**

**The transactions are executed in off-line mode, hence on-line data capture and modification is not possible.**

**Off-line reports cannot be generated due to batch mode execution.**

**Hence, there is a need of reformation of the system with more advantages and flexibility.**

** Details are stored in papers**

** Maintenance is a huge problem**

** Updations,changes in details is a tedious task**

** Performance is not achieved up to the requirements**

** In the present system,company do maintain a person for the allocating and proper functioning of transportation.**

**Proposed system**

The Car Rental System eliminates most of the limitations of the existing system. The proposed system overcome the problems in the Existing system.It has the following objectives:

 Data is centralized which has overcome the sharing problems in previous system.

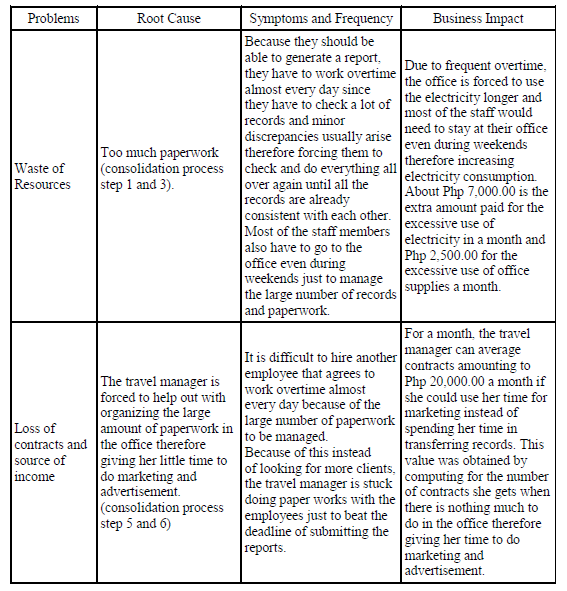
 As data is maintained electronically ,it’s easy for a person to update the details,which has overcome the tedious updations in previous system

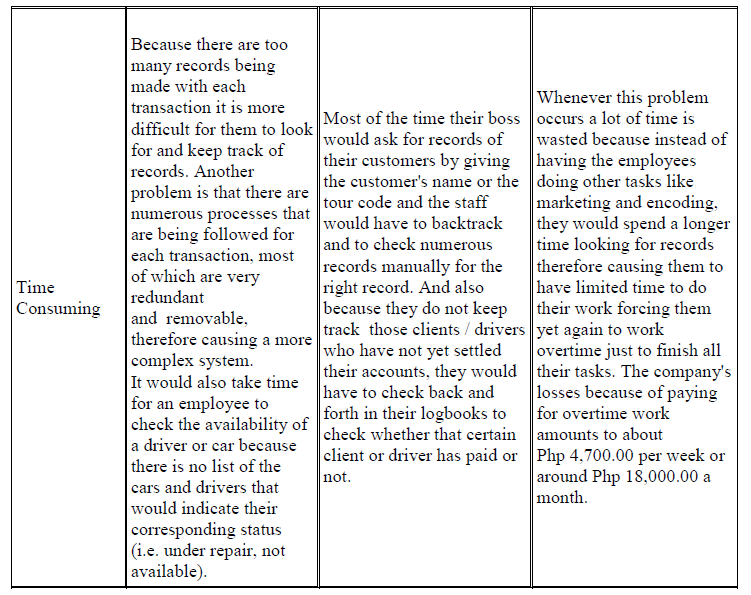
 Maintenance is easy and performance is good

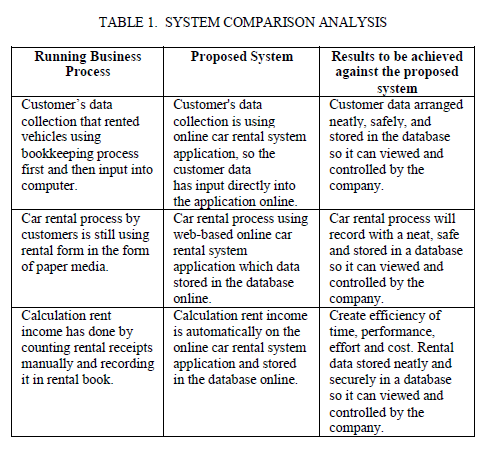
 Mainly the system is automated the transporation process

* The new system is totally computerized system.
* A new system provides features like time efficiency to show car details, user profiles and whatever the customer will give the feedback to the admin.
* This system provides tourism and travelling facilities.
* An inquiry is easily done by user in the system.
* It is the most software application for managing online car rental business.

**1.1.3. Problems to be addressed**

****

****

****

**2.4 Methodology:-**

It is based on Client-Server Technology. Client/Server computing provides the capability to use the most cost – effective user interface, data storage, and connectivity and application services. Frequently, Client/Server products are developed within the present organizations but are not used effectively. The client/Server model provides the technological means to use previous investments in concern with current technology options. There has been a dramatic decline in the cost of technology component of Client/Server computing. Organizations see completions in the market place further increase the need to take the advantage of benefits available from applications build on Client/Server model.

**2.2 Technologies and Tools used :-**

**2.1 Methodology :-**

* Client Server Technology Used.

***CHAPTER***

***– 3 – ANALYSIS***

**4.5 Requirement analysis :-**

Requirement analysis is a software engineering task that bridges gap between system level requirement engineering and software design. Requirements analysis provides the software designer with a representation of information. Function and behavior that can be translated to data architectural, interface, and component level designs.

Requirements are feature of a system or description of something that the system is capable of doing in order to fulfill the system’s purpose. It provides the appropriate mechanism for understanding what the customer wants, analyzing the needs, assessing, feasibility negotiating a responsible solution specifying the solution unambiguously , validating the specification and managing the requirement as they are translated into a operational system.

**3.2 Requirement Analysis**

**3.2.1 Software Requirements**

Software is a collection of computer programs and related data that provide the instructions for telling a computer what to do and how to do it. In other words, software is a conceptual entity which is a set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system. We can also say software refers to one or more computer programs and data held in the storage of the computer for some purposes. In other words software is a set of programs.

* **Operating system**:- Windows XP or Higher version.
* **Web Server:-** Apache Tomcat Server
* **Web Browser :**- Google Chrome, Mozilla Firefox, Internet Explorer or any other browser.
* **Database Tool :-** SQLyog.
* **IDE**:- Netbeans.
* **Designing Tools**:- Dreamweaver,Paintbrush
* **Java JDK :** JDK 1.6 or higher.
* **UML Diagram Tool** :- Draw.io, StarUML, Creately or any other online tool.

3.2.2 **Hardware Requirements**

computer hardware are component devices which are typically installed into or peripheral to a computer case to create a personal computer upon which system software is installed including a firmware interface such as a BIOS and an operating system which supports application software that performs the operator's desired functions. Minimum hardware requirement as follows:

Hardware requirements :-

1. RAM : 512 MB.

2. Cache Memory : 512 KB.

2. Processor : Processor(P5) core

3. Server : Any server like Apache tomcat/Glassfish server.

4. HDD : 40GB.

5. CPU : Intel Pentium 4(1GHz) or higher.

6. GPU : Any GPU that is compatible.

7. Other Hardware : Keyboard, Mouse, Monitor, LAN Card etc.

* 1. **Process Model Adopted**

**The system uses “Prototype Model”**

* + 1. **Description**

**SDLC - Software Prototype Model**

The Software Prototyping refers to building software application prototypes which displays the functionality of the product under development, but may not actually hold the exact logic of the original software.

Software prototyping is becoming very popular as a software development model, as it enables to understand customer requirements at an early stage of development. It helps get valuable feedback from the customer and helps software designers and developers understand about what exactly is expected from the product under development.

**What is Software Prototyping?**

Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation.

Prototyping is used to allow the users evaluate developer proposals and try them out before implementation. It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.

Following is a stepwise approach explained to design a software prototype.

* Basic Requirement Identification
* Developing the initial Prototype
* Review of the Prototype
* Revise and Enhance the Prototype

****

* + 1. **Advantages and Disadvantages**

Software prototyping is used in typical cases and the decision should be taken very carefully so that the efforts spent in building the prototype add considerable value to the final software developed. The model has its own pros and cons discussed as follows.

**4.4.2.1 Advantages:**

The advantages of the Prototyping Model are as follows −

• Increased user involvement in the product even before its implementation.

• Since a working model of the system is displayed, the users get a better understanding of the system being developed.

• Reduces time and cost as the defects can be detected much earlier.

• Quicker user feedback is available leading to better solutions.

• Missing functionality can be identified easily.

• Confusing or difficult functions can be identified.

**4.4.2.2 Disadvantages**:

The Disadvantages of the Prototyping Model are as follows −

• Risk of insufficient requirement analysis owing to too much dependency on the prototype.

• Users may get confused in the prototypes and actual systems.

• Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.

• Developers may try to reuse the existing prototypes to build the actual system, even when it is not technically feasible.

• The effort invested in building prototypes may be too much if it is not monitored properly.

**3.1.3 Reasons for Use**

* Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
* Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.
* Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems.

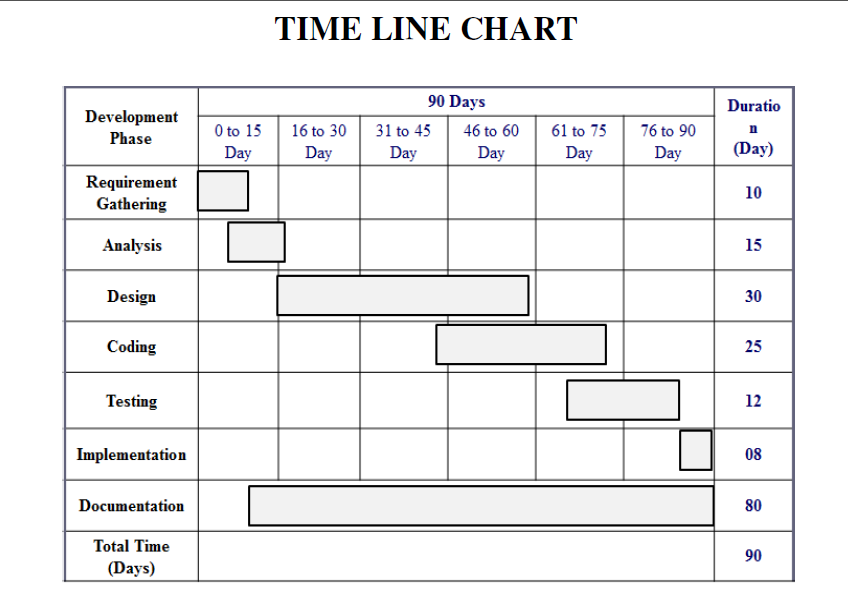
**3.3 Feasibility Study**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

ü Technical Feasibility

ü Operation Feasibility

ü Economical Feasibility



**3.3.1 Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

ü Does the necessary technology exist to do what is suggested?

ü Do the proposed equipments have the technical capacity to hold the data required to use the new system?

ü Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

ü Can the system be upgraded if developed?

ü Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

In technical feasibility the following issues are taken into consideration. Whether the required technology is available or not.

Whether the required resources are available: Manpower- programmers, testers & debuggers, Software and hardware.

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out. For are project we are study all technical resources are available.

**4.8.2 Economical feasibility :-**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system.

In economic feasibility, the most important is cost-benefit analysis. As the name suggests, it is an analysis of the costs to be incurred in the system and benefits derivable out of the system. Click on the link below which will get you to the page that explains what cost benefit analysis is and how you can perform a cost benefit analysis.

**4.8.3 Operational feasibility:-**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

ü Is there sufficient support for the management from the users?

ü Will the system be used and work properly if it is being developed and implemented?

ü Will there be any resistance from the user that will undermine the possible application benefits?

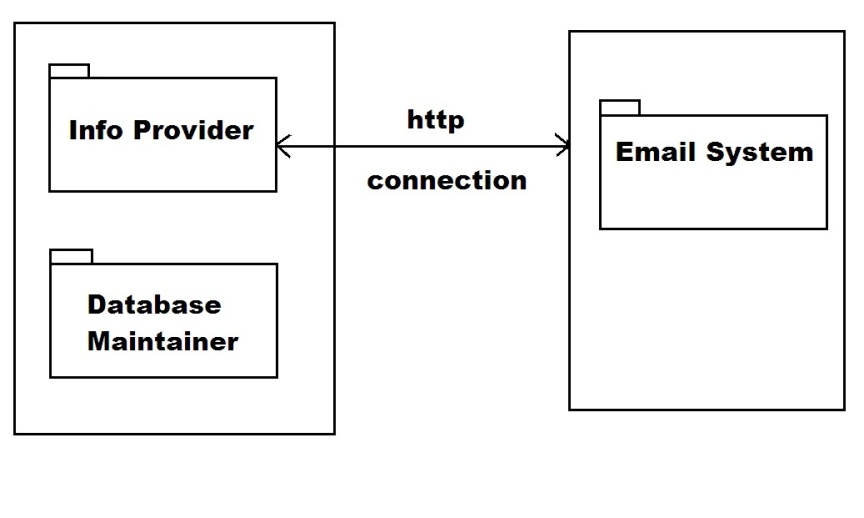
This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits? The essential questions that help in testing the operational feasibility of a system are following.

* Does management support the project?
* Are the users not happy with current business practices? Will it reduce the time (operation) considerably? If yes, then they will welcome the change and the new system.
* Have the users been involved in the planning and development of the project? Early involvement reduces the probability of resistance towards the new system. Our system is operational feasible..

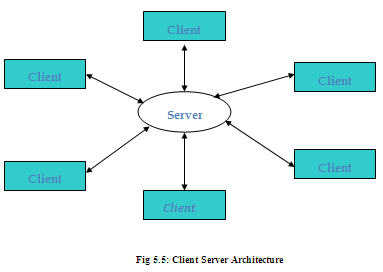
**3.4 Architectural Specification**

The primary objective of Architectural Design is to develop a modular program structure and represent the control relationships between modules. It also melds program structure and data structure, defining interfaces that enable data to flow throughout the program.

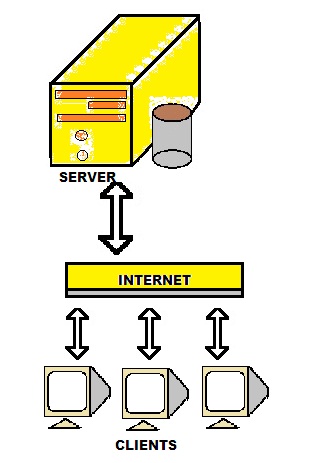


**Fig : E-mailing System Working**

There can be various kind of architectural style of software like- Data Centred Architecture, Data Flow Architecture, Entity Relationship Diagram Call and Return Architecture, Object Oriented Architecture, Layered Architecture, Client Server Architecture, MVC Architecture, Three Tier Architecture / Multi Tier Architecture.



**Client Server Architecture** is used in this system i.e. Desktop Monitoring system. Client server architecture consists of two part Client and server. A server is anything that has some resource to share. There are computer server, which provides computing power; print server which manage collection of printers, disk server which provide network disk space, the Web server which stores web pages. A client is any other entity which want to gain access other entity at server. The interaction between client and server is similar as lamp and electrical socket. Lamp is client which request for electricity and socket is server which server the request and provide electric current. The server is permanently available resource, which the client is free to unplug after it is has been served.

****

3.5 **Use Case Model**

3.6 **Use Case Description**

**6.3.1 USECASE DIAGRAMS:-**

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called an actor. A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor– Sender, Secondary Actor- Receiver.

Use case diagram can be useful for getting an overall view of the system and clarifying what can do and more importantly what they can’t do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

* The purpose is to show the interactions between the use case and actor.
* To represent the system requirements from user’s perspective.
* An actor could be the end-user of the system or an external system.

**4.5.2 Use Case description :-**

The main purpose of a **use case diagram** is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**4.3 Use Case Specification :-**

A use case is a description of a set of sequence of action including variants that a system performance to yield an observable value to an actor a use case describes a set of sequences which each sequence represents the interaction of the things outside the system with the system itself a use case represents functional requirement of your system as a whole a use case describe what a system does but it does not specify how it does.

Actor and use case description shows the detail description of interaction between the actors and their use cases. The description enables to have a proper understanding of how actor interacts with the system through their use cases.

**Fig : E-mailing System (Use Case Diagram)**

***CHAPTER***

***– 4 – DESIGN***

|  |  |
| --- | --- |
| **Chapter-4 Design** | **57** |
| **4.1 Sequence diagrams and Activity diagrams.** |  |
| **4.2 Class Diagram (With relationships, name of the attributes and methods of each class)** |  |
| **4.3 Database Design (Complete E-R diagram with cardinalities and relationships, It must also include the snapshot of each table structure with all its fields and a short description of the content of the table under it.)** |  |
|  |  |

**4.9 Design:-**

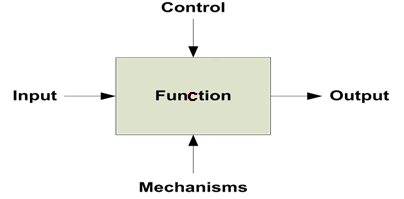
**4.9.1. Design constraints :-**

A constraint is a design target that must be met for the design to be successful. In system design, a design constraint refers to some limitation on the conditions under which a system is developed, or on the requirements of the system. The design constraint could be on the systems form, fit or function or could be in the technology to be used, materials to be incorporated, time taken to develop the system, overall budget, and so on. A design constraint is normally imposed externally, either by the organization or by some external regulation. During system design, it is as important to identify each design constraint as it is to elicit requirements since the design constraints place an overall boundary around the system design process.

**4.9.2. Design technique :-**

Design technique is a diagrammatic notation designed specifically to help people describe and understand systems. It offers building blocks to represent entities and activities, and a variety of arrows to relate boxes. These boxes and arrows have an associated informal semantics.Design technique can be used as a functional analysis tool of a given process, using successive levels of details. The Design technique method allows to define user needs for IT developments, which is very used in the industrial Information Systems, but also to explain and to present an activity’s manufacturing processes, procedures.

The Design technique supplies a specific functional view of any enterprise by describing the functions and their relationships in a company. These functions fulfill the objectives of a company, such as sales, order planning, product design, part manufacturing, and human resource management. The Design technique can depict simple functional relationships here and can reflect data and control flow relationships between different functions.

****

**Fig :Design Technique**

**6.1 Design :-**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

**6.1.1 Design Concepts :-**

The primary objective of Architectural Design is to develop a modular program structure and represent the control relationships between modules. It also melds program structure and data structure, defining interfaces that enable data to flow throughout the program.

There can be various kind of architectural style of software like- Data Centred Architecture, Data Flow Architecture, Entity Relationship Diagram Call and Return Architecture, Object Oriented Architecture, Layered Architecture, Client Server Architecture, MVC Architecture, Three Tier Architecture / Multi-Tier Architecture.

**6.1.2 Design Technologies :-**

Client Server Architecture is used in this system i.e. Desktop Monitoring system. Client server architecture consists of two part Client and server. A server is anything that has some resource to share. There are computer server which provides computing power; print server which manage collection of printers, disk server which provide network disk space, the Web server which stores web pages. A client is any other entity which wants to gain access to other entity at server. The interaction between client and server is similar as lamp and electrical socket. Lamp is client which request for electricity and socket is server which server the request and provide electric current. The server is permanently available resource, which the client is free to unplug after it is has been served.

System development has two major components. System Analysis and System design. System Analysis and Design refers to the process for examining a business situation with the intent of improving it through better methods and procedures. System design is the process of planning a new business system to replace the old. But before this planning can be done, we must thoroughly understand the old system and determine how the computer can be best used to make its operation more effective.

System analysis is the process of understanding the current system by gathering and interpreting facts, diagnosing problems, and using the facts to improve the current system. This is the job of the system analyst.

Having determined the requirements and what is to be done by the system designer in the new system keeping in mind the objective set during the system analysis.

**6.3 UML Diagrams :-**

UML stands for Unified Modelling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

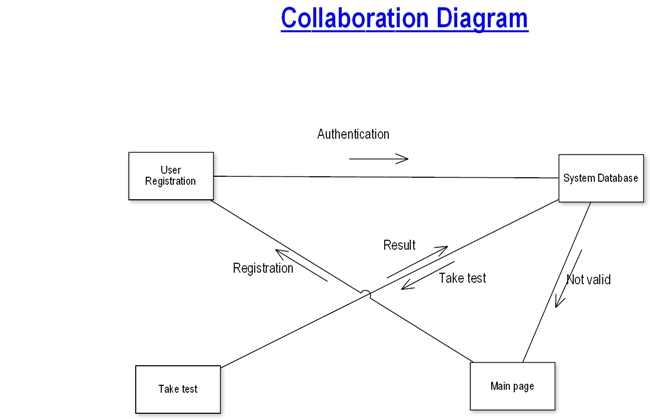
There are various kinds of methods in software design:

They are as follows:

* Use case Diagram
* Sequence Diagram
* Collaboration Diagram
* Activity Diagram
* State chat Diagram

**6.3.2 COLLABORATION DIAGRAM:-**

A collaboration diagram is an introduction diagram that emphasizes the structural organization of the objects that send and receive messages. Graphically a collaboration diagram is a collection of vertices and arcs.



**Fig : Collaboration Diagram**

**6.3.3 CLASS DIAGRAM:-**

Class is nothing but a structure that contains both variables and methods. The Class Diagram shows a set of classes, interfaces, and collaborations and their relating ships. There is most common diagram in modeling the object oriented systems and are used to give the static view of a system. It shows the dependency between the classes that can be used in our system.

The interactions between the modules or classes of our projects are shown below. Each block contains Class Name, Variables and Methods.

**4.5.3 Conceptual level class diagram :-**

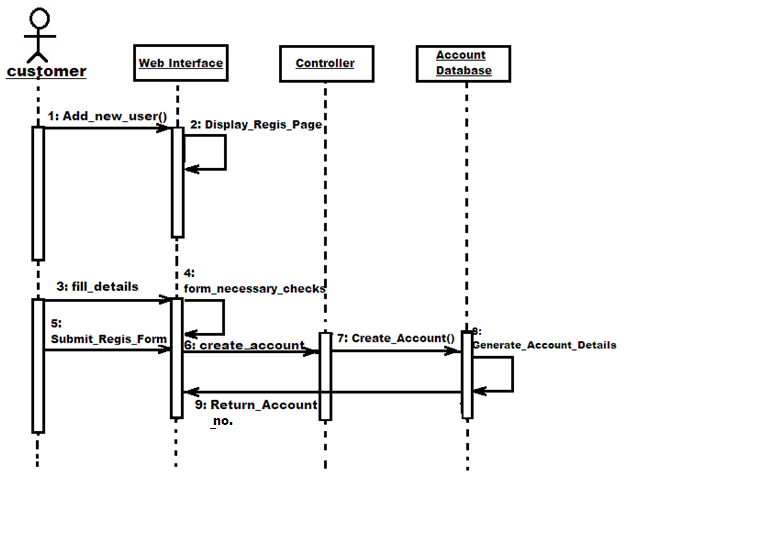
A **class diagram** in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

**Fig : E-mailing System (Class Diagram)**

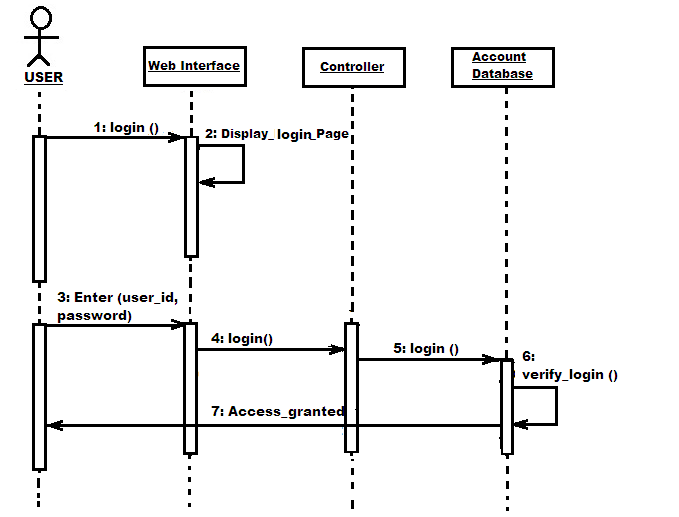
**4.5.4 Conceptual level sequence diagram :-** processes operate with one another and in what order. It is a construct of a Message

A **sequence diagram** in Unified Modelling Language (UML) is a kind of interaction diagram that shows how Sequence Chart. A sequence diagram shows object interactions arranged in time sequence .

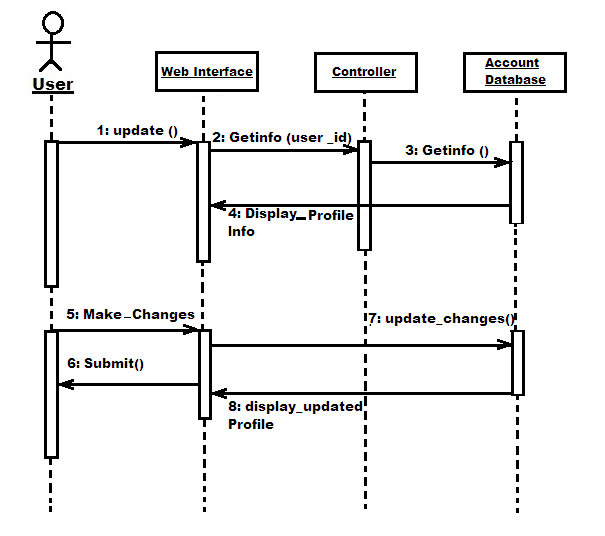
Sequence diagrams are used to demonstrate the behavior of objects in a use case by describing the objects and the messages they pass. It provides a graphical representation of object interactions over time. Sequence diagrams show an actor, the objects and components they interact with in the execution of a use case. One sequence diagram represents a single Use Case 'scenario' or events. Sequence diagrams show the flow of messages from one object to another, and as such correspond to the methods and events supported by an object.



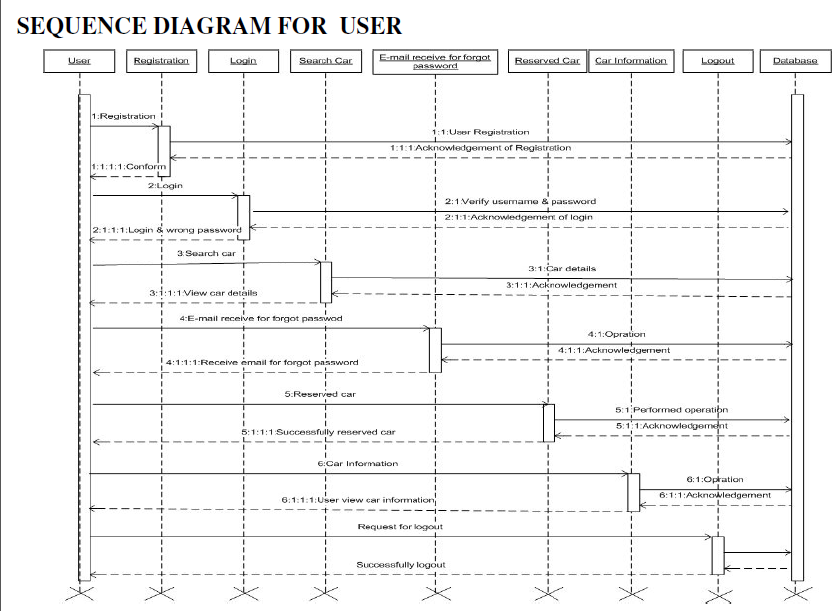
**Fig : Register New User (Sequence Diagram)**

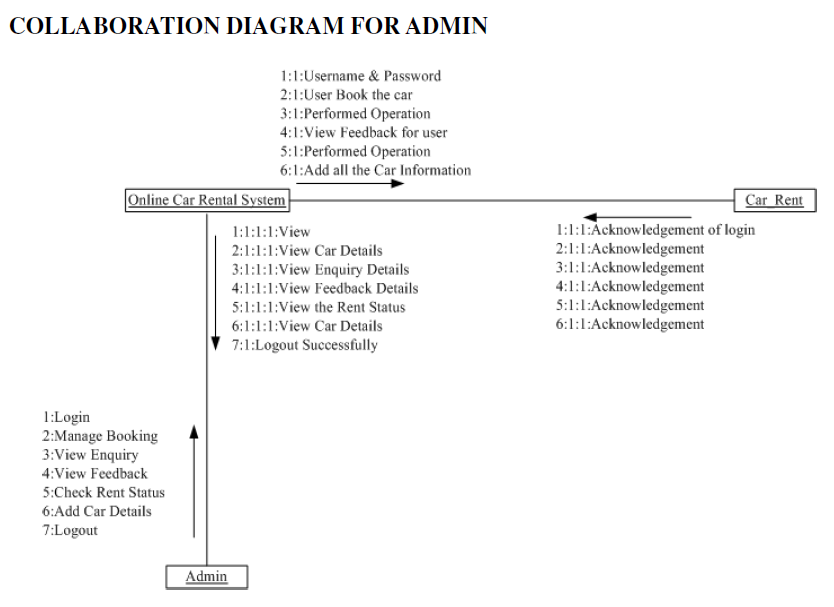


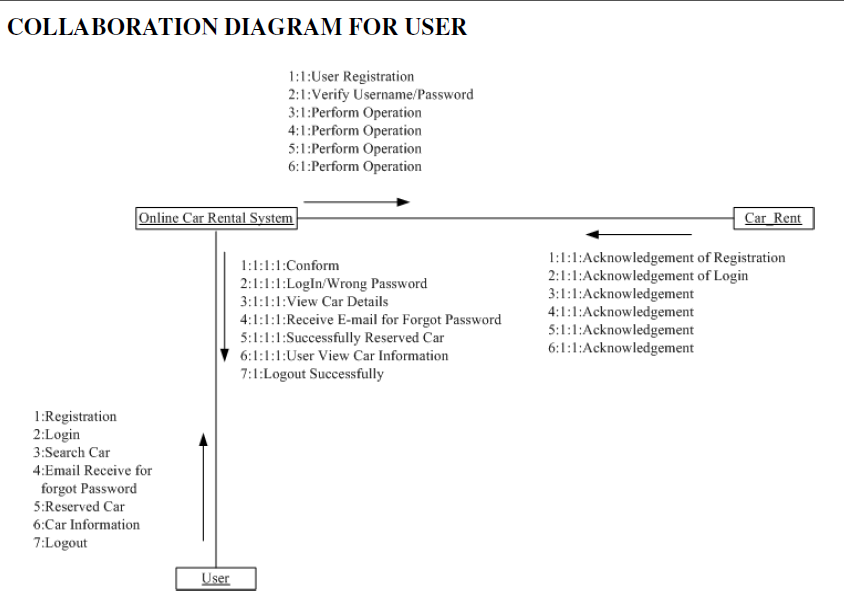
**Fig : Login into System (Sequence Diagram)**

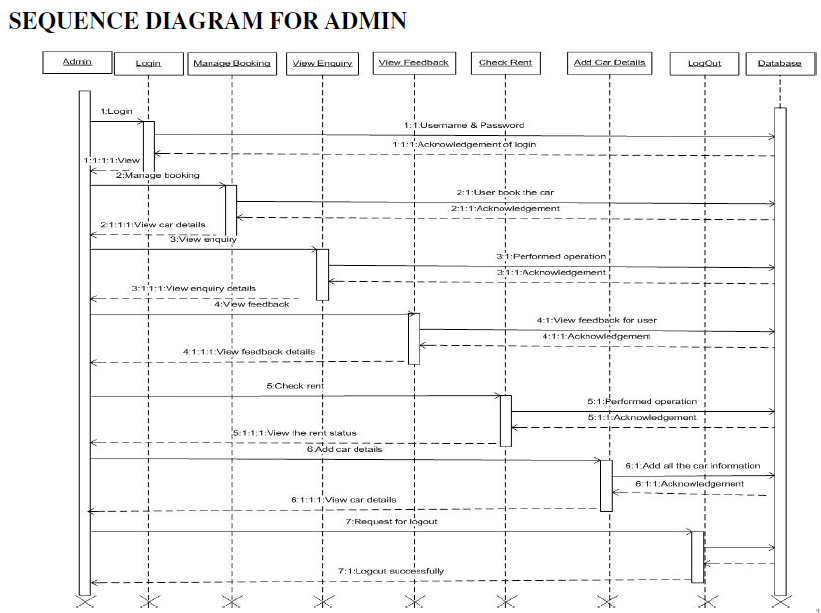


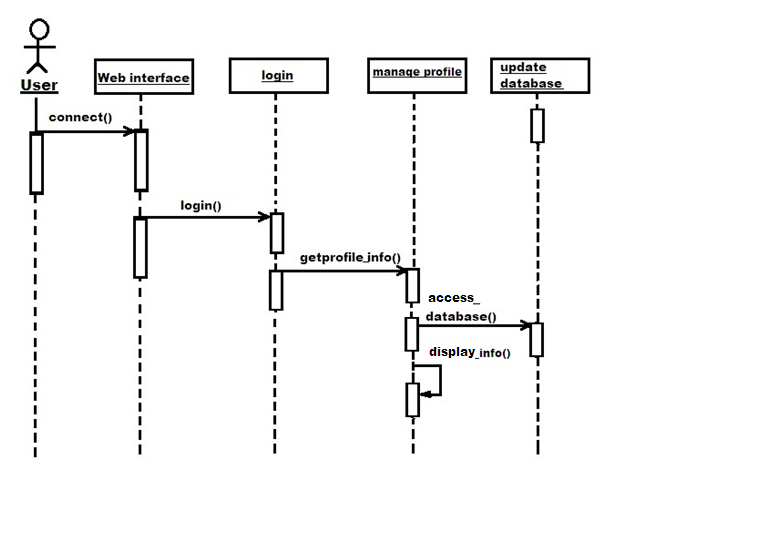
**Fig : Update User Profile (Sequence Diagram)**

****

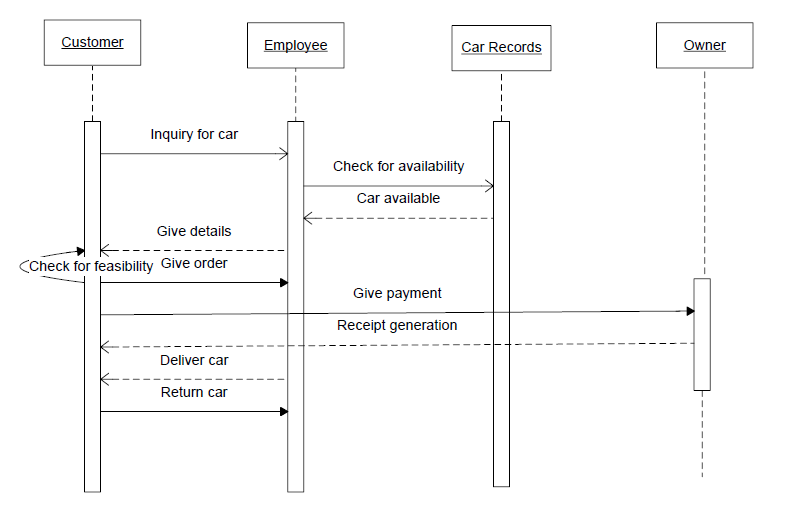
****

****

****

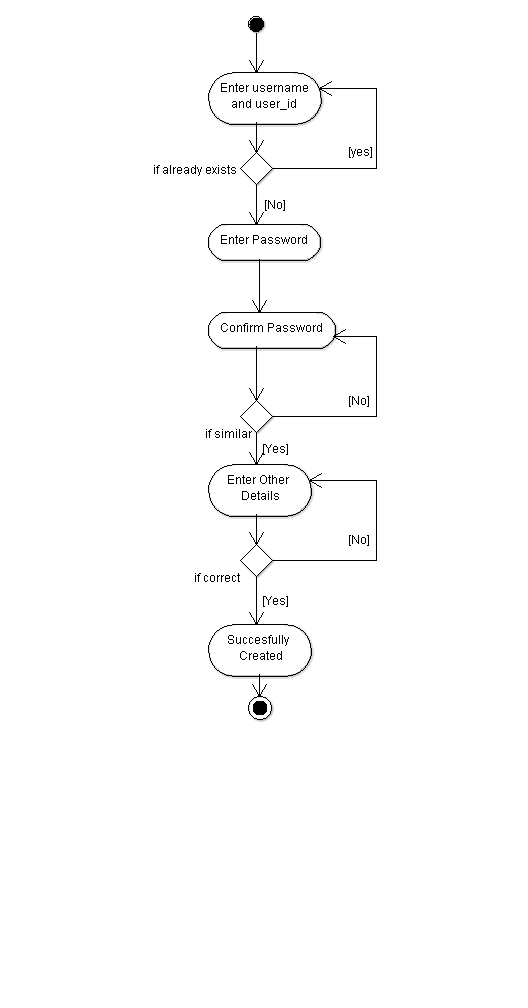


**Fig : Displaying User Information (Sequence Diagram)**

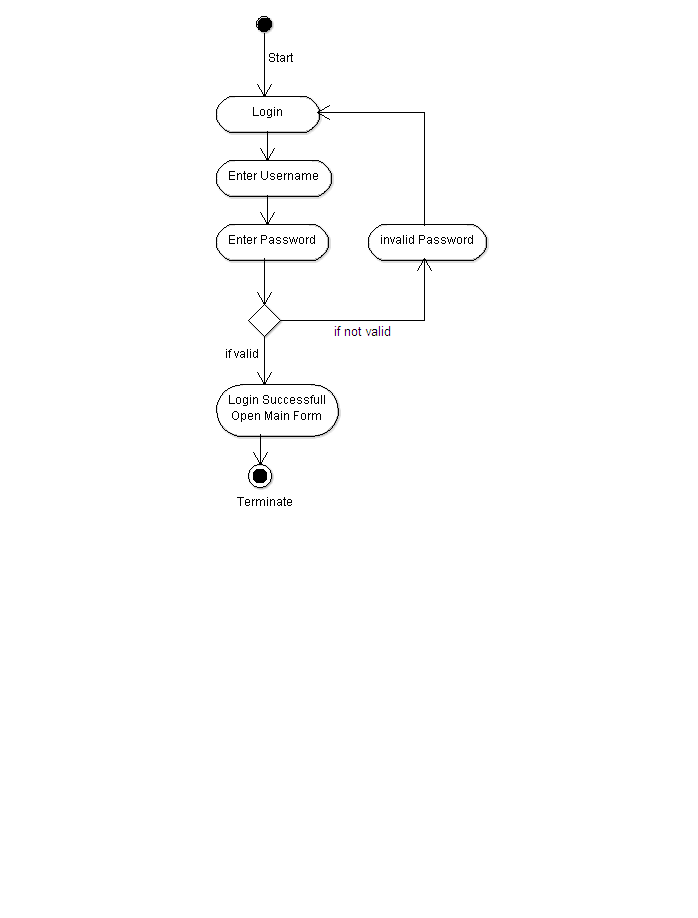


**4.5.5 Conceptual level activity diagram :-**

**Activity diagrams** are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

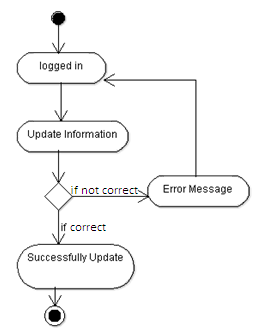


**Fig : Registration (Activity Diagram)**

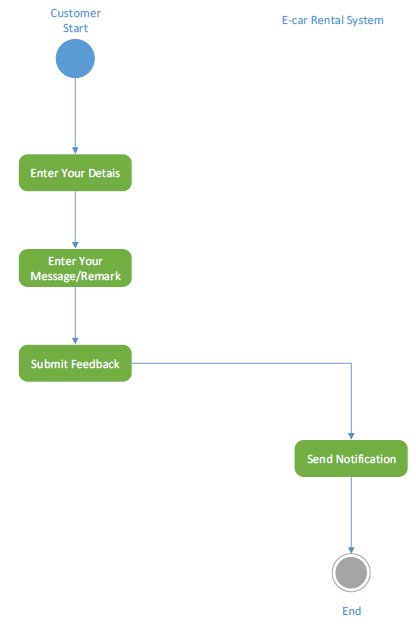


**Fig : Login into System (Activity Diagram)**

**Fig : Sending an E-Mail (Activity Diagram)**



**Fig : Updating User Information (Activity Diagram)**

****

* + 1. **Structured flow diagram :-**
       1. **ER-Diagram :-**

Entity-relationship (ER) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

**Fig : ER Diagram**

* 1. **Structured flow diagram :-**
     1. **ER-Diagram :-**

Entity-relationship (ER) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

10 Entity–relationship model

The entity relationship diagram describes the relationship between entities, cardinality and their attributes. Entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them. In here we provide a description of entities with all their attributes. Describing entity name, business definition for the entities and there attribute and domain.

10.1 Entity Description:

|  |  |
| --- | --- |
| **Entity Name** | **Business definition** |
| Employee | This entity is responsible to store Employee information in the database. |
| Customer | Attribute stores customers’ details information in the database, in order to identify the customer. |
| Car | This entity is stores the information of the vehicle in the database. |
| Reservation | This stores information about the reservations made by a customer. |
| Rent | This stores rental information of the vehicle, payments |
| Maintenance | This checks for repairing and replacing. |
| Payment | This produce payment and rent the car. |

## 7.1.2 ER Notation :-

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

**7.1.3 Conceptual level activity diagram :-**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

* 1. **LOGICAL DATABASES :-**

Different Tables in the Database are as follows :

# Database : mk

**\*\*user\_id is primary key for all tables.**

**4.1 Sequence diagrams and Activity diagrams.**

**4.2 Class Diagram**

**4.3 Database Design**

**DATA DICTIONARY**

# Database caratrent\_db

**Table Name** Admin

**Description** This table is store information about Admin

## Table structure for table tbladmin

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| **UserName** | varchar(100) | No |  |
| Password | varchar(100) | No |  |
| updationDate | timestamp | No | 0000-00-00 00:00:00 |

## Table structure for table tbldriver

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **d\_id** | int(10) | No |  |
| d\_name | varchar(20) | No |  |
| pwd | varchar(30) | No |  |
| d\_add | varchar(30) | No |  |
| d\_contact | varchar(10) | No |  |
| **car\_id** | varchar(10) | No |  |
| ***d\_email*** | varchar(30) | No |  |
| base\_charge | varchar(6) | No |  |
| wait\_charge | varchar(6) | No |  |

## Table structure for table tblpayment

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***p\_id*** | int(10) | No |  |
| userid | varchar(30) | No |  |
| p\_status | varchar(10) | No |  |
| car\_name | varchar(20) | No |  |
| email\_id | varchar(30) | No |  |
| car\_id | varchar(20) | No |  |
| amount | varchar(10) | No |  |
| contact | varchar(10) | No |  |
| book\_date | datetime | No |  |
| p\_date | datetime | No |  |

## Table structure for table tblbooking

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| userEmail | varchar(100) | No |  |
| VehicleId | int(11) | No |  |
| contact\_no | tinytext | No |  |
| s\_add | tinytext | No |  |
| d\_add | tinytext | No |  |
| FromDate | datetime | No |  |
| ToDate | datetime | No |  |
| message | varchar(255) | Yes | NULL |
| Status | int(11) | No |  |
| PostingDate | timestamp | No | CURRENT\_TIMESTAMP |

## Table structure for table tblbrands

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| BrandName | varchar(120) | No |  |
| CreationDate | timestamp | No | CURRENT\_TIMESTAMP |
| UpdationDate | timestamp | No | 0000-00-00 00:00:00 |
| color | varchar(15) | Yes | NULL |
| seat | int(3) | Yes | NULL |
| fuel\_type | varchar(10) | No |  |
| no\_of\_pieces | int(4) | No |  |
| available | varchar(3) | No |  |

## Table structure for table tblcontactusinfo

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| Address | tinytext | Yes | NULL |
| **EmailId** | varchar(255) | No |  |
| ContactNo | char(11) | Yes | NULL |

## Table structure for table tblcontactusquery

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| name | varchar(100) | No |  |
| EmailId | varchar(120) | No |  |
| ContactNumber | char(11) | No |  |
| Message | varchar(200) | No |  |
| PostingDate | timestamp | No | CURRENT\_TIMESTAMP |
| status | int(11) | No |  |

## Table structure for table tblpages

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| PageName | varchar(255) | Yes | NULL |
| type | varchar(255) | No |  |
| detail | longtext | No |  |

## Table structure for table tblsubscribers

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| **SubscriberEmail** | varchar(120) | No |  |
| PostingDate | timestamp | No | CURRENT\_TIMESTAMP |

## Table structure for table tbltestimonial

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| UserEmail | varchar(100) | No |  |
| Testimonial | mediumtext | No |  |
| PostingDate | timestamp | No | CURRENT\_TIMESTAMP |
| status | int(11) | No |  |

## Table structure for table tblusers

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| id | int(11) | No |  |
| FullName | varchar(120) | No |  |
| ***EmailId*** | varchar(100) | No |  |
| Password | varchar(100) | No |  |
| ContactNo | varchar(25) | No |  |
| dob | date | Yes | NULL |
| Address | varchar(255) | No |  |
| City | varchar(100) | No |  |
| Country | varchar(100) | Yes | NULL |
| RegDate | timestamp | No | CURRENT\_TIMESTAMP |
| UpdationDate | timestamp | Yes | NULL |
| status | varchar(20) | No |  |
| img | blob | Yes | NULL |

## Table structure for table tblcars

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| VehiclesTitle | varchar(150) | No |  |
| VehiclesBrand | int(11) | No |  |
| VehiclesOverview | longtext | No |  |
| PricePerDay | int(11) | No |  |
| FuelType | varchar(100) | No |  |
| ModelYear | int(6) | Yes | NULL |
| SeatingCapacity | int(11) | No |  |
| Vimage1 | varchar(120) | No |  |
| Vimage2 | varchar(120) | Yes | NULL |
| Vimage3 | varchar(120) | Yes | NULL |
| Vimage4 | varchar(120) | Yes | NULL |
| Vimage5 | varchar(120) | Yes | NULL |
| AirConditioner | int(11) | No |  |
| PowerDoorLocks | int(11) | No |  |
| AntiLockBrakingSystem | int(11) | No |  |
| BrakeAssist | int(11) | No |  |
| PowerSteering | int(11) | No |  |
| DriverAirbag | int(11) | No |  |
| PassengerAirbag | int(11) | No |  |
| PowerWindows | int(11) | No |  |
| CDPlayer | int(11) | No |  |
| CentralLocking | int(11) | No |  |
| CrashSensor | int(11) | No |  |
| LeatherSeats | int(11) | No |  |
| RegDate | timestamp | Yes | CURRENT\_TIMESTAMP |
| UpdationDate | timestamp | Yes | NULL |
| color | varchar(15) | Yes | NULL |
| Driver\_id | int(20) | No |  |
| base\_rate | int(5) | Yes | NULL |

***CHAPTER***

***– 5 – IMPLEMENTATION AND TESTING***

|  |  |
| --- | --- |
| **Chapter - 5 Implementation and Testing** | **75** |
| **5.1 Language Used Characteristics** |  |
| **5.2 Class diagram (explanation each class in detail, What is the purpose of class? Then explain the working of methods of the class with the parameter it takes.)** |  |
| * 1. **Testing**      1. **Testing Objectives**   **5.3.2 Testing Methods and Strategies**  **5.3.3 Test Case** |  |
|  |  |

**8.1 Implementation :-**

Implementation means converting a new or revised system design into an operational one. Implementation is the stage of the project when theoretical design is turned into a working system. This means that the new design is implemented to establish a working system design. This stage should be handled carefully to achieve a new successfully running system giving the user confidence that the new system will work efficiently and as per the requirements.

For understanding any project some basic requirement should be fulfilled. For example, this project is build around the client server architecture. So the reader should be clear with the architecture.

Apart from the technology there are some specific terms used in the context of the project. So the meaning of all the terms should be clear, before diving into the intricacies of the project. Nevertheless, we are providing a brief overview of the technology that we have used and also we are providing the illustrated meaning of the terms that we have used, so that reader can have the basic understanding about our project at the first look.

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be

the systems analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

**5.1 Language Used Characteristics**

**5.2 Class diagram**

**5.3 Testing**

**5.3.1 Testing Objectives**

**5.3.2 Testing Methods and Strategies**

**5.3.3 Test Case**

**5.1 Technologies Used :-**

**5.1.1 Why PHP:-**

**5.1.2 Why HTML :-**

HTML, which stands for Hyper Text Markup Language, is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows images and objects to be embedded and can be used to create interactive forms. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. It can include or can load scripts in languages such as JavaScript which affect the behavior of HTML processors like Web browsers; and Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both HTML and CSS standards, encourages the use of CSS over explicit presentational markup.

Most graphical e-mail clients allow the use of a subset of HTML (often ill-defined) to provide formatting and semantic markup not available with plain text. This may include typographic information like coloured headings, emphasized and quoted text, inline images and diagrams. Many such clients include both a GUI editor for composing HTML e-mail messages and a rendering engine for displaying them. Use of HTML in e-mail is controversial because of compatibility issues, because it can help disguise phishing attacks, because it can confuse spam filters and because the message size is larger than plain text.

5.1.3 Why JAVASCRIPT :-

JavaScript is an object-oriented scripting language used to enable programmatic access to objects within both the client application and other applications. It is primarily used in the form of client-side JavaScript, implemented as an integrated component of the web browser, allowing the development of enhanced user interfaces and dynamic websites. JavaScript is a dialect of the ECMAScript standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

5.1.4 Why MySQL:-

There are a large number of database management systems currently available, some commercial and some free. Some of them: Oracle, Microsoft Access, Mysql.

These database systems are powerful, feature-rich software, capable of organizing and searching millions of records at very high speeds.

5.1.4.1 Understanding Databases, Records, and Primary Keys:

Every Database is composed of one or more tables. These Tables, which structure data into rows and columns, impose organization on the data.

The records in a table are not arranged in any particular order.

To make it easy to identify a specific record, therefore, it becomes necessary

5.1.4.2 standing Relationships and Foreign Keys (RDBMS):

You already know that a single database can hold multiple tables.

In a Relational database management system (RDBMS), these tables can be linked to each other by one or more common fields, called foreign keys.

5.1.4.3 What is Database administrator (DBA) ?

Database administrator is the super user of database; he has unrestricted rights and privileges to access database, grant permission to other database users.

5.1.4.4 What is Database user (DBU) ?

Database user is the person who uses the database in a restricted privileges, provided by database administrator.

**9.1 Testing :-**

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing is concerned with uncovering of the system error. The testing phase is an important part of software development. It is the pauperized system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Software testing is carried out in three steps:

1. The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is the important and major part of the project. So errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So unit testing is conducted to individual modules.

2. The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. This is due to poor interfacing, which may results in data being lost across an interface. A module can have inadvertent, adverse effect on any other or on the global data structures, causing serious problems.

3. The final step involves validation and testing which determines which the software functions as the user expected. Here also some modifications were. In the completion of the project it is satisfied fully by the end user.

**9.2 Objective of Testing** :-

The objectives of testing are:

* Testing is done with intent of finding an error.
* A good test case is one that has high probability of finding an uncovered error.
* Testing is done in order to check the various control flow of system.

**9.3 Scope of Testing :-**

The scope of testing is the extensiveness of the test process. A narrow scope may be limited to determining whether or not the software specifications were correctly implemented.

**9.4 Testing Principles :-**

1. All tests should be traceable to end user requirements.

2. Tests should be planned long before testing begins.

3. Testing should begin on a small scale and progress towards testing in large.

4. Exhaustive testing is not possible.

5. To be most effective testing should be conducted by a independent third party.

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used.

The various form of testing include:-

White box testing.

Black box testing.

**9.4.1 White Box Testing:-**

Each and every module of the project has been tested sufficiently by applying various test cases. White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed. **White box testing** is when the tester has access to the internal data structures and algorithms including the code that implement these.

White box testing is test case design method that uses the control structure of procedural design to derive test cases. Using white box testing methods, the software engineer can derive cases that: Using White-Box testing methods, the software engineer can drive test cases that

* Guarantee that logical decisions on their true and false sides.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structure to assure their validity.
* Guarantee that all independent paths within a module level have been exercised at least once.

**9.4.2 Black Box Testing:-**

This testing focuses on the functional requirements of the project. Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category. treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, exploratory testing and specification-based testing.

It focuses on the functional requirements of the software. The project has been sufficiently gone through the tests discovering following:

1. Incorrect or missing functions

2. Interface errors

3. Errors in data structures or external database access.

4. Performance error.

5. Initialization and termination error.

**9.5 Test Results:-**

After applying alpha and beta test, we have observed that the system is working fine and all functional requirements are satisfied but interface need to be improved.

**9.5.1 Unit testing:-**

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors within the boundary of the modules. These tests were carried out during the programming stage itself. All units of ViennaSQL were successfully tested.

**9.5.2 Integration testing:-**

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

**9.5.3 System testing:-**

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by giving input values to the system and by comparing with expected output. Top-down testing implementing here.

## 9.5.4 Acceptance Testing:-

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

**9.5.5 Alpha & Beta testing:-**

It is virtually impossible for software developer to foresee how the customer will really use a program. When customer software is built for one customer of a series of acceptance test is conducted to enables the customer to validate all requirements.

A customer conducts the alpha test at developer side. Alpha test were performed at our development side. Error and usage problems were noted and code was updated to remove all of them.

The beta test is conducted at one or more customer sides by the end user of the software. The beta testing of our system is not performed fully, as the product is not yet fully deploy completely at the user’s site.

**9.6 Test Plan:-**

A test plan is general document for the entire project that defines the scope, approach to be taken and the schedule of testing as well as identifies the test item for the entire testing process.

The input for test process:-

1. Project Plan

2. Requirements Document

3. System Design Document

**A test plan should contain the following:-**

* Test unit specification
* Feature to be tested
* Approach for testing
* Test deliverable
* Schedule
* Personal allocation

**9.7 Test case design:-**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error.

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

The test case specification for system testing has to be submitted for review before system testing commences.

**9.7.1 Test Case ‘A’**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No.** | **Action** | **Expected Result** | **Success** | **Comments** |
| **1** | User tries to type symbols or alphabets in login account. | Not allowed to do so. Prompted by error. | Yes | Action successfully tested |
| **2** | User types incorrect password. | Invalid user id or password display. Prompted to re-type. | Yes | Action successfully tested |
| **3** | View profile option selected. | Account details field loaded dynamically to the same screen. | Yes | Action successfully tested |

**9.7.2 Test Case ‘B’**:

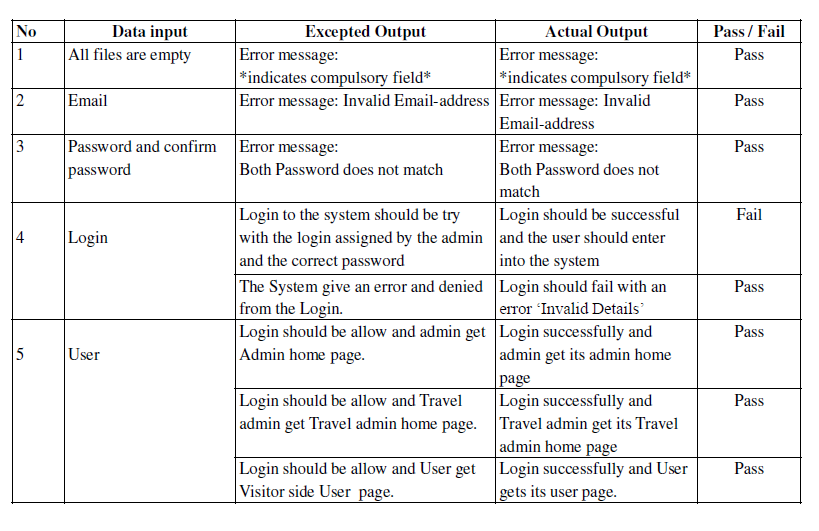
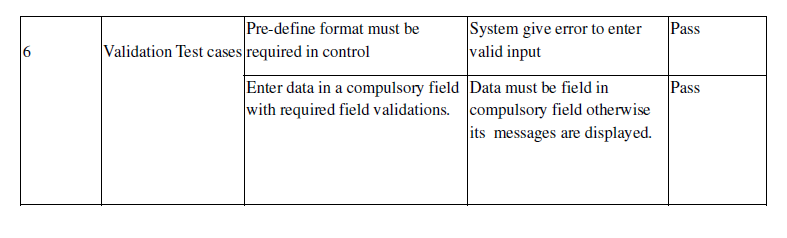
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No.** | **Action** | **Expected Result** | **Success** | **Comments** |
| **1** | User leaves user id field empty. | Prompted to write the correct input. | Yes | Action successfully tested |
| **2** | User leaves password field empty. | Prompted to write the correct input. | yes | Action successfully tested |
| **3** | User leaves mobile no field empty. | Prompted to write the correct input. | Yes | Action successfully tested |
| **4** | User types weak password. (length too small) | Prompted to write correct input. | Yes | Action successfully tested |
| **5** | Typed password doesn’t match to confirm password. | Prompted to write correct input. | Yes | Action successfully tested |
| **6** | User typed incorrect mobile no. | Prompted to write correct input. | Yes | Action successfully tested |
| **7** | User tries to type symbols are alphabet in mobile number field. | Not allowed to do so. | yes | Action successfully tested |

**9.7.3 Test Case ‘C’**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No.** | **Test Case title** | **Description** | **Expected outcome** | **Result** |
| **1** | Successful new account creation. | Registration with new userid, similar password and confirm password. | Registration should be successful and the user should enter into the login section. | passed |
| **2** | Unsuccessful registration due to dissimilar password and confirm password. | The password string is not similar in password and confirm password. | Registration should fail with an error “password doesn’t match”. | Passed |
| **3** | Unsuccessful registration due to already existing userid. | Check with this id, is any user already exist. | Registration should fail with an error ‘user already exist’. | Passed |
| **4** | Unsuccessful registration due to incomplete or other incorrect details. | When all the information are not provided correctly by the user. | Registration should fail with an error ‘fill all the entries’. | Passed |

**9.7.4 Test Case ‘D’**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No** | **Test Case title** | **Description** | **Expected outcome** | **Result** |
| **1** | Update the information. | Programmer should be able to update the record. | Successfully updated the record. | Passed |
| **2** | User login. | Verify the password given by the user. Shows main screen. | Verify the password and display main screen. | Passed |
| **3** | User logout | User should logout from the program. | Logout from the program. | Passed |

**** ****

**9.7.5 Test Case Result :-**

After the testing of modules, no errors were found & our project is well tested.

***CHAPTER***

***– 6 – CONCLUSION***

|  |  |
| --- | --- |
| * 1. **Chapter – Conclusion and Discussion**   **It should include your learning and achievements from the project, limitations of the project, future expansions.** | **97** |
|  |  |

6. **Conclusion and Discussion**

**“Technology feeds on itself. Technology makes more technology possible. And thus, it is very rightly called by Alvin Toffler, ‘That’s great, growling engine of change ---Technology’”.**

**11.1 Conclusion**:-

Nowadays, customers can reserve cars online, rent car online, and have the car brought to their door step once the customer is a registered member or go to the office to pick the car. The web based car rental system has offered an advantage to both customers as well as Car Rental Company to efficiently and effectively manage the business and satisfies customers’ need at the click of a button.

7. EXPECTED OUTCOME

This online Car Renting Service will be very easy to use that will saves a lot of time, money and labour. This website will be eco friendly, the monitoring of the car activity and the overall business becomes easy and includes the least of paper work. It increases the efficiency of the management at offering quality services to the customers. It provides custom features development and support with the software. In this website the customer can also give a feedback of the drivers or the car rental service such as the driver is able to drive a car or not, the behavior of driver is good or bad and they satisfied or not from using this car rental service. It increases the efficiency of the management at offering quality services to the customers.

It provides custom features development and support with the software.

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

* + Automation of the entire system improves the efficiency
  + It provides a friendly graphical user interface which proves to be better when compared to the existing system.
  + It gives appropriate access to the authorized users depending on their permissions.
  + It effectively overcomes the delay in communications.
  + Updating of information becomes so easier.
  + System security, data security and reliability are the striking features.
  + The System has adequate scope for modification in future if it is necessary.
  + This Web application is restricted to only limited type of users.
  + Here the details of customers, drivers and cars are maintained.

**11.1.1 Limitation of project:-**

* Our project have limited scope, it will provide services only for Sub-Domain.
* In this project we cannot give current status of service provider.

**11.1.2 Difficulties encountered:-**

We had to design such a site which was removing the demerits of the existing system. We have to keep in mind and make the system as user friendly as possible for the convenience of the people working in this environment.

* Due to less Team size we face many problem like survey, documentation etc.

**11.1.3 Suggestions for future enhancements:-**

**In future our application overcome the flaws if occurred and attain new features offered for the flexible and easy use.**

* In future we will enhance user interface.
* Mobile application can also be created for this System.
* We can attach this system to Social Networking.
* In future, we can provide the website space for advertisements purpose.
* In future, we may try to expand the system and provide service for sending E-mails to other servers such as Gmail, Yahoo, Rediffmail etc.
* In future, we can provide the website space for organisational purpose and also educational organisations.
* Also try to make it easier.
* An option of marking the mails as favourite and forwarding of a mail can be added in future.
* Preview of the attachment facility can be added as future enhancement.

**6. APPLICATIONS**

There are some areas in which our application work effectively are as follows :-

1. The main scope of Car Renting Service is used for travelers(tourists) and visitors having small groups or individuals.

2. Car Rental Service website can be used in transport services trucks, loading etc.

3. This can be used as a cab service and public transport etc.

4. It can be used in marriage, family functions and meetings etc as a pickup service.

The user manual is drafted In such a way that every common man can understand that how to use this application.

**12.1.1 Books Referred:-**

Beginning PHP, Wrox Publications.

Web Technologies by Godbole, Tata Mc Graw .

Web Technology – A Developer’s Perspective – PHI by N. P. Goplan

BLACK BOOK HTML ---WILEY DREAMTECH

HTML, complete reference, B.P.B. Publication.

Design Web Pages, BPB Publication.

BEGINNING PHP 5---DAVE MERCER

1. System, Online. 'Online Car Rental System'. Academia.edu. N.p., 2015. Web. 9 June 2015.

2. Scribd.com, Online. '49930505 Car Rental System Project Report'. N.p., 2015. Web. 9 June 2015.

3. Scribd.com, Online. 'Car Rental System Documentation'. N.p., 2015. Web. 9 June 2015.

4. Freelancer, Online. 'Project Documentation Car Rental Company Software Development Freelancers and Jobs - Freelancer'. N.p., 2015. Web. 9 June 2015.

5. Slideshare.net, Online. 'Zook Car Rental System Project'. N.p., 2015. Web. 9 June 2015.

6. Kaewman, Sasitorn. 'Online Decision Support System of Used Car Selection using K-Nearest Neighbor Technique'. IJFCC (2012): 164-166. Web.

7. Wikipedia, Online. 'Use Case Diagram'. N.p., 2015. Web. 9 June 2015.

8. Wikipedia, Online. 'Activity Diagram'. N.p., 2015. Web. 9 June 2015.

9. Tutorialspoint.com, Online. 'UML - Activity Diagrams'. N.p., 2015. Web. 9 June 2015.

10. Wikipedia, online. 'Swim Lane'. N.p., 2015. Web. 9 June 2015.

11. Mindtools.com, Online. 'Swim Lane/Rummler-Brache Diagrams: Mapping and Improving Processes in Your Organization'. N.p., 2015. Web. 9 June 2015.

12. Laudon, Kenneth C, and Jane Price Laudon. Management Information Systems. Upper Saddle River, NJ: Prentice Hall, 2000. Print.

13. Menkus, Belden. 'Car Rental Chain Former Owners Charged With Computer Frauds'. Computer Fraud & Security Bulletin 1993.3 (1993): 3-4. Web.

14. Li, Zhang. 'Design And Realization Of Car Rental Management System Based On AJAX+SSH'. Information Technology J. 12.14 (2013): 2756-2761. Web.

Booklogic’s Website, 2015, Booklogic’s Dynamic Car Booking System Retrieved from

http://www.booklogic.net/products/car-booking-system

Tousif Khan, Source Forge Website, November 27, 2012, Vehicle Management System Project

Retrieved from http://sourceforge.net/projects/vehiclebooking/

Portcalls Website, September 22, 2014, Vehicle Booking System: What’s Cooking? Retrieved from http://www.portcalls.com/vehicle-booking-system-whats-cooking/#

Iaweb Website, July 6, 2010, Central Reservation System Retrieved from http://www.iaweb.net/global-distribution-systems/central-reservation-system.html

Diamond Rent-a-Car Website, 2015, About Us Retrieved from

http://www.diamond-rentacar.com/about.php

Mei Lei, Netsuite Website, January 16, 2008, Diamond IGB Goes Live with Netsuite,

Simplifying Processes for Car Rental Company Retrieved from http://www.netsuite.com/portal/press/releases/nlpr01-16-08.shtml

System Development Life Cycle Methodology. Retrieved from https://www.wirc-icai.org/material/2-System-Development-Life-Cycle- Methodology.pdf

[1] R. Mcleod and Jr. G. P. Schell, Management Information System, Tenth Edition. India: Pearson Education, Inc, 2007.

[2] Y. Damayanti, “Perancangan Sistem Informasi Penyewaan Mobil Rama Rental Car Dengan Menggunakan Microsoft Visual Basic Versi 6.0,” unpublished. Undergraduate Thesis. Jakarta: Gunadarma University, 2005.

[3] M. R.T. Cahyanto et al., “Sistem Informasi Manajemen Rental Mobil CV. Opal Transport dengan Java SE (Standart Edition) dan MySql Berbasis OOP (Object Oriented Programming),” unpublished. Undergraduate Thesis. Yogyakarta: STMIK AKAKOM, 2010.

[4] Yodiyanto, “Analisis dan Perancangan Sistem Informasi Rental Mobil dan Angkutan Travel Berbasis Web pada PT. Kembang 88,” unpublised. Undergraduate Thesis. Jakarta: Bina Nusantara University, 2006.

[5] R. D. Sari, “Building Application System Car Rental Reservation and Payment Online Web-Based (Case Study in The Rental Daras Corporation),” unpublished. Undergraduate Thesis. Bandung: Unikom, 2011.

[6] Jogiyanto, Analisis & Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. Yogyakarta: Andi Yogyakarta, 2005.

[7] Avis Indonesia, “Jurnal Avis Indonesia,” Jakarta: Avis Indonesia, 2009.

[8] Jogiyanto, Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis, Edisi 2. Yogyakarta: Andi Yogyakarta, 2001.

[9] K. Andri, Rekayasa Perangkat Lunak. Yogyakarta: Gava Media, 2004.

[10] K. Kendall, Analisis dan Perancangan Sistem. Jakarta: Penerbit Indeks, 2006.

[11] T. Connolly and C. Begg, Database System: A practical Approach to Design, Implementation, and Management, Third Edition. England: Addison Wesley, 2002. Proceedings of The 1st International Conference on Information Systems For Business Competitiveness (ICISBC) 2011

[12] Fathansyah. Basis Data. Bandung: CV. Informatika, 2000.

[13] A. Ladjamudin, Analisis dan Desain Sistem Informasi. Yogyakarta: Graha Ilmu, 2005.

[14] L. Hakim, 150 Rahasia dan Trik Menguasai PHP. Jakarta: Elex Media Komputindo, 2004.

[15] B. Sidik, Pemrograman Web dengan PHP. Bandung: Informatika, 2001.

[16] B. Sidik, MYSQL untuk Pengguna Administrator dan Pengembangan Aplikasi Web. Bandung: Informatika, 2005.

[17] W. Siswoutomo, PHP Enterprise. Jakarta: Elek Media Komputindo, 2008.

[18] Sudirman, Membangun Aplikasi Web dengan Php dan Mysql. Yogyakarta: Graha Ilmu, 2003.

[19] D. Suprianto, Buku Pintar Pemrograman Php. Jakarta: Oase Media, 2008.

|  |  |
| --- | --- |
| **Appendix : Screen and Report Formats** |  |
|  |  |
| **Bibliography and References** |  |

**13.1 Glossary/Acronyms used :-**

**LIST OF ABBREVIATIONS**

* **Actor**: An actor is someone who is interacting with the system and is external the system. They help in defining the system and makes clear what the system should do. An actor may be a user, any hardware device or any other system.
* **Association**: An association is a kind of relationship among classes that tells one class knows about the other. Instance of one class is made in the other.
* **Attributes**: Attributes of classes are the property of classes. They define the state of the system. Attributes of classes can be found out from the system domain.
* **Availability**: Availability indicates the time interval up to which the system Is available without having to change the system.
* **Class**: A class is collection of objects with common structure, common behaviours, common relationship and common semantics. Classes can be found out by examining the objects in the sequence and the collaboration diagram.
* **Class diagram**: A class diagram models the user requirements in the form of classes and relationships among them. A class diagram shows the existing classes and relationship among them.
* **Constraints**: Conditions under which the system perform accurately.
* **Database**: A collection of data that is treated as a unit composed of logical and physical structure. Databases are designed to store and retrieve related information.
* **Extendibility**: The extent to which architectural, data or procedural design can be extended.
* **Forward engineering**: Forward engineering refers to the process n which skeleton code is generated from the class diagram.
* **GUI**: Graphic user interface.
* **Relational database**: A relational database follows the relational data model. It stores data in the form of relational and relationship among relations.
* **Relationship**: Relationship shows the interaction and communication among the classes. They show how the classes are interacting with each other. Main types of relationship are association, dependency, realization, inheritance.
* **Sequence diagram**: Sequence diagram displays object inheritance with respect to time. Sequence diagram show how step by step objects interact and communicate with each other. These diagrams emphasize the sequence of events what are taking place. These diagrams are used to find the links between the object and thus their behavior.
* **Software**: Software consists of program that when provide desired results, data structures which can be manipulated and documentations that tells the operation and use of the program.
* **Software architecture**: Software architecture refers to the overall structure of the software and the ways in which that structure provides conceptual integrity for the system.
* **UML**: Unified Modeling Language while using the Object Oriented Approach the UML is the standard language for visualizing, specifying, constructing and documenting the artefacts of a software intensive system.
* **Use Case**: A Use Case is a pattern of behavior the system exhibits. Each Use Case shows a series of transaction performed by an actor and the system in a single dialogue there are created to visualize how the actor interact with the system respond to it.
* **User**: A user is someone who is interacting with the system and is external to the system. They help in defining the system and make it clear what the system should do.