HOTEL ROOM BOOKING SYSTEM

Project Report Submitted By

RIYA ABRAHAM

Reg. No: AJC17MCA-I047

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

(INTMCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**Hotel Room Booking System"** is the bonafide work done by **RIYA ABRAHAM** (**Reg.No:AJC17MCA-I047**) in partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Ms. Navyamol K T
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **"HOTEL ROOM BOOKING SYSTEM"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date: **RIYA ABRAHAM**

KANJIRAPPALLY **Reg.No:AJC17MCA-I047**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms**. **Meera Rose Mathew** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Navyamol K T** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

RIYA ABRAHAM

ABSTRACT

Online Hotel Room Booking System is a ecommerce platform which meant to help the customers to make their online booking easy. The customer can also reduce the time and effort in searching hotels by using this system. It mainly focuses on helping the users to book the hotels. It provides User to easily search room's availability, category & easy updating of the room's records. The system allows the manager to post available rooms in the system. Customers can view and book room online. Hotels has the power of either approving or disapproving the customer's booking request. The system is hence useful for both customers, admins and hotels to portable manage the hotel activities.

Hotel Registration

In this module hotel can register by giving username, password and other hotel details. Admin, user and hotels can login to the website.

Approval

Admin can approve the hotels registration by checking hotel details. Once approved hotel can login to the system and complete their profile.

Filtering

Filtering the hotels based on the hotel price and location by the customer.

Searching

Searching the hotels based on the places, vacant dates and room availability.

Room Booking

Booking rooms based on the selected hotel. Report based on the booking hotels to the corresponding hotels. Hotels can view the vacant room details

Cancellation

Cancel the booked rooms.

Payments

Pay the amount of room price to hotel.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
2.5	ADVANTAGES OF PROPOSED SYSTEM	7
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	10
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	PHP	11
3.3.2	MYSQL	12
4	SYSTEM DESIGN	14
4.1	INTRODUCTION	15
4.2	UML DIAGRAM	15
4.2.1	USE CASE DIAGRAM	16
4.2.2	SEQUENCE DIAGRAM	18
4.2.3	CLASS DIAGRAM	21
4.2.4	STATE DIAGRAM	22
4.2.5	ACTIVITY DIAGRAM	24

4.2.6	**COLLABORATION DIAGRAM**	**26**
4.2.7	**COMPONENT DIAGRAM**	**28**
4.2.8	**OBJECT DIAGRAM**	**30**
4.2.9	**DEPLOYMENT DIAGRAM**	**32**
4.3	**USER INTERFACE DESIGN USING FIGMA**	**34**
4.4	**DATA BASE DESIGN**	**36**
5	**SYSTEM TESTING**	**43**
5.1	**INTRODUCTION**	**44**
5.2	**TEST PLAN**	**45**
5.2.1	**UNIT TESTING**	**45**
5.2.2	**INTEGRATION TESTING**	**46**
5.2.3	**VALIDATION TESTING**	**46**
5.2.4	**USER ACCEPTANCE TESTING**	**47**
5.2.5	**SELENIUM TESTING**	**47**
5.2.6	**TEST CASE**	**48**
6	**IMPLEMENTATION**	**53**
6.1	**INTRODUCTION**	**54**
6.2	**IMPLEMENTATION PROCEDURE**	**54**
6.2.1	**USER TRAINING**	**55**
6.2.2	**TRAINING ON APPLICATION SOFTWARE**	**55**
6.2.3	**SYSTEM MAINTENANCE**	**55**
7	**CONCLUSION & FUTURE SCOPE**	**57**
7.1	**CONCLUSION**	**58**
7.2	**FUTURE SCOPE**	**58**
8	**BIBLIOGRAPHY**	**59**
9	**APPENDIX**	**61**
9.1	**SAMPLE CODE**	**62**
9.2	**SCREEN SHOTS**	**66**

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

"HOTEL ROOM BOOKING SYSTEM" is a web application which is meant to help the customers to book the rooms using this application. The customer can also reduce the time and effort in searching the hotel rooms in local area. The proposed system includes three users they are administrator, customer and hotel. Registered customers can login to the site and can book the rooms and can also make the payment to the hotel owner through online. The admin can also access the daily progress of the customer report through "daily progress report" option. The administrator has the central control over the whole system. When the hotel creates a room and room types in the application, and the hotel can change the room's amount and images based on construction of the rooms. Hotel can do the view the report based on the daily and weekly assessment of room booking and give a result in chart. The customer can give feedback about the rooms and services.

1.2 PROJECT SPECIFICATION

The proposed system is made to help the customers for an easy and convenient way of booking the rooms. We will also provide users to give feedbacks, they can view the room details, payment details, complaint details etc.

The system includes 3 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can view all registered customer and hotels. Admin can able to approve or reject users and also reject the registered hotels.

2. Customer Module

Customer can register and they can login to the application. Customer can book their rooms and do secure online payment. Customer can view the whole rooms in this application and they can give feedback to the admin.

3. Hotel Module

Hotels can add room type and view the customer booking details. View the rooms details and activate & deactivate the rooms during maintains work.

Functional modules are :

Hotel Registration

In this module hotel can register by giving username, password and other hotel details. Admin, user and hotels can login to the website.

Approval

Admin can approve the hotels registration by checking hotel details. Once approved hotel can login to the system and complete their profile.

Filtering

Filtering the hotels based on the hotel price and location by the customer.

Searching

Searching the hotels based on the places, vacant dates and room availability.

Room Booking

Booking rooms based on the selected hotel. Report based on the booking hotels to the corresponding hotels. Hotels can view the vacant room details

Cancellation

Cancel the booked rooms.

Payments

Pay the amount of room price to hotel.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Customer can register and they can book their hotels through the project. Customer can done payment through the system. The proposed system rectify the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- If there is a power failure while making a booking.
- The software needs to be renewed by each year.
- If there is a virus attack the stored information might get corrupt.
- If there is a power cut while there is a checkin or checkout, the customer need to wait for server start up.
- Customers do not get a chance to identify the quality of rooms and do not expect the room clean used

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed. In our proposed system there is admin who can view all the customer details. It allows customers to book the rooms and do their transactions by using online payment method .Users of this proposed system are admin and customer. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Our website is multifunctional which includes customer introduction and project details etc.

The master file that is file which contains all the details of the data's which are kept for long time is customer master. The customer master contains personal details of the customer like customer id, customer name, residential address, etc. This system is designed to help the customers to make the booking easy. The proposed system provides

consistency of data and reduces the paper work. Also, the customer can easily select the rooms based on there on option. This helps the customer to fulfill his dream rooms through online. This system helps the customer to get its booking smoothly, efficiently and in less time. This system is made to help the customer.

1.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- **Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

- **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

- **Better service: -**

The system will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, HOTEL ROOM BOOKING SYSTEM was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project.

The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of

Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

HOTEL ROOM BOOKING SYSTEM, GUI is simple so that users can easily use it. HOTEL ROOM BOOKING SYSTEM is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 4 GB

Hard disk - 1 TB

3.2.2 Software Specification

Front End - HTML, CSS

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page, it now stands for

PHP: Hypertext Pre-processor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on

most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

  The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL92" refers to the standard released in 1992,

"SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

  - **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available

  - **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram

is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Use case diagram
- Sequence diagram
- Class diagram
- State diagram
- Activity diagram
- Collaboration diagram
- Component diagram
- Object diagram
- Deployment diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

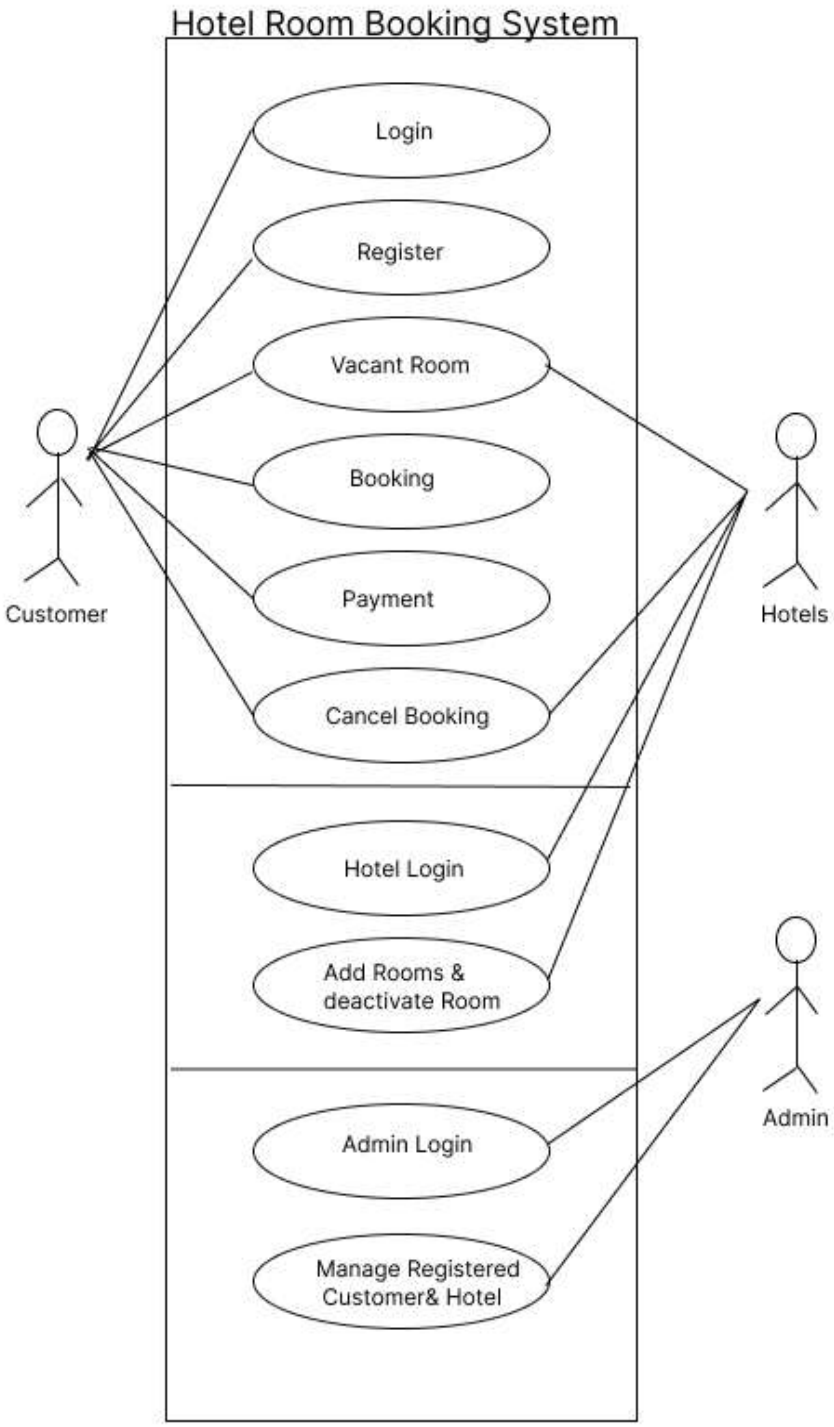
System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points

Fig 1 : Use case diagram for Hotel Room Booking System



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by

businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

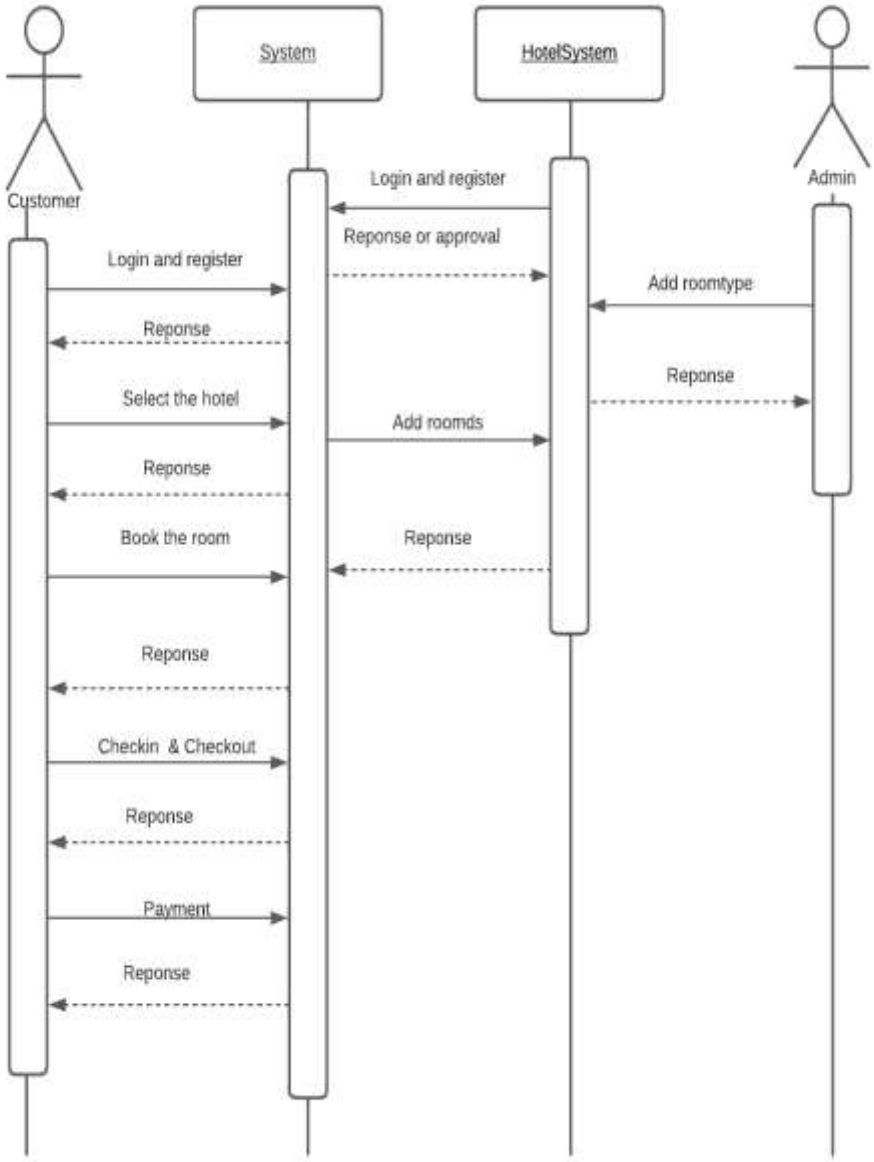
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

iv. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

1. Used to model and visualize the logic behind a sophisticated function, operation or procedure.
2. They are also used to show details of UML use case diagrams.
3. Used to understand the detailed functionality of current or future systems.
4. Visualise how messages and tasks move between objects or components in a system.

Fig 1 : Sequence diagram for Hotel Room Booking System



4.2.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Purpose of Class Diagrams

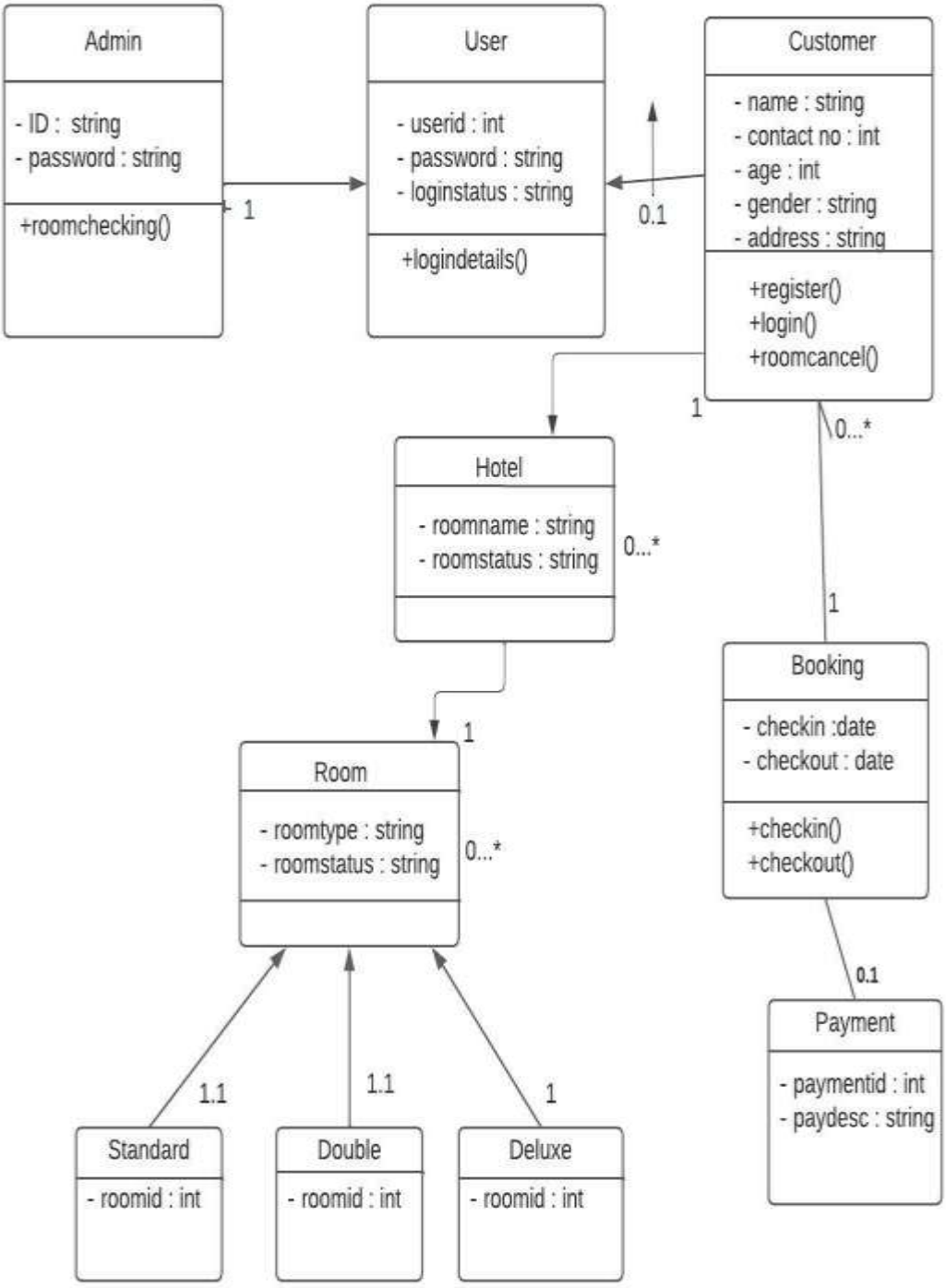
The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

ML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Fig 1 : Class diagram for Hotel Room Booking System



4.2.4 STATE DIAGRAM

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

A classic form of state diagram for a finite automaton (FA) is a directed graph with the following elements (Q, Σ, Z, δ, q0, F):

Vertices Q: a finite set of states, normally represented by circles and labeled with unique designator symbols or words written inside them

Input symbols Σ: a finite collection of input symbols or designators

Output symbols Z: a finite collection of output symbols or designators

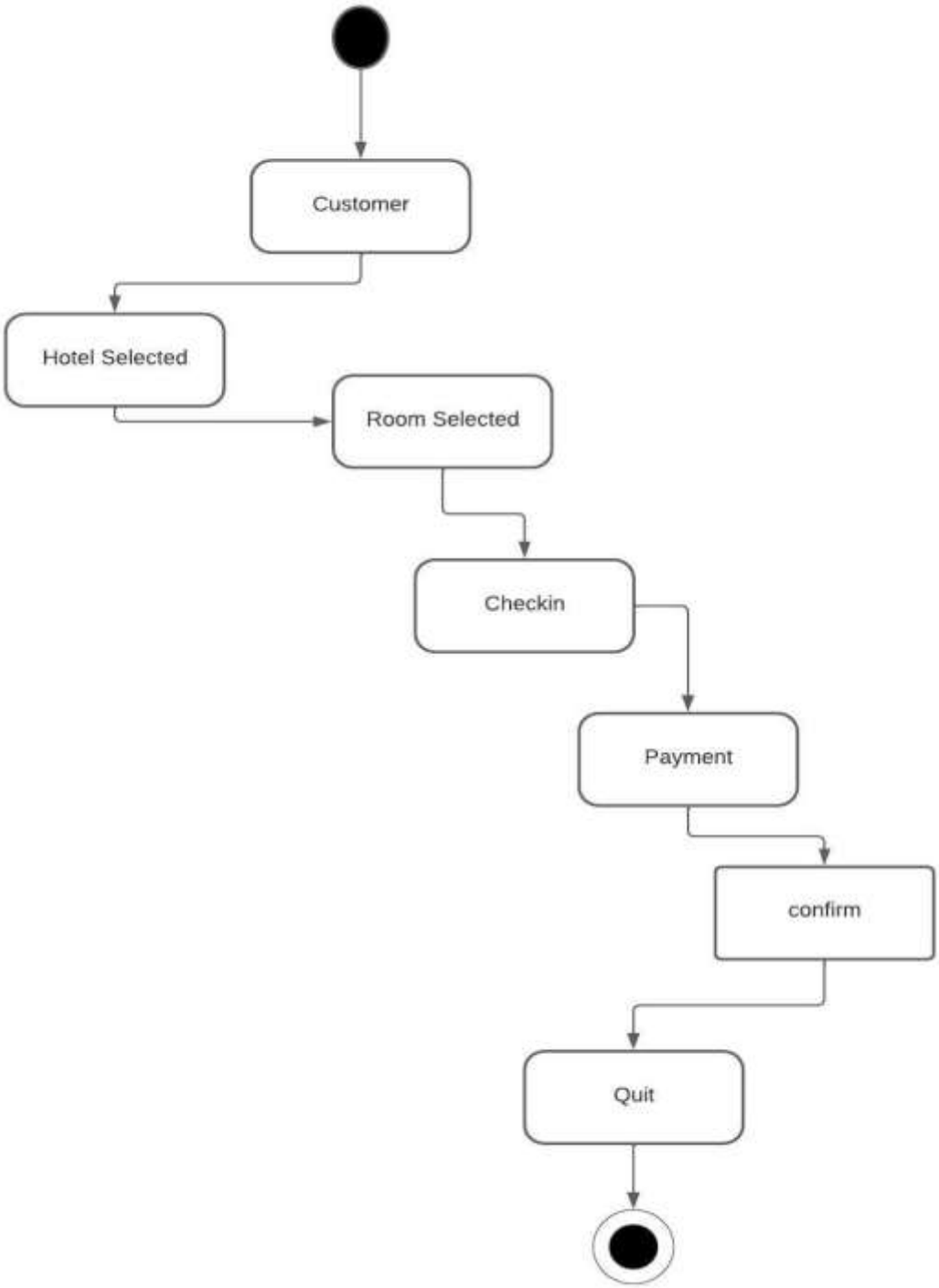
The output function ω represents the mapping of ordered pairs of input symbols and states onto output symbols, denoted mathematically as $\omega : \Sigma \times Q \rightarrow Z$.

Edges δ: represent transitions from one state to another as caused by the input (identified by their symbols drawn on the edges). An edge is usually drawn as an arrow directed from the present state to the next state. This mapping describes the state transition that is to occur on input of a particular symbol. This is written mathematically as $\delta : Q \times \Sigma \rightarrow Q$, so δ (the transition function) in the definition of the FA is given by both the pair of vertices connected by an edge and the symbol on an edge in a diagram representing this FA. Item $\delta(q, a) = p$ in the definition of the FA means that from the state named q under input symbol a, the transition to the state p occurs in this machine. In the diagram representing this FA, this is represented by an edge labeled by a pointing from the vertex labeled by q to the vertex labeled by p.

Start state q0: (not shown in the examples below). The start state $q0 \in Q$ is usually represented by an arrow with no origin pointing to the state. In older texts, the start state is not shown and must be inferred from the text.

Accepting state(s) F: If used, for example for accepting automata, $F \in Q$ is the accepting state. It is usually drawn as a double circle. Sometimes the accept state(s) function as "Final" (halt, trapped) states.

Fig 1 : State diagram for Hotel Room Booking System



4.2.5 ACTIVITY DIAGRAM

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity

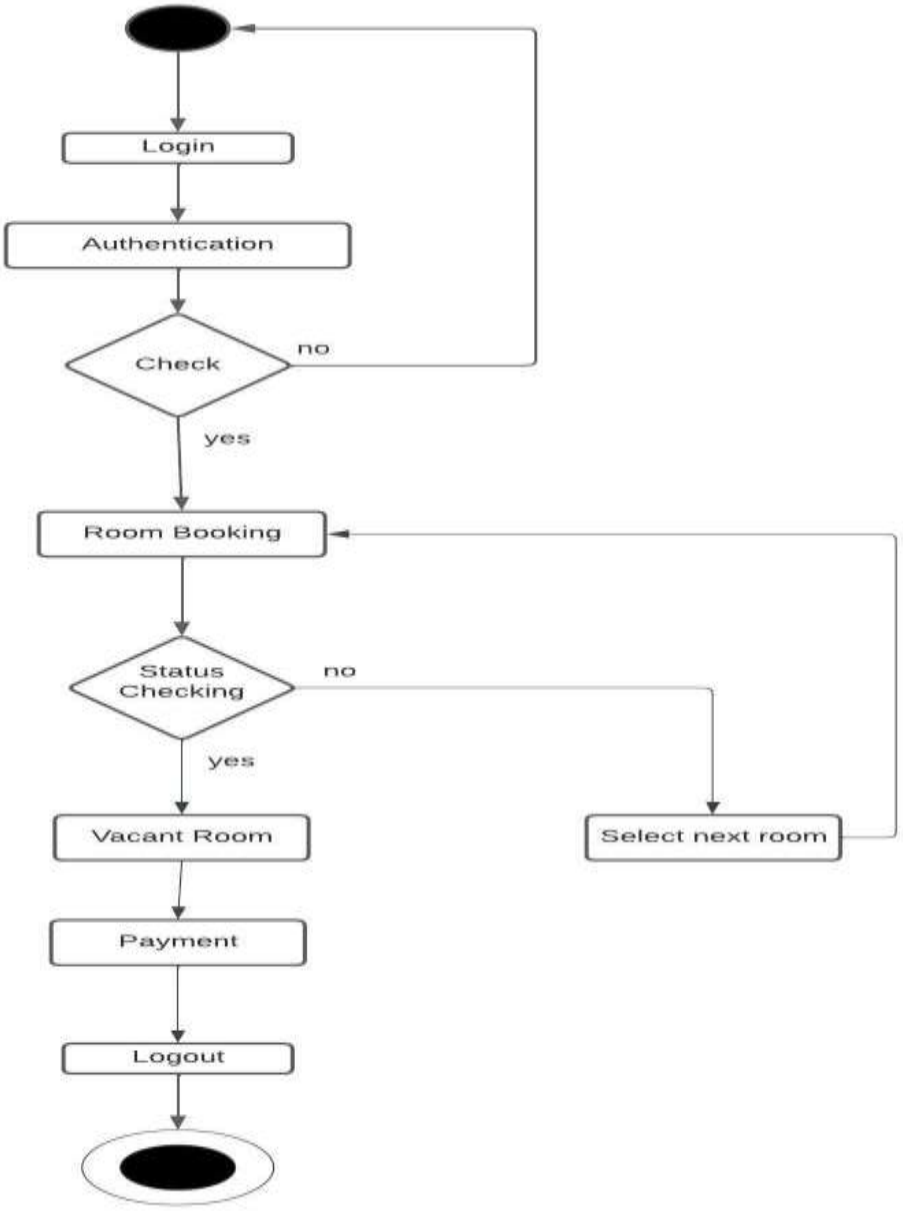
diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.

An activity diagram is very similar to a flowchart.

Purpose of activity diagram

An activity diagram shows business and software processes as a progression of actions. These actions can be carried out by people, software components or computers. Activity diagrams are used to describe business processes and use cases as well as to document the implementation of system processes.

Fig 1 : Activity diagram for Hotel Room Booking System



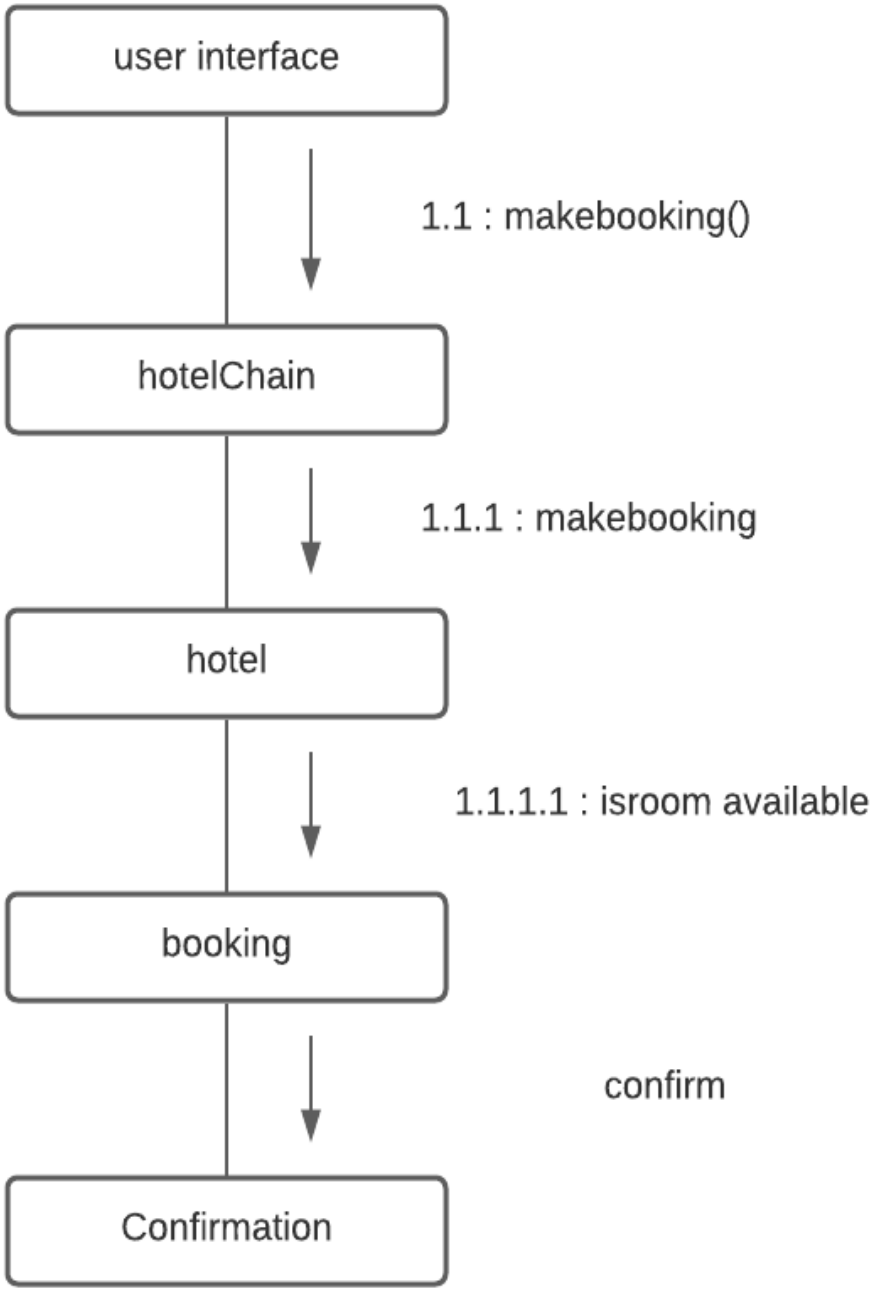
4.2.6 COLLABORATION DIAGRAM

Collaboration diagrams (known as Communication Diagram in UML 2.x) are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

Because of the format of the collaboration diagram, they tend to better suited for analysis activities (see Activity: Use-Case Analysis). Specifically, they tend to be better suited to depicting simpler interactions of smaller numbers of objects. However, if the number of objects and messages grows, the diagram becomes increasingly hard to read. In addition, it is difficult to show additional descriptive information such as timing, decision points, or other unstructured information that can be easily added to the notes in a sequence diagram. So, here are some use cases that we want to create a collaboration diagram for:

- Model collaborations between objects or roles that deliver the functionalities of use cases and operations
- Model mechanisms within the architectural design of the system
- Capture interactions that show the messages passing between objects and roles within the collaboration
- Model alternative scenarios within use cases or operations that involve the collaboration of different objects and interactions
- Support the identification of objects (hence classes) that participate in use cases.
- Each message in a collaboration diagram has a sequence number.
- The top-level message is numbered 1. Messages sent during the same call have the same decimal prefix but suffixes of 1, 2, etc. according to when they occur.

Fig 1 : Collaboration diagram for Hotel Room Booking System



4.2.7 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose of Component Diagrams

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

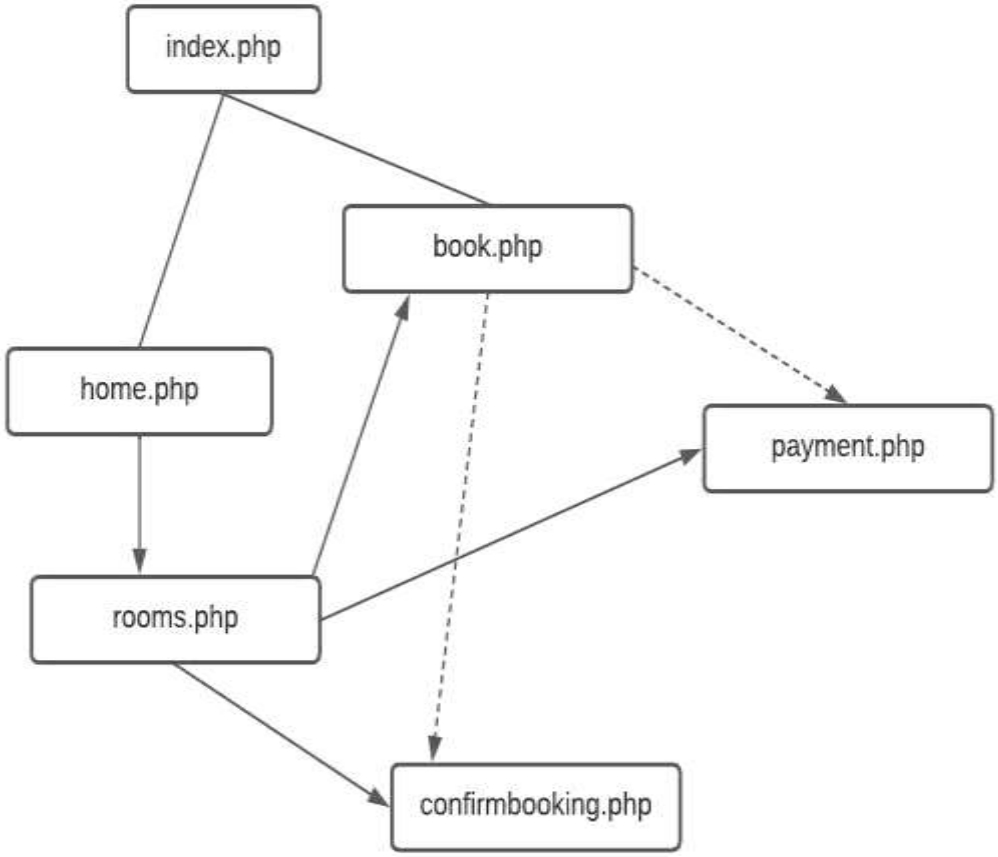
Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as –

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.

Fig 1 : Component diagram for Hotel Room Booking System



4.2.8 OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

Purpose of Object Diagrams

The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams.

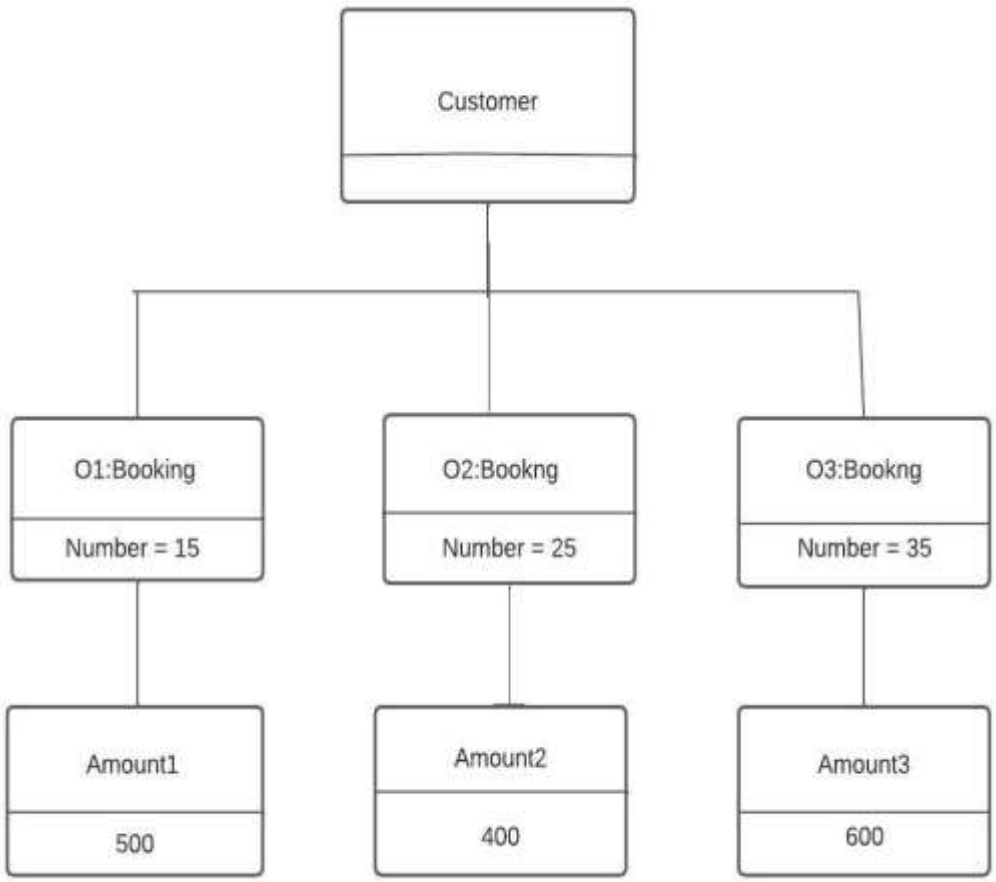
The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature.

It means the object diagram is closer to the actual system behavior. The purpose is to capture the static view of a system at a particular moment.

The purpose of the object diagram can be summarized as −

- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behaviour and their relationship from practical perspective

Fig 1 : Object diagram for Hotel Room Booking System



4.2.9 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

Purpose of Deployment Diagrams

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

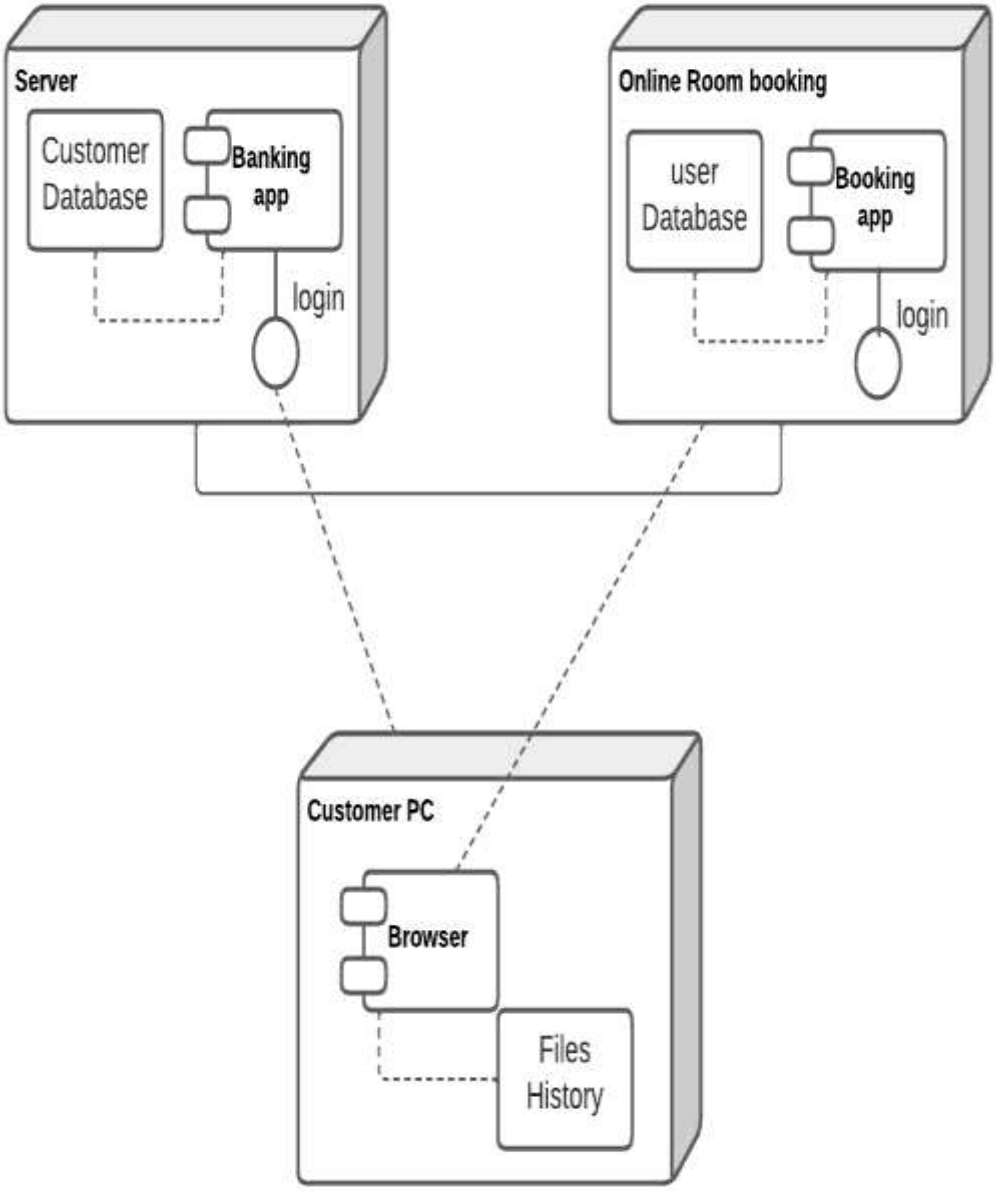
UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as −

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

Fig 1 : Deployment diagram for Hotel Room Booking System

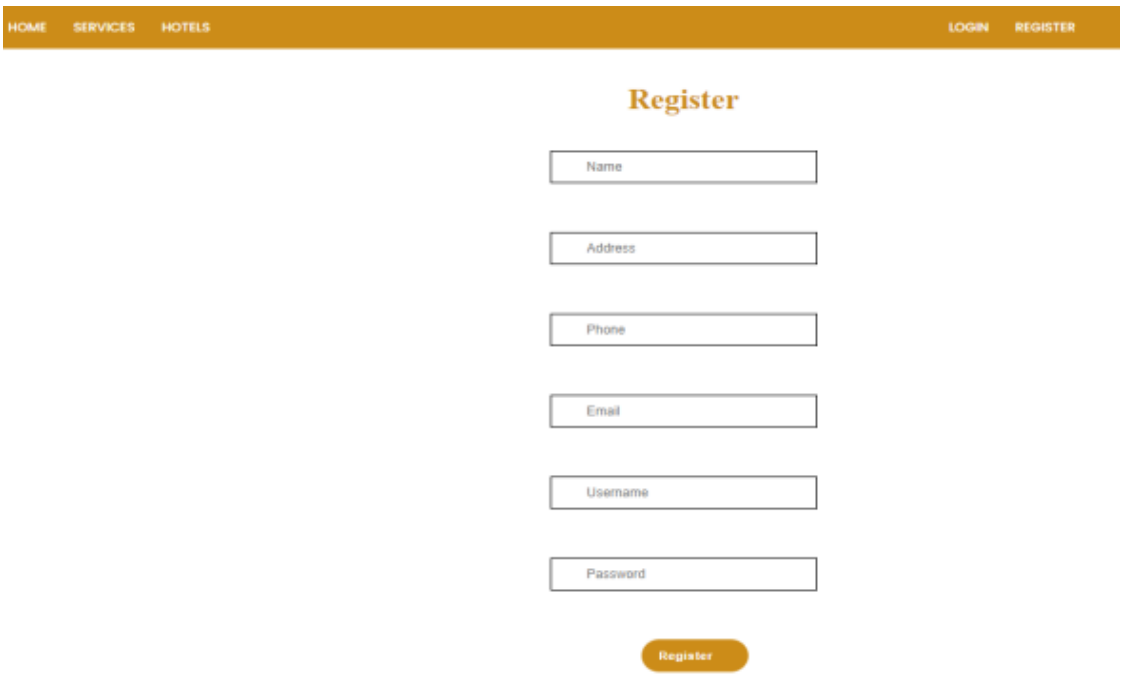

Server
Customer Database
Banking app
login
Online Room booking
user Database
Booking app
login
Customer PC
Browser
Files History

4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1-INPUT DESIGN

Form Name : User Login

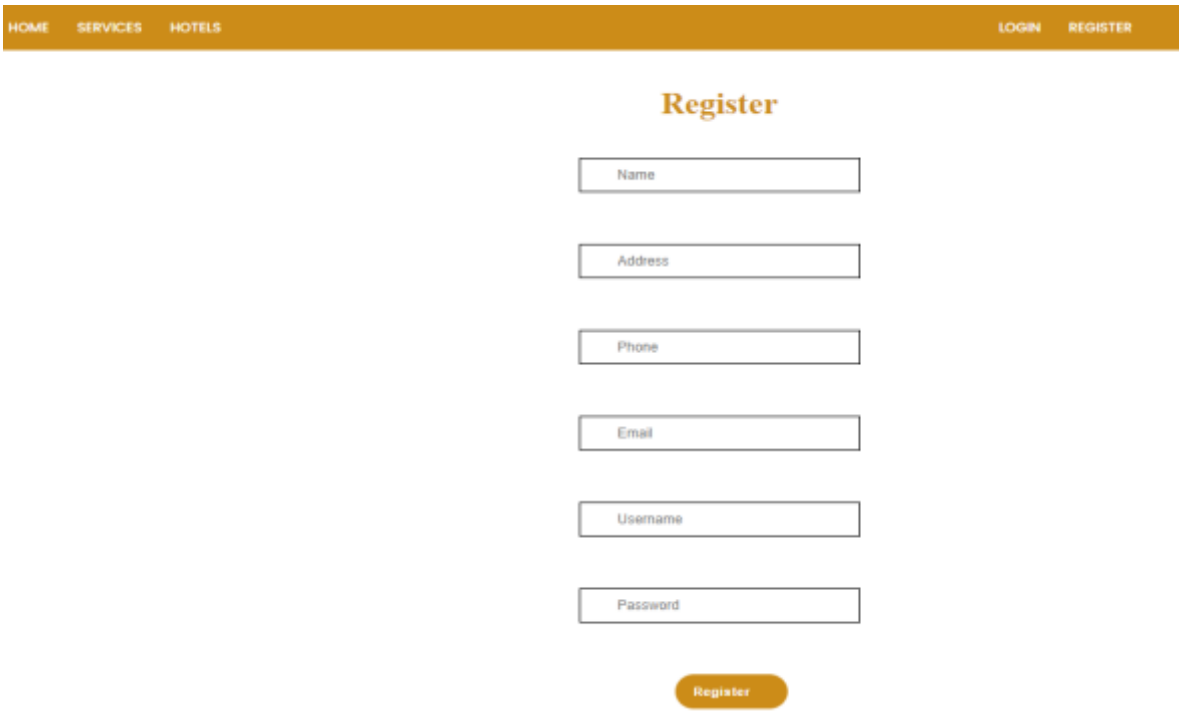
Form Name : User Register



4.3.2 OUTPUT DESIGN

User Login

User Register



4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN

Table No : 01

Table Name : tbl_customerregister

Primary Key : regid

Foreign Key : nil

Table Description : To store user Login information

Fieldname	Data Type	Size	Description
regid	Int	10	Primary key of login table
name	Varchar	20	name to identify
Address	Varchar	20	Address of user
Phone	Int	20	Phone no of users
Email	Varchar	20	Email of the user
Username	Varchar	30	Username to login
Password	Varchar	20	Password to login
Status	Varchar	20	Identify user and admin

Table No : 02

Table Name : tbl_hotelregister

Primary Key : hid

Table Description: To store hotel registration details

Fieldname	Data Type	Size	Description
hid	Int	10	Primary key of login table
hotelname	Varchar	20	name to identify
phones	Int	20	Phone no of hotels
emails	Varchar	20	Email of the hotel
username	Varchar	30	Username to login
password	Varchar	20	Password to login

Table No : 03

Table Name : tbl_hotels

Primary Key : hotel_id

Foreign Key : hid

Table Description: Details of hotels

Fieldname	Data type	Size	Description
hotel_id	Int	10	Primary key
hid	Int	20	Id of table register2
place	Varchar	50	hotel located place
description	Varchar	20	Hotel description
picture	Varchar	30	Hotel image

Table No : 04

Table Name : tbl_roomtype

Primary Key : type_id

Table Description: Name of roomtype

Fieldname	Data type	Size	Description
type_id	Int	10	Primary key
roomtype	varchar	20	room type names

Table No : 05

Table Name : tbl_rooms

Primary Key : room_id

Foreign Key : hid

Table Description: Details of rooms

Fieldname	Data type	Size	Description
room_id	Int	10	Primary key
hid	varchar	20	Id of table register2
type_id	Int	50	Rooms name
price	varchar	20	Room price
picture	varchar	30	Rooms image
roomqty	int	30	Quatity of room
status	int	30	booked or checkout

Table No : 06

Table Name : tbl_book

Primary Key : bid

Foreign Key : regid,hid,room_id

Table Description: To book the hotel room details

Fieldname	Data type	Size	Description
bid	Int	10	Primary key
regid	Varchar	20	Id of table register2
hid	Int	50	Id from table hotels
room_id	Varchar	20	Id from table rooms
checkin	Int	30	Check in date
checkout	Varchar	30	Check out date
no_of_room	Int	30	No of rooms booked

Table No : 07

Table Name : tbl_payment

Primary Key : pid

Foreign Key : regid

Table Description: Payment Details

Fieldname	Data type	Size	Description
pid	Int	10	Primary key
regid	Int	20	Register id of customer
name	Varchar	30	Registered name
amt	Int	30	Payed amount
pay_id	Varchar	20	Pay id from gateway
date_added	Varchar	30	Payment done date

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are: Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan. The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing ✞ Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when

an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Selenium Testing

Selenium is one of the most widely used open-source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages. Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android. Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari. Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven, Jenkins, & Docker to achieve continuous testing. It can also be integrated with tools such as TestNG, & JUnit for managing test cases and generating reports.

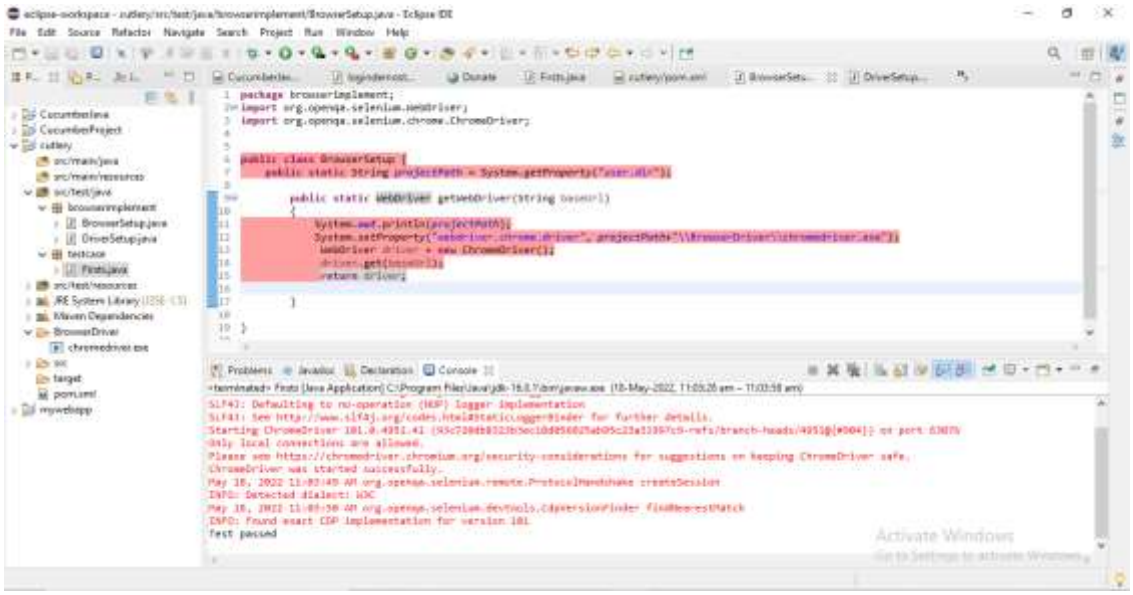
5.2.6 Test Case

Test Case 1

Project Name: Hotel Room Booking System					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By: Riya Abraham**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 19-05-2022		
Module Name: Login Screen			**Test Executed By : Ms Navyamol K.T**		
Test Title : Verify login with validusername and password			**Test Execution Date:** 19-05-2022		
Description: Test the Login Page					
Pre-Condition : User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to Login Page		Login Page should be displaye d	Login page displayed	Pass
2	Provide Valid Userna me	User Name: user	User should be able to Login	User Logged in and navigated to Subadmin Dashboard with records	Pass
3	Provide Valid Password	Password: user@12			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Username: user123 Password: User@11	User shouldnot be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null Username orPassword	Username: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Code

```
package testcase;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimplement.DriverSetup;
public class Firsts {


public static WebDriver driver;

public static void main(String[] args) {
// TODO Auto-generated method stub

driver =  DriverSetup.getWebDriver("http://localhost/project2/login.php");

//login-Invalid case
driver.findElement(By.name("username")).sendKeys("riya");
driver.findElement(By.name("password")).sendKeys("123");
driver.findElement(By.name("submit")).click();

String actualUrl="http://localhost/Restaurant/custhome.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed"); } else { System.out.println("Test failed"); }
driver.quit();
}
}
```



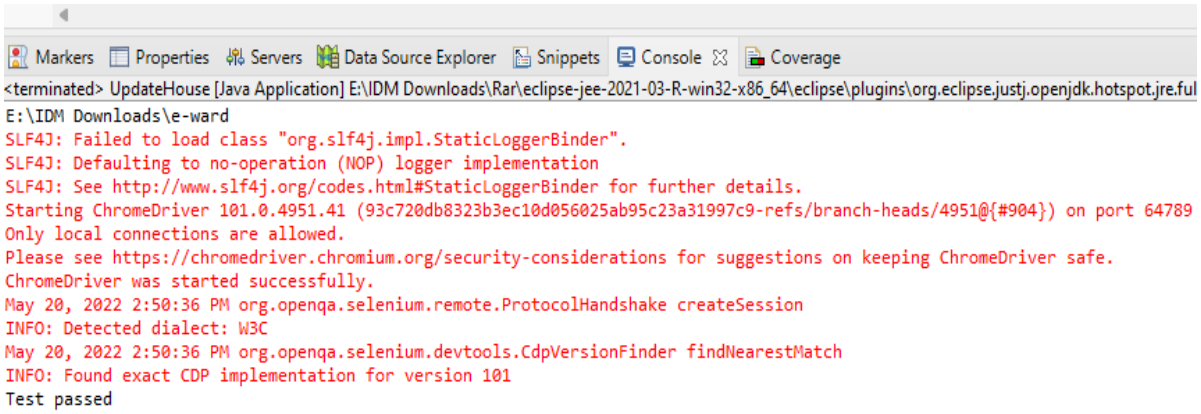
Test Case 2

Project Name: Hotel Room Booking System					
Udpate Test Case					
Test Case ID: Fun_2			**Test Designed By: Riya Abraham**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 24-05-2022		
Module Name: Update			**Test Executed By : Ms Navyamol K.T**		
Test Title : Update Details			**Test Execution Date:** 24-05-2022		
Description: Login to System and update details, if some error occur test will fail.					
Pre-Condition : User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Username	User Name: user	User should be able to Login	User Logged in and navigated to Subadmin Dashboard with records	Pass
3	Provide Valid Password	Password: user@12			
4	Click on Sign In button				
5	Provide room information	Input room details	User will be redirected to dashboard	User will be redirected to dashboard	Pass
6	Clickon update button				
7	Provide invalid details	Input invalid details	User will be redirected to dashboard	User will be stay on page showing error message	Pass
8	Click on Update button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;
public class UpdateRoom {
public static WebDriver driver;
public static void main(String[] args) {
driver = DriverSetup.getWebDriver(&quot;https://e-
ward.000webhostapp.com/public/index.php&quot;);
driver.findElement(By.name(&quot;login&quot;)).click();
driver.findElement(By.name(&quot;userName&quot;)).sendKeys(&quot;2160&quot;);
driver.findElement(By.name(&quot;password&quot;)).sendKeys(&quot;Qwerty@123&q
uot;);
driver.findElement(By.name(&quot;submitButton&quot;)).click();
driver.get(&quot;https://e-
ward.000webhostapp.com/application/view/pages/house_member/update_
room_details.php&quot;);
driver.findElement(By.name(&quot;hname&quot;)).sendKeys(&quot;Dream
Hotels&quot;);
driver.findElement(By.name(&quot;locality&quot;)).sendKeys(&quot;Miami&quot;);
driver.findElement(By.name(&quot;po&quot;)).sendKeys(&quot;Kuthiravattom
po&quot;);
driver.findElement(By.name(&quot;uphbtn&quot;)).click();
String actualUrl=&quot;https://e-
ward.000webhostapp.com/application/view/pages/house_member/dashboa
rd.php&quot;;
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println(&quot;Test passed&quot;);
} else {
System.out.println(&quot;Test failed&quot;);
}
}
```

}


Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Coverage
<terminated> UpdateHouse [Java Application] E:\IDM Downloads\Rar\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.ful
E:\IDM Downloads\e-ward
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 64789
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 20, 2022 2:50:36 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 20, 2022 2:50:36 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In

many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner.

System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The proposed system introduces facility for customer to book the rooms. Provides lots of advantages like search rooms, view profile of customer, enhanced user interface, payment options, add feedback, daily progress report option, and complaint status and may more. Increasing the security and confidentiality of each files and report by providing a security module. The system will provide a monitoring module that will display an overview of the system which is accessible only to the admin.

7.2 FUTURE SCOPE

While virtual reality can be used to attract customers to a specific hotels. From the point of view of sales and advertising, AR is a great way to show offers and promotions in real-time. Viewing of rooms in 3D. The monitoring module will display an overview of the system, the admin is the only authorized person to view and manipulate the content of this module

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 SAMPLE CODE

Customer

viewrooms.php

```
 <?php
include('connection.php');
session_start();
$ids = $_SESSION['hid'];
$sid = $_SESSION['rid'];
$name = $_SESSION['names'];
?>

<!DOCTYPE html>
<html lang="en" dir="ltr">
<body>
 <header class="header">
  <div class="container">
   <nav class="navbar flex1">
    <div class="sticky_logo logo">
     <img src="images/logo.png" width="50" height="50">
    </div>

    <ul class="nav-menu">
     <li> <a href="home.php">Home</a> </li>
     <li> <a href="#services">services</a> </li>
                        <li class="dropdown" style="padding-left: 700px;">
     <a class="dropbtn"><?php echo $name ?></a>
     <div class="dropdown-content">
                                        <a href="#">Profile</a>
                                        <a href="index.php">Logout</a>
                                      </div>
                                </li>
    </ul>
    <div class="hamburger">
     <span class="bar"></span>
     <span class="bar"></span>
     <span class="bar"></span>
    </div>
   </nav>
  </div>
 </header>
<form action="book.php" method="post">
  <div class="main">
   <div class="row">
    <center>
    <?php
```

```
    $result1=mysqli_query($con,"SELECT * FROM `rooms` WHERE hid='$ids'");
     while($row=mysqli_fetch_array($result1))
      {
     echo "<div class='column'><div class='card'>
<div class='image'><img src='./images/".$row['picture']."' width= 200px height=
150px></div>
<p class='roomn' style='padding-left: 30px; color:black;'>".$row['roomname']."</p>
<p class='price' style='padding-left: 30px; color:black;'>".$row['price']."</p><button
type='submit' class='btn' name='submit'><a href='book.php'>BOOK
NOW</a></button> </div></div>";
}
?>
</center>
<br><br>
    </div>
   </div>
  </form>
</body>
</html>
```

Book.php $ Payment.php

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<body>
  <header class="header">
    <div class="container">
      <nav class="navbar flex1">
        <div class="sticky_logo logo">
          <img src="images/logo.png" width="50" height="50">
        </div>

        <ul class="nav-menu">
          <li> <a href="index.php">Home</a> </li>
          <li> <a href="#services">services</a> </li>
          <li class="dropdown" style="padding-left: 700px;">
          <a class="dropbtn"><?php echo $name ?></a>
          <div class="dropdown-content">
          <a href="#">Profile</a>
          <a href="index.php">Logout</a></div>
         </li>
        </ul>
        <div class="hamburger">
          <span class="bar"></span>
          <span class="bar"></span>
          <span class="bar"></span>
        </div>
      </nav>
```

```
    </div>
  </header>
<div>
    <div class="main">
      <div class="row">
        <center>  <div class='column'>
       <div class='card'><?php
       $result=mysqli_query($con,"SELECT * FROM `book`,rooms");
     while($row=mysqli_fetch_array($result))
          {
          ?>
  <label> Check-in Date : </label>
<?php echo $row['checkin'];?>
<br><br>
<label> Check-out Date : </label>
<?php echo $row['checkout'];?>
<br><br>
<label> Amount : </label>
<label id="grandtotal"><?php echo $row['price'];?></label>
<br><br>
<button class="btn" type="submit" name="pay" id="payment">PAY NOW</button>
<?php      }     ?>
</div>    </div>
</center>
<br><br>        </div>   </div>   </div>
  <script src="https://checkout.razorpay.com/v1/checkout.js"></script>
      <script>
  document.getElementById('payment').onclick = function(e) {
  let amt = $('#grandtotal').html();
   var options = { "key": "rzp_test_GhL9PFaTCAYjmm", // Enter the Key ID generated
from the Dashboard
"amount": amt * 100, // Amount is in currency subunits. Default currency is INR. Hence,
50000 refers to 50000 paise
 "currency": "INR",
 "name": "HotelBooking",
  "description": "Hotel Rooms booking",
  "image":
"https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.dreamstime.com%2
Fsimple-illustration-dark-blue-hotel-logo-design-template-business-icon-inspiration-
travel-tourism-sticker-idea-simple-
image165633303&psig=AOvVaw2Tr2HF52IrgC267PmZqgRg&ust=1651811341231
000&source=images&cd=vfe&ved=0CAwQjRxqFwoTCNiAn_XCx_cCFQAAAAAd
AAAAABAD",
                        "handler": function(response) {
                          alert("ok");
                          //success function
                        $.ajax({
                        url: "cartform.php",
```

```
                    type: "POST",
                    data: {
                    razorpay_payment_id: response.razorpay_payment_id,
                    Amount:amt
                    },
                    success: function(data, status) {
                    console.log(data);
                    window.location.href = "hotelsview.php";
                    },
                    error: function(responseData, textStatus, errorThrown) {
                    console.log(responseData, textStatus, errorThrown);
                     }
                 });

                    }
                };
                var rzp1 = new Razorpay(options);
                rzp1.open();
                e.preventDefault();

            }
        </script>
</body>
</html>
```

Hotels

Addroms.php

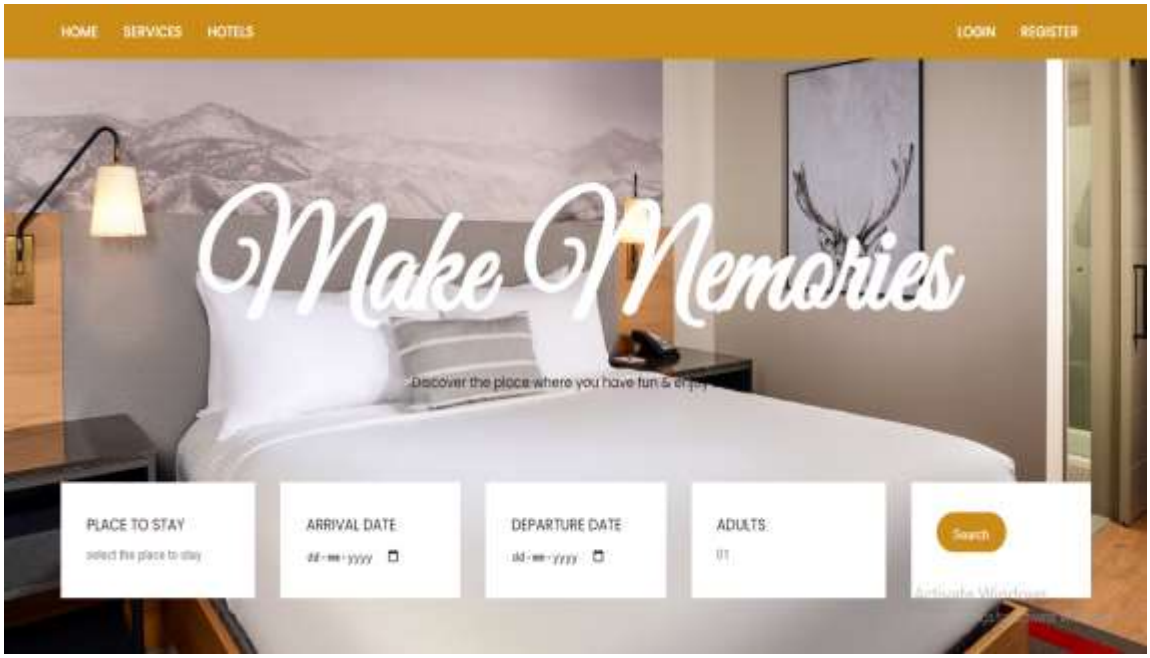
```
<?php
include('connection.php');
  session_start();
  $id = $_SESSION['hid'];
  $name = $_SESSION['hotelname'];

if(isset($_POST["ok"]))
{
echo $id=$_SESSION['hid'];
$rname=$_POST["rname"];
$price=$_POST["price"];
$qua=$_POST["quat"];
$upl=$_FILES["img"]["name"];
move_uploaded_file($_FILES["img"]["tmp_name"],"images/".$_FILES["img"]["name"]
);
mysqli_query($con,"INSERT INTO `rooms`(`hid`, `roomname`, `price`, `qua`,  `picture`)
VALUES ('$id', '$rname', '$price', '$qua', '$upl')");
}
```

```
?>
<!DOCTYPE html>
<html lang="en">
<div id="layoutSidenav_content">     <main>
        <div class="container-fluid px-4">   <br>
           <div class="addroom">
<form action="#" method="post" enctype="multipart/form-data">
<h3>Add Rooms</h3>
 <table>   <tr>    <td>Hotel name</td>
<td><?php echo $name ;?> </td>    </tr>
<tr>      <td>Add Room Type</td>
<td><input type="text" name="rname"></td>   </tr>
<tr>         <td>Price</td>
<td><input type="text" name="price"></td>         </tr>
<tr>          <td>Number of Rooms</td>
<td><input type="text" name="quat"></td></td>    </tr>
<tr>       <td>Picture</td> <td><input type="file" name="img"></td>    </tr>
<tr> <td>     </td>
<td><input type="submit" class="btn1" name="ok" value="Save"></td>  </tr>
 </table>   </form> </div>
</body>   </html>
```

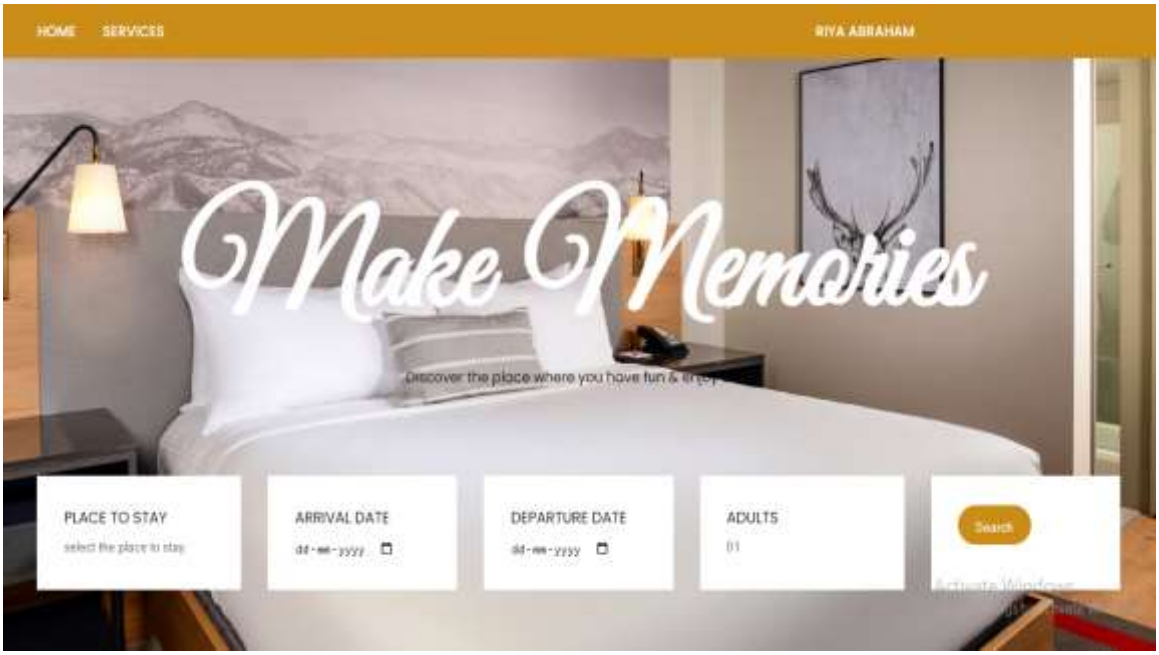
9.2 SCREEN SHOTS

CUSTOMER PAGES

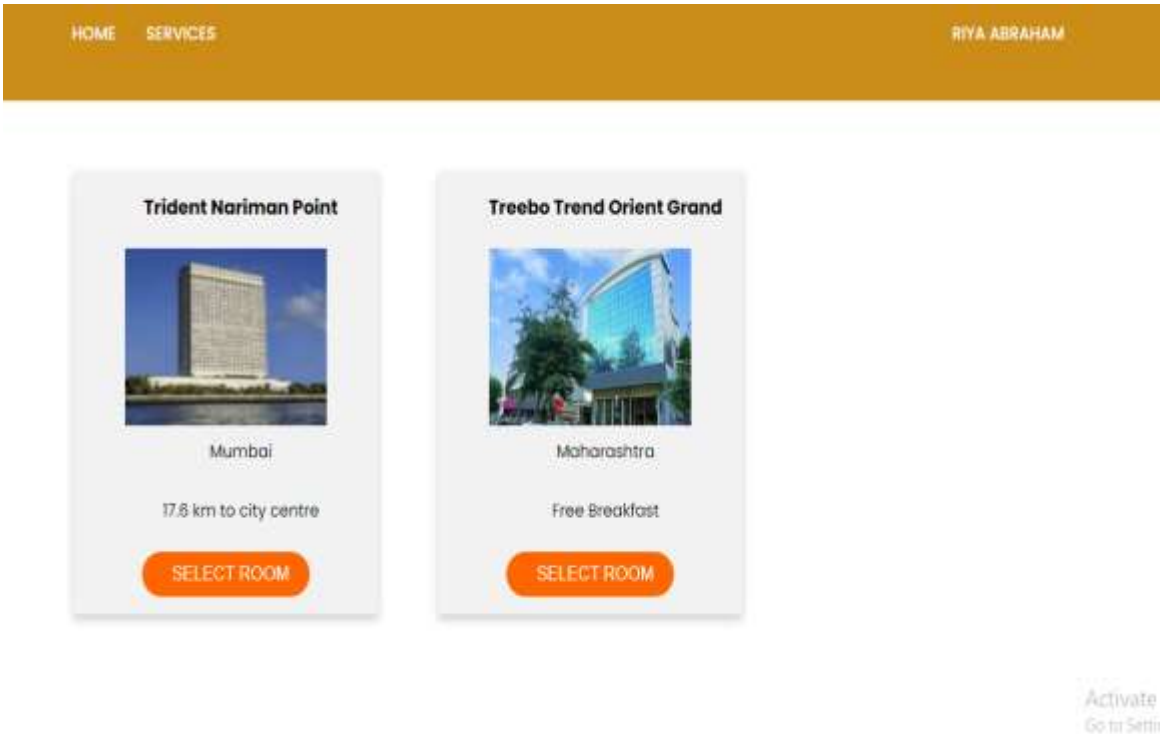
Index Page



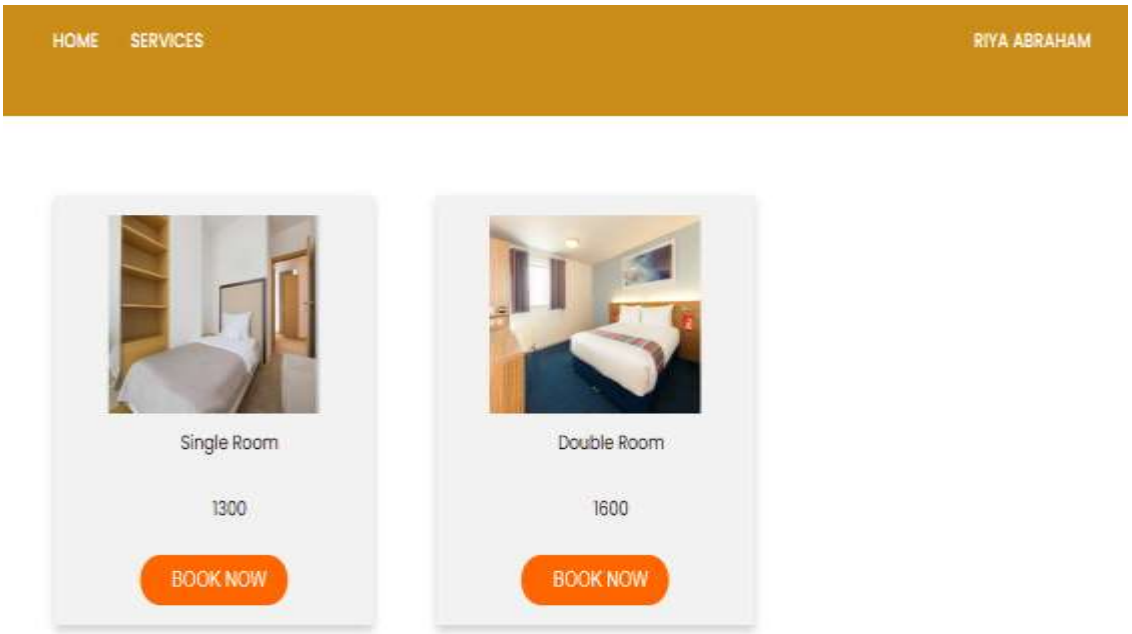
Customer Home page



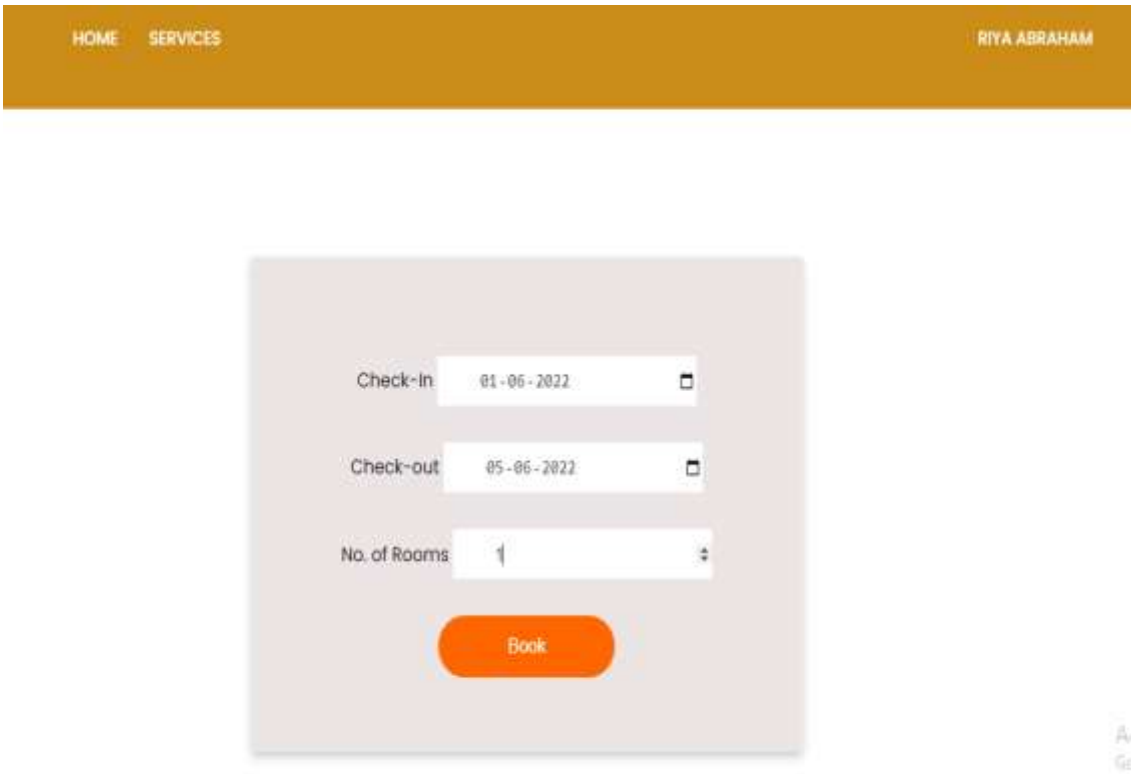
Customer hotel page



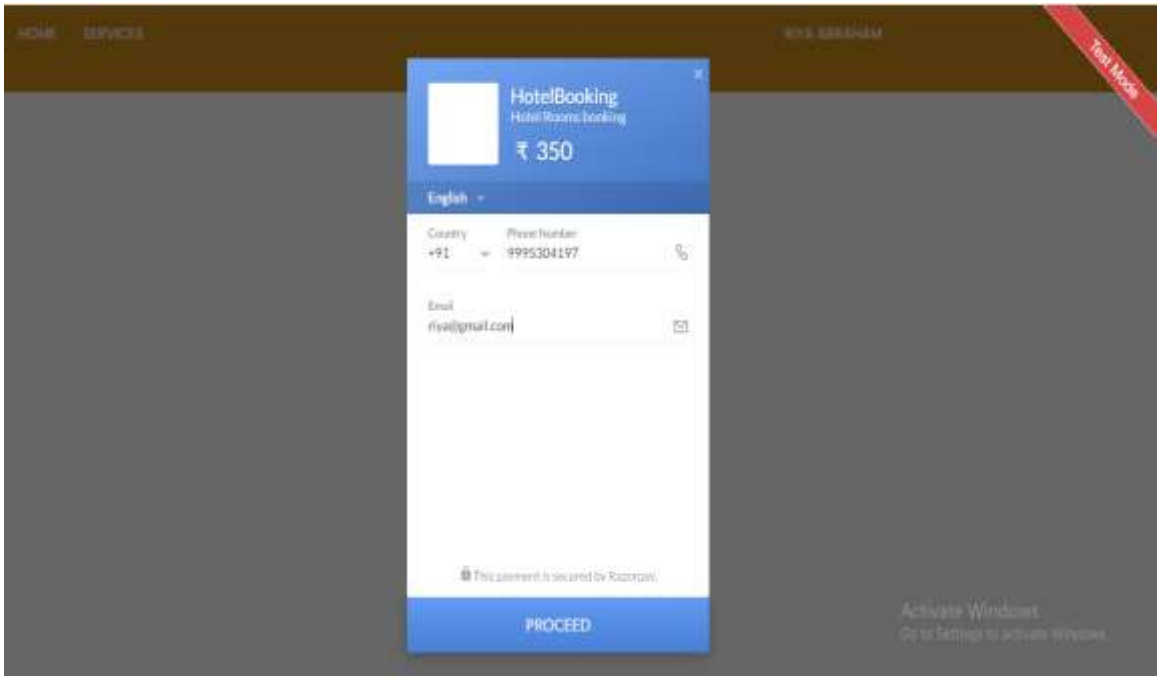
Customer hotel page



Bookform Page

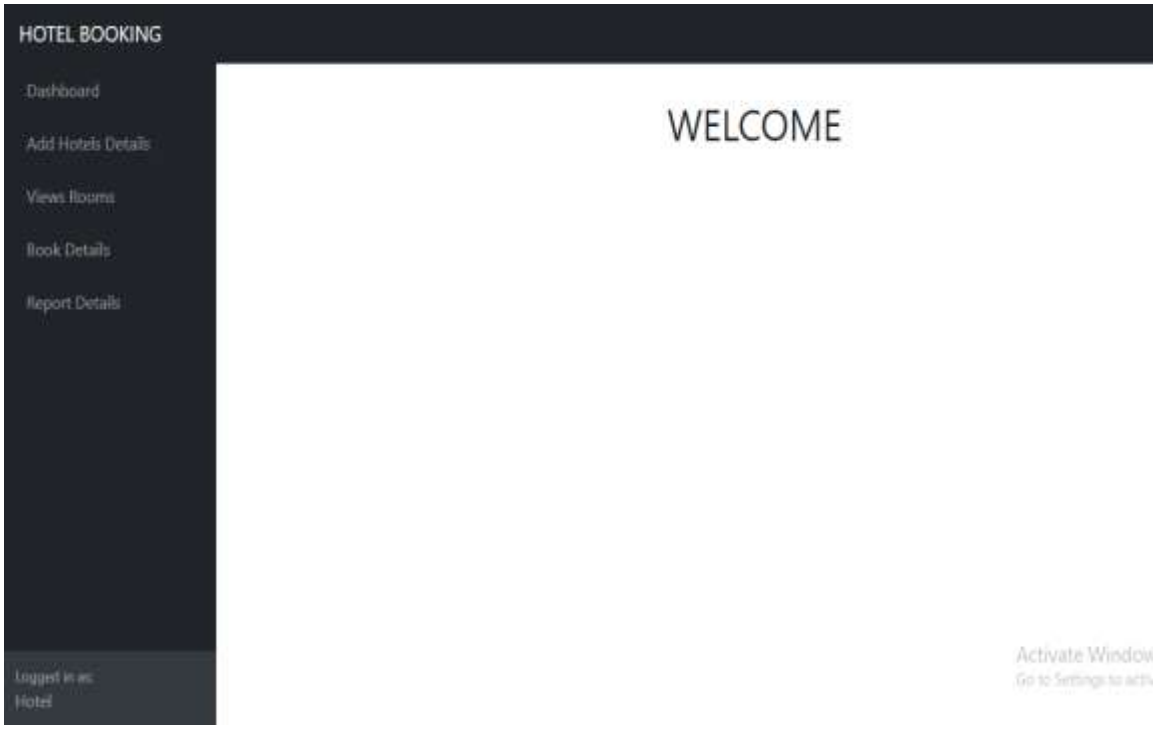


Payment Page

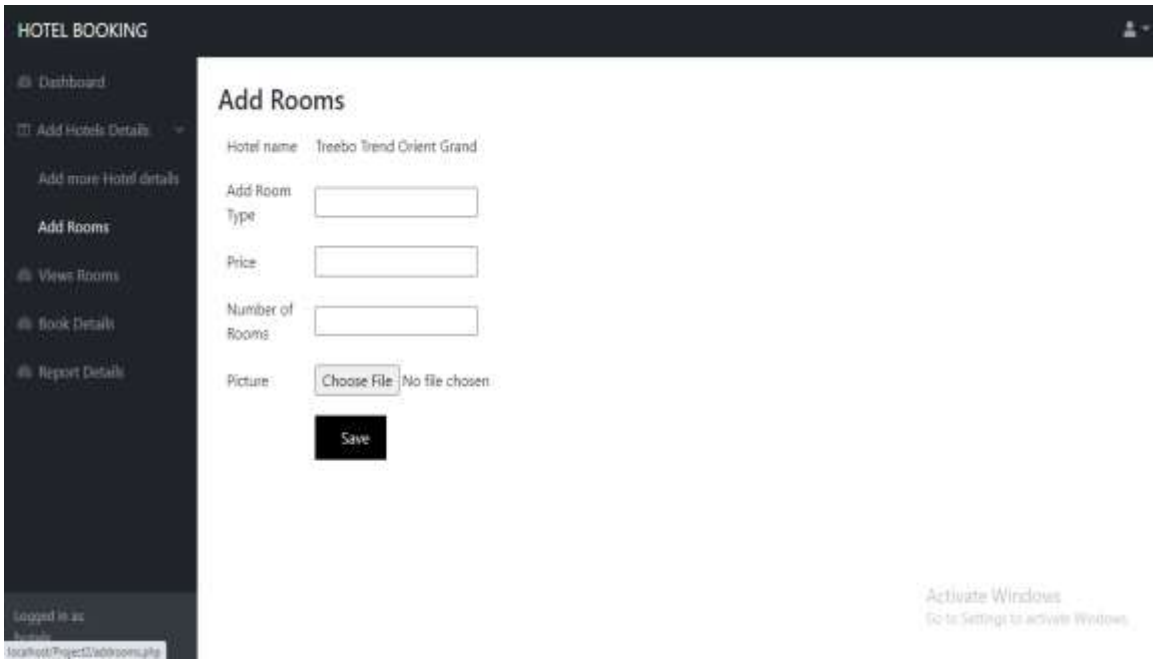


HOTEL PAGES

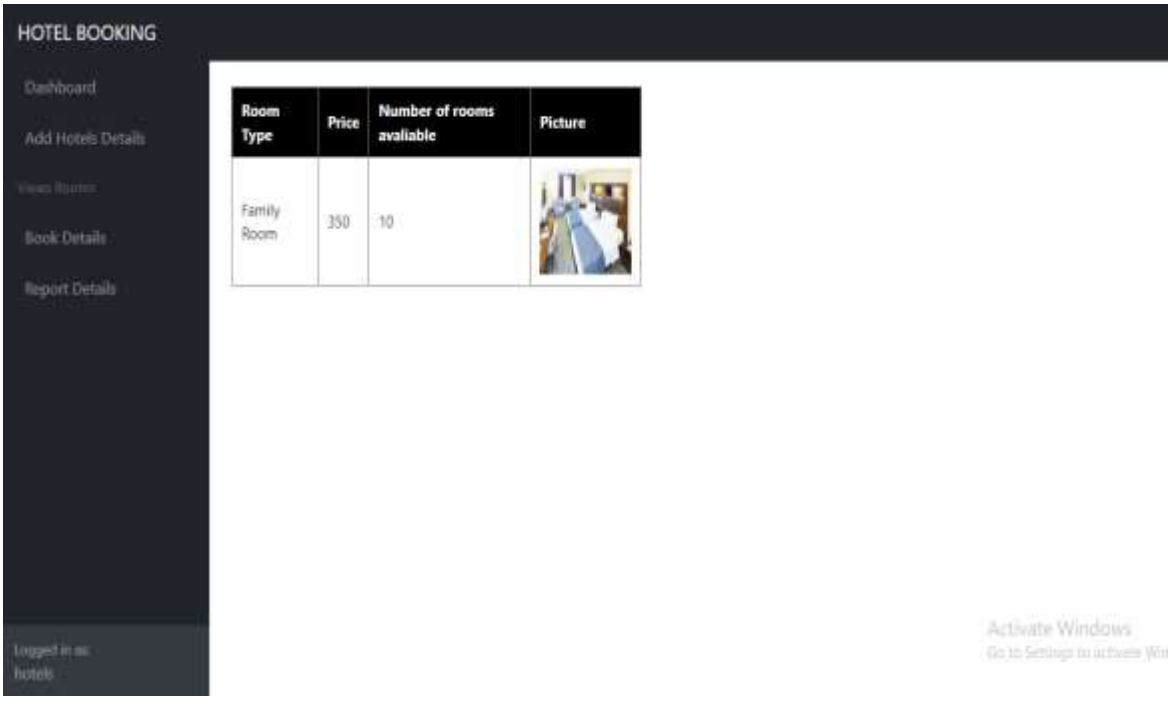
Dashboard Page



Add Room Categories

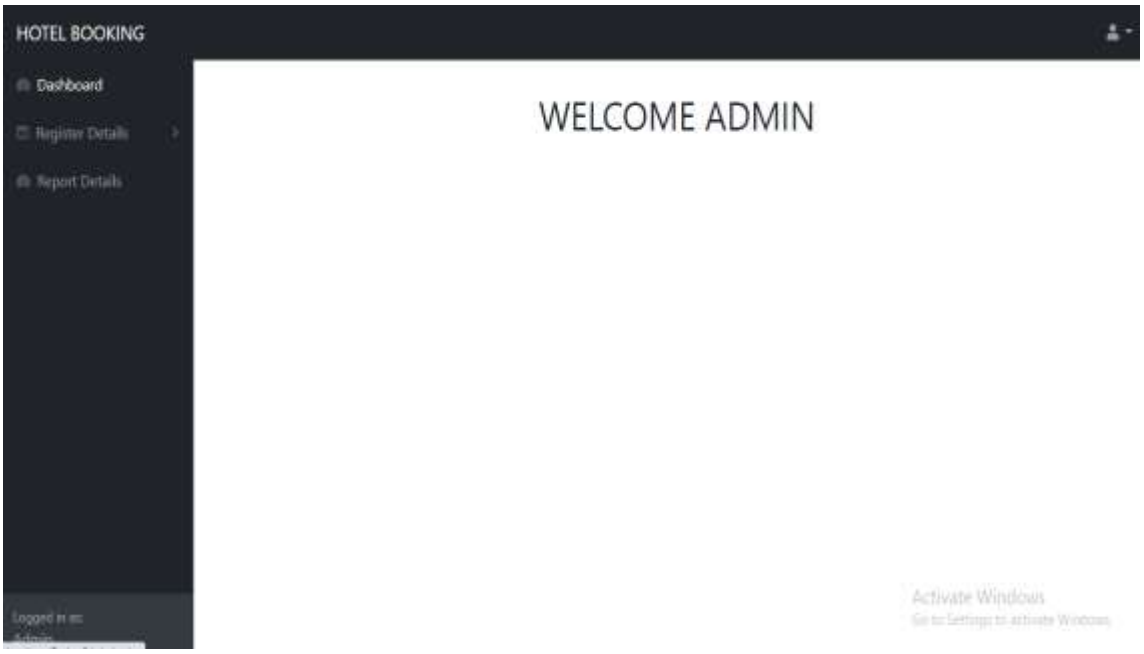


View Room Categories

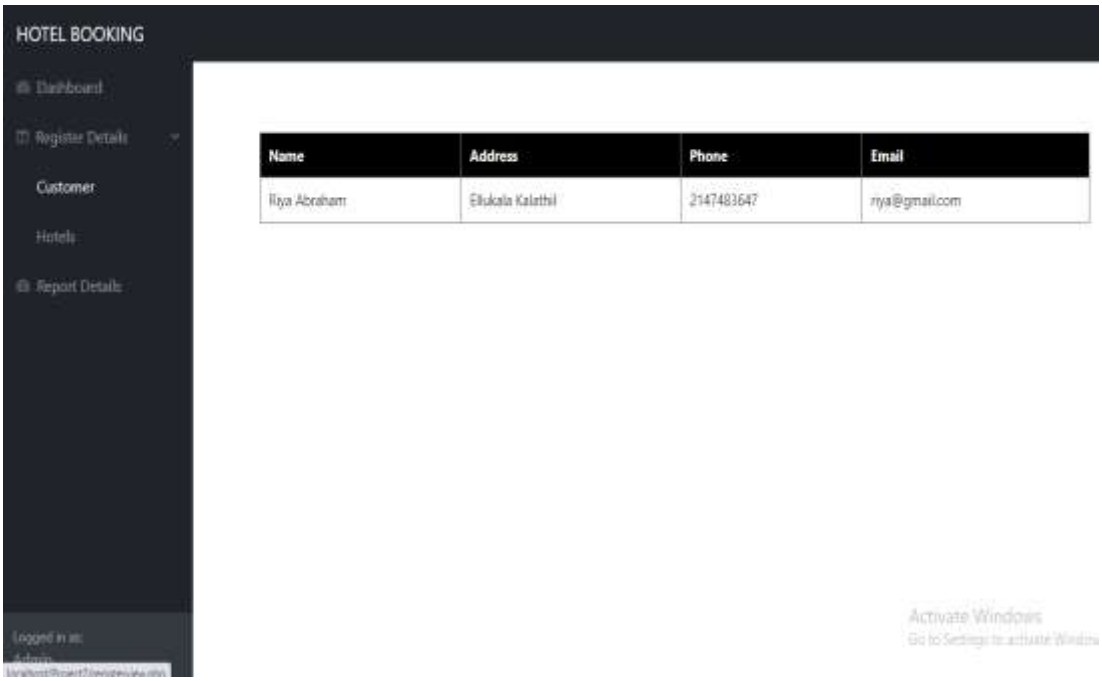


ADMIN PAGES

Dashboard Page



View customer details



View Hotel Details