

HOTEL MANAGEMENT SYSTEM

PROJECT REPORT

18CSE202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY

(2018 Regulation)

II Year/ III Semester

Academic Year: 2022 -2023

By

B. SAI SRI VEER

(RA2111003011789)

V. VENKAT ADITYA

(RA2111003011799)

K. JAYANTH

(RA2111003011802)

Under the guidance of

Dr. P. MURALI

Associate Professor

Department of Computing Technologies



**FACULTY OF ENGINEERING AND TECHNOLOGY
SCHOOL OF COMPUTING SRM INSTITUTE OF SCIENCE
AND TECHNOLOGY Kattankulathur, Kanchipuram**

NOVEMBER 2022

BONAFIDE

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY project report** titled “**HOTEL MANAGEMENT SYSTEM**” is the bonafide work of **B. SAI SRI VEER (RA2111003011789)** and **V. VENKAT ADITYA (RA2111003011799)** and **K. JAYANTH (RA2111003011802)** who undertook the task of completing the project within the allotted time.

Signature of the Guide

Dr. P. MURALI

Department of Computing Technologies

SRM Institute of Science and Technology

Technology

About the course:-

Signature of the II Year Academic Advisor

Associate Professor Professor and Head

Department of Computing Technologies

SRM Institute of Science and

18CSC202J/ 8AIC203J - Object Oriented Design and Programming are 4 credit courses with **L T P C as 3-0-2-4** (Tutorial modified as Practical from 2018 Curriculum onwards)

Objectives:

The student should be made to:

- Learn the basics of OOP concepts in C++
- Learn the basics of OOP analysis and design skills.
- Be exposed to the UML design diagrams.
- Be familiar with the various testing techniques

Course Learning Rationale (CLR): The purpose of learning this course is to:

- 1.Utilize class and build domain model for real-time programs
- 2.Utilize method overloading and operator overloading for real-time application development programs
3. Utilize inline, friend and virtual functions and create application development programs
- 4.Utilize exceptional handling and collections for real-time object-oriented programming applications
- 5.Construct UML component diagram and deployment diagram for design of applications
- 6.Create programs using object-oriented approach and design methodologies for real-time application development

Course Learning Outcomes (CLO): At the end of this course, learners will be able to:

- 1.Identify the class and build domain model
- 2.Construct programs using method overloading and operator overloading
- 3.Create programs using inline, friend and virtual functions, construct programs using standard templates
4. Construct programs using exceptional handling and collections
- 5.Create UML component diagram and deployment diagram
6. Create programs using object oriented approach and design methodologies

Table 1: Rubrics for Laboratory Exercises

(Internal Mark Split-up: - As per Curriculum)

CLAP-1	5=(2(E-lab Completion) + 2(Simple Exercises)(from Code Zinger, and any other coding platform) + 1(HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-2	7.5=(2.0(E-lab Completion)+ 2.0 (Simple Exercises)(from Code Zinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-3	7.5=(2.0(E-lab Completion(80 Programs)+ 2.0 (Simple Exercises)(from Code Zinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	2 Mark - E-lab Completion 80 Program Completion from 10 Session (Each session min 8 program) 2 Mark - Code to UML conversion GCR Exercises 3.5 Mark - Hacker Rank Coding challenge completion
CLAP-4	5= 3 (Model Practical) + 2(Oral Viva)	<ul style="list-style-type: none"> • 3 Mark – Model Test • 2 Mark – Oral Viva
Total	25	

COURSE ASSESSMENT PLAN FOR OODP LAB

S.No	List of Experiments	Course Learning Outcomes (CLO)	Blooms Level	PI	No of Programs in each session
1.	Implementation of I/O Operations in C++	CLO-1	Understand	2.8.1	10
2.	Implementation of Classes and Objects in C++	CLO-1	Apply	2.6.1	10
3,	To develop a problem statement. 1. From the problem statement, Identify Use Cases and develop the Use Case model. 2. From the problem statement, Identify the conceptual classes and develop a domain model with a UML Class diagram.	CLO-1	Analysis	4.6.1	Mini Project Given
4.	Implementation of Constructor Overloading and Method Overloading in C++	CLO-2	Apply	2.6.1	10

5.	Implementation of Operator Overloading in C++	CLO-2	Apply	2.6.1	10
6.	Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams and Collaboration diagrams	CLO-2	Analysis	4.6.1	Mini Project Given
7.	Implementation of Inheritance concepts in C++	CLO-3	Apply	2.6.1	10
8.	Implementation of Virtual function & interface concepts in C++	CLO-3	Apply	2.6.1	10
9.	Using the identified scenarios in your project, draw relevant state charts and activity diagrams.	CLO-3	Analysis	4.6.1	Mini Project Given
10.	Implementation of Templates in C++	CLO-3	Apply	2.6.1	10
11.	Implementation of Exception of Handling in C++	CLO-4	Apply	2.6.1	10
12.	Identify the User Interface, Domain objects, and Technical Services. Draw the partial layered, logical architecture diagram with UML package diagram notation such as Component Diagram, Deployment Diagram.	CLO-5	Analysis	4.6.1	Mini Project Given
13.	Implementation of STL Containers in C++	CLO-6	Apply	2.6.1	10
14.	Implementation of STL associate containers and algorithms in C++	CLO-6	Apply	2.6.1	10
15.	Implementation of Streams and File Handling in C++	CLO-6	Apply	2.6.1	10

LIST OF EXPERIMENTS FOR UML DESIGN AND MODELING:

To develop a mini-project by following the exercises listed below.

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant statecharts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.

Suggested Software Tools for UML:

StarUML, Rational Suite, ArgoUML (or) equivalent, Eclipse IDE and Junit

Table of Contents

no	Contents	Page no
1	ABSTRACT and INTRODUCTION	1
2	PROBLEM DESCRIPTION	2
3	Class Diagram	3
4	Use Case Diagram	4
5	Sequence Diagram	5
6	Collaboration Diagram	7
7	Deployment Diagram	9
8	Activity Diagram	10
9	Conclusion	12

ABSTRACT:

Customer satisfaction is the key to success for any business. In a Hotel, the traditional hand-waving method for calling services is inefficient often leading to many complaints. The Hotel Management System increases operational efficiency through use of an internal wired communications system. The communications system increases customer satisfaction by leaving a system at each table which the customer can use to request for a server. This system allows managers and owners to easily monitor Hotel functions and employee progress.

In many popular Hotels, waiters/waitresses tend to miss out on tables or customers' calls during busy hours potentially decreasing ones clientele. While this is an ongoing issue, there is still no product that drastically improves the communication between the servers and the customers in the current market. Hence, the goal is to design a system in which the customers can call their servers easily and help the Hotel increase overall efficiency.

INTRODUCTION

In many popular Hotels, waiters/waitresses tend to miss out on tables or customers' calls during busy hours potentially decreasing ones clientele. While this is an ongoing issue, there is still no product that drastically improves the communication between the servers and the customers in the current market. Hence, the goal is to design a system in which the customers can call their servers easily and help the restaurant increase overall efficiency. An internal wired communication system will allow prompt notification to the server when a customer requires service. Moreover, servers can also be more focused on serving their current customers and save their time and energy from always keeping an eye out for needy customers.

On top of meeting the needs of customers, Hotel managers can also monitor the response time of their waiters/waitresses through use of this system. Hardworking, proficient employees will become more recognized while lazy, inefficient employees become motivated to improve. As a result, the Hotel becomes more efficient and possibly increases morale while improving the level of customer satisfaction.

PROBLEM DESCRIPTION

Traditionally the method in which customers specify their desired menu to the waiter who takes the order on a paper. Personally, he then takes the order to the kitchen department and then he supplies the food item to the customer. So, it was a time-consuming process. It leads to wastage of paper and also it requires reprinting of all menu cards. Also, in many cases for small changes to be made in menu cards it is not convenient to print all menu cards again and again. Simply saying that the menu card once printed can't be changed. After some days, the menu card lost its worthy look and attractiveness.

Many of the hotel are managed their workflow and services by paper work and manually which take time and high budget for management. busy day of people schedule and value of time become important day by day so Smart Restaurant handle the system as booking and ordering by digitally which become save manpower and time of staff of hotel. Generating KOT by application is easy to save paper, staff and time. The hotel are services and facilized in a traditional way, from waiters to kitchen also on account. More recently, these system is one of the major problem for hotel to manage their manpower and guests.

PROJECT DESCRIPTION

FUNCTIONAL REQUIREMENTS

The Hotel management system should at least include the following features,

1. Allow to add and remove item to order
2. Allow to add and remove table reservation
3. Allow to add and remove menu items
4. Generate bills
5. Manage payment of bills

6. Generate kitchen order tickets

NON-FUNCTIONAL REQUIREMENTS:

Usability

The system should provide an interactive user-friendly interface that is easily understandable for all users.

Availability

The System should be available at least during the restaurant operating hours and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

Dependability

The system should provide consistent performance with easy tracking of records and updating of records.

Maintainability

The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible.

Security

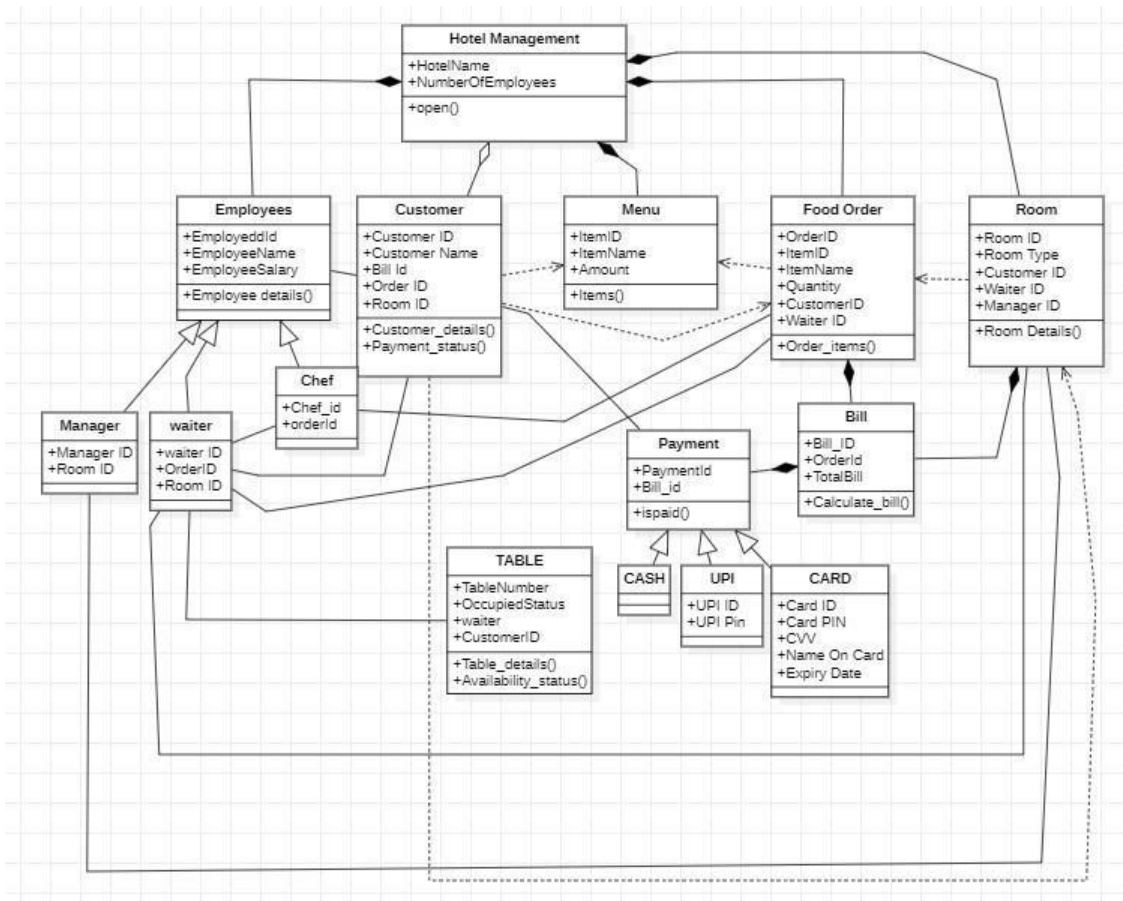
Only authorized users must be able to access the system and view and modify the data.

CLASS DIAGRAM:

Class Diagram of Hotel Management System

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 modeling 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. The various classes required for the execution of the system are mentioned below. By creating the necessary classes, we can create the program required to execute the program.

The customer class contains information about the customer and can able to modify the user details. The menu class contains the details of the item and shows the price of it. The payment class gives the information about the type of the payment and the progress in the payment.



USE CASE DIAGRAM:

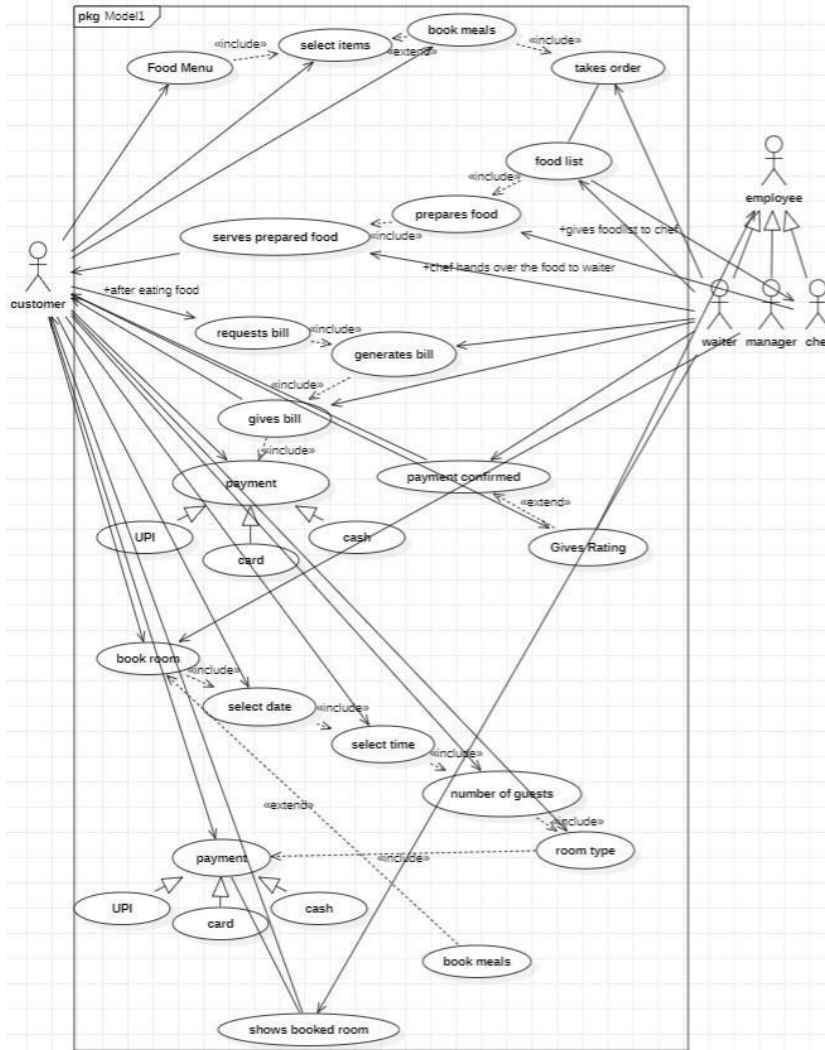
Use Case Diagram of Hotel Management System

Use cases are used to describe how the system should behave under stimuli. The intent is to anchor the process of domain analysis in actual work that needs to be done by the system. As you will see, the uses cases drive Rosenberg's design method at every stage. Note that uses cases are part of the "domain model" not the design model. This is to say they record how an actor might interact with the objects that make up the problem domain. Naturally

We will attempt to structure our design so that some of the classes of our design, the so-called “entity” classes, correspond to the problem domain.

In this particular use case diagram there are five actors that provide the necessary input to the system. The customer performs tasks like searching for the required items and making payments if the item has been purchased. The employee actor performs the task such as managing details about various items present in the restaurant and managing customer data. The waiter checks the various items present in a Hotel and handles the requirements aspect of the Hotel. The manager is responsible for the actual sale of items and manages the finances in the Hotel.

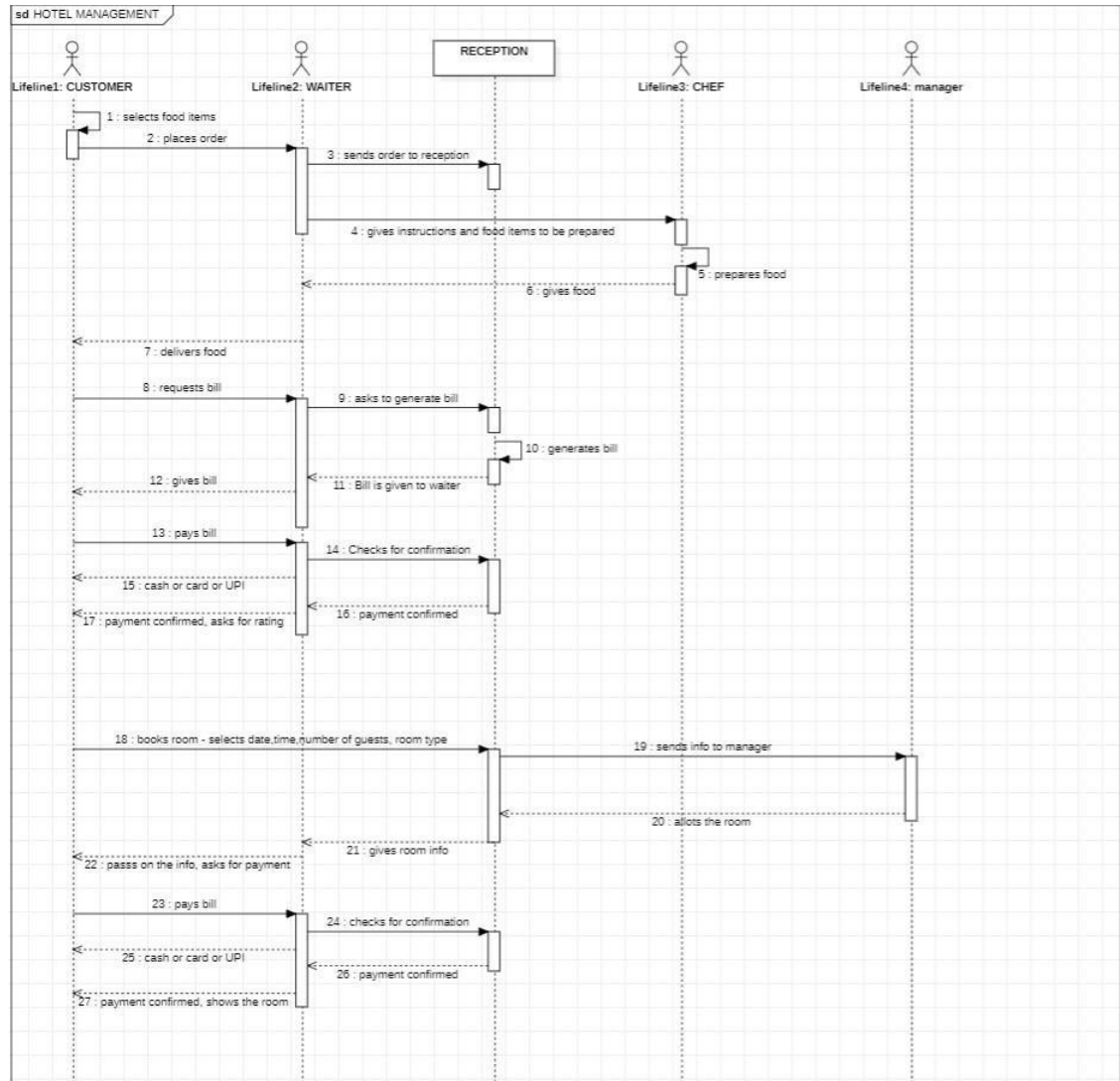
This Use Case Diagram depicts the relationships between the pieces of the Hotel Management System graphically. It is the process used in system analysis to discover, clarify, and organize Hotel Management System requirements. The major components of the Hotel Management System UML use case diagram are depicted in the graphic below.



SEQUENCE DIAGRAM:

Sequence Diagram of Hotel Management System

The sequence diagram shows the process involved for performing a particular action. This is the ' Food order sequence diagram of the hotel management system' , where the customer can select the food items to order with the waiter and the food prepared by the chef and return to the customer. After the food is delivered the customer can access the type of the payment and check the availability of the rooms and pay the bill for it.



Checklist

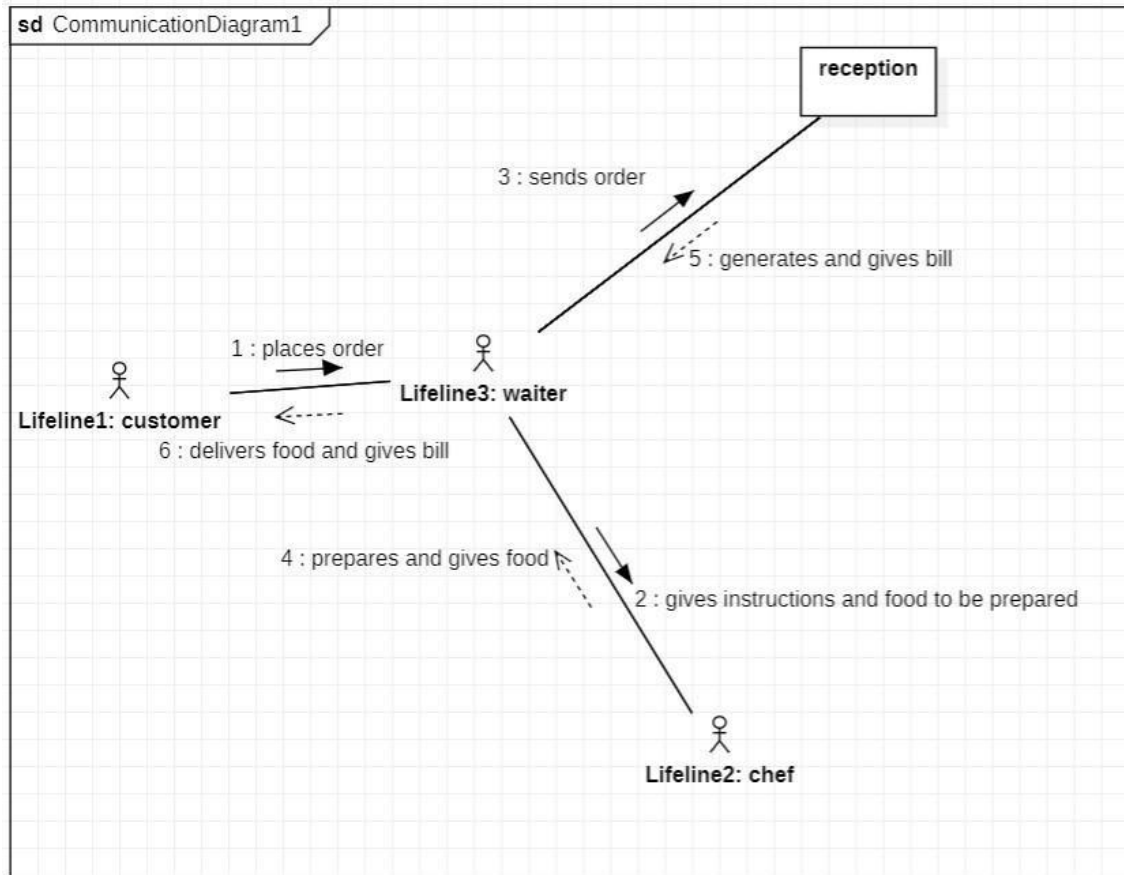
S. No.	Sequence Checklist	Check(Y/N)
1	Messages are from Left-To-Right	Yes
2	Actors named consistently with Use Case Diagram	Yes
3	Classes named consistently with class Diagram	Yes
4	Human and Organisation actors on left most side	Yes
5	Reactive system actors on right most	Yes
6	Proactive system actors on left most side	Yes
7	Message names beside arrowhead justified	Yes
8	Do not return value when it is obvious	Yes
9	Use return value only when you need to refer it elsewhere	Yes

COLLABORATION DIAGRAM:

Collaboration Diagram of Hotel Management System

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.

The given collaboration diagram shows the process of ordering the food with the waiter and performing necessary transactions for the item that is ordered. It also gives information from the waiter about the food to be prepared by the chef. The chef prepares the given food item and the waiter delivers food and gives the bill which was given by the reception.



CHECK LIST :

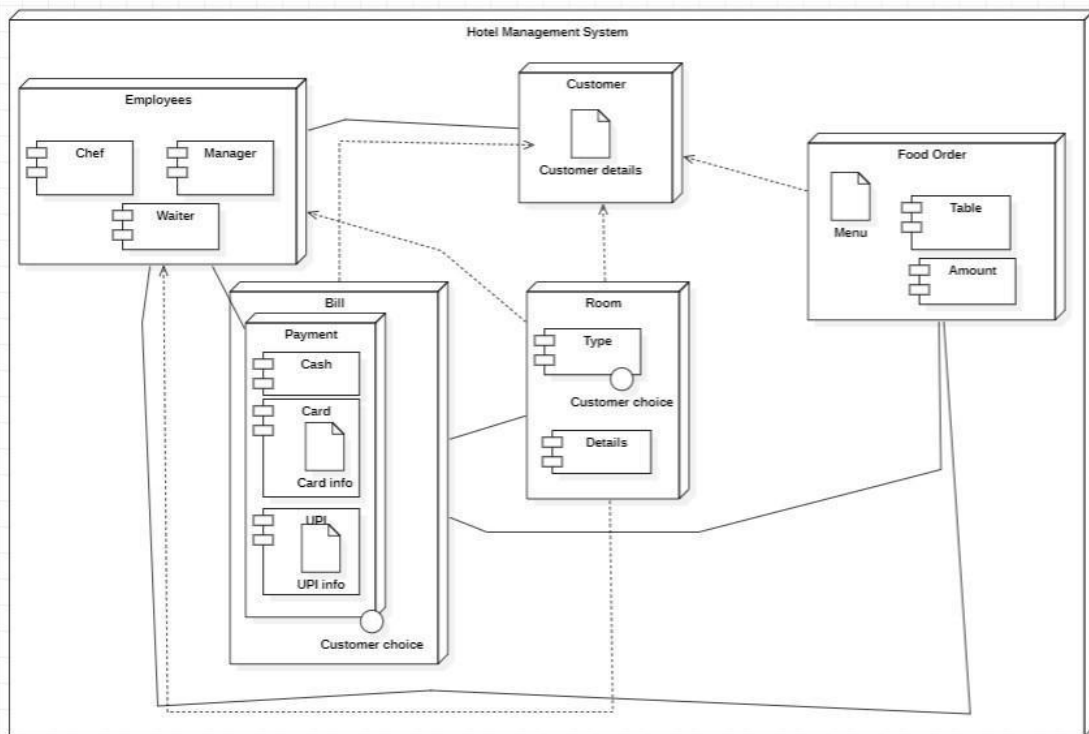
S. No.	Communication Checklist	Check(Y/N)
1	Name objects when referred to in messages	Yes
2	Name objects when several of same type exist	Yes
3	Do not model return value when it is obvious	Yes
4	Model return value when you need to refer to elsewhere	Yes
5	Indicate return value when it is not clear	Yes
6	Indicate parameters when they are not clear	Yes
7	Depict arrow for each message	Yes
8	Indicate navigability sparingly	Yes
9	Prefer roles on links instead of within classes	Yes

DEPLOYMENT DIAGRAM:

Deployment Diagram of Hotel Management System

In UML, deployment diagrams model the physical architecture of a system. Deployment diagrams show the relationships between the software and hardware components in the system and the physical distribution of the processing. Deployment diagrams, which you typically prepare during the implementation phase of development, show the physical arrangement of the nodes in a distributed system, the artifacts that are stored on each node and the components and other elements that the artifacts implement. Nodes represent hardware devices such as computers, sensors, and printers, as well as other devices that support the runtime environment of a system.

Communication paths and deploy relationships model the connections in the system.

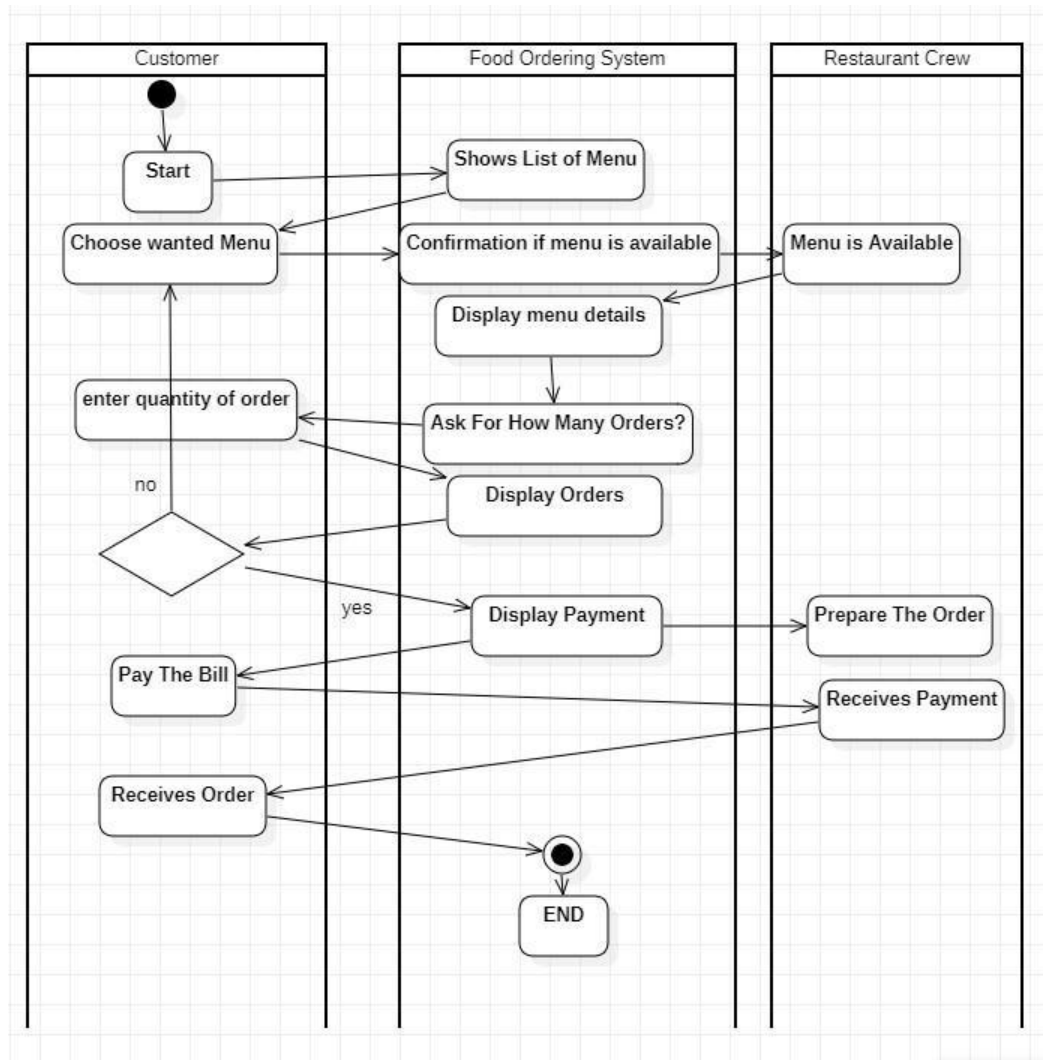


ACTIVITY DIAGRAM:

Activity Diagram of Hotel Management System

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 use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram.

The given activity diagram shows the process of ordering the food with the waiter and performing necessary transactions for the item that is ordered. It also gives information from the waiter about the food to be prepared by the chef. The chef prepares the given food item and the waiter delivers food and gives the bill which was given by the reception.



CHECK LIST :

S.NO	Activity Checklist	Check(Y/N)
1	Place start point on top left corner	Yes
2	Always include end point	Yes
3	Simplify Flow charting operations	Yes
4	Decision points should reflect previous activity	Yes
5	Avoid Superfluous decision points	Yes
6	Each transition leaving point must have a guard	NIL
7	Guards should not overlap	NIL
8	Guards on decision points must form a complete set	NIL
9	Exit transition guards and Activity Invariants must form a complete set	NIL
10	A fork should have a corresponding join	NIL
11	Forks only have one entry transition	NIL
12	Joins only have one exit transition	NIL
13	Avoid superfluous forks	NIL
14	Order Swim lanes in logical manner	NIL
15	Apply swim lanes to linear processes	NIL
16	Have less than 5 swim lanes	NIL
17	Consider horizontal swim lanes for business processes	NIL

CONCLUSION:

In the preceding chapters a variety of views and developments regarding hotel management has been presented. The hotel management system allows the user to order the food in the menu to the waiter. This software allows storing the details of all the data related to the Hotel. The implementation of the system will reduce data entry time and provide readily calculated reports.