

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

FACULTY OF ENGINEERING & TECHNOLOGY

MINI PROJECT

SUBJECT CODE: 18CSC202J

SUBJECT TITLE: OBJECT ORIENTED DESIGN AND PROGRAMMING

PROJECT TITLE

BY

TEAM MEMBERS

SHUBHANKAR MISHRA

SANGAM SMILIKA REDDY



SRM University, SRM Nagar, Kattankulathur-603203

Chengalpattu District, Tamil Nadu

Rubrics

Experiment Component	Max. Marks	Grading Rubrics		
Documentation/ Procedure	10	UML Diagrams are well documented. The documentation supporting all functional requirement and non-functional requirement (10 Marks)	Missing two or more required functional requirement. The documentation work not up to the mark. (5 Mark)	
Concept	5	Completeness of concept, consistent variable naming and relationship in static view. (5 Marks)	Completeness of concept, inconsistent variable naming and relationship in static view. (3 Marks)	Incomplete static view. (1 Mark)
Usage of Symbols	3	Precise usage of symbols in dynamic view. (3 Marks)	Improper usage of Symbol's. (2 Marks)	Symbol's misplaced in diagram. (1 Mark)
Diagrams	4	Completion of all 8 UML Diagrams using Visual Paradigm Tool. (4 Marks)	Construction of UML Diagrams using other tools. (2 Marks)	Construction of few diagrams. (1 Mark)
Viva and Innovative Idea	3	Oral Viva and Innovative approach. (3 Marks)	Oral Viva and partial idea. (2 Marks)	Oral Viva not fulfilled. (1 Mark)
TOTAL	25			

BONAFIDE

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY project report** titled “**ELEVATOR CONTROL SYSTEM**” is the bonafide work of **MEMBER 1 (REG NO), MEMBER 2 (REG NO)** who undertook the task of completing the project within the allotted time.

Signature of the Guide

Dr. Gouthaman. P

Assistant Professor

Department of NWC

SRM Institute of Science and Technology

Signature of the II Year Academic Advisor

Professor and Head

Department of NWC

SRM Institute of Science and Technology

About the course: -

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

LIST OF EXPERIMENTS FOR UML DESIGN AND MODELLING:

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 state charts 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, Argo UML (or) equivalent, Eclipse IDE and Junit

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	Rubrics	ii
	About the Course	iv
	List of Experiments	v
	Abstract	1
	Module Description	2
1	Use Case & Class Case Diagram	3
2	Sequence & Communication Diagram	4
3	State Chart & Activity Diagram	5
4	Package, Component & Deployment Diagram	6
	Conclusion	7
	References	8

ABSTRACT

This Management System was intended to help in creating a simplified blueprint of a system software that works on establishing a proper setup for a Hotel's business. The software itself is a system for hotels where they can deep link to from their own website to let visitors book a room in the hotel. This means the system is very simple-minded, and no introduction is necessary, because once a customer enters, they have already pressed the "Book Now" button on the hotel's front page. The system will simply display a series of forms that will query the user about dates, rooms, additions, etc., and the hotel management can perform certain tasks within the digital frame.

In this project, we will not go into the exact details of the interface, but we will give a description meant for a programmer who needs to perform maintenance on the system, to help him understand the software faster. This method can be used to create these ecosystems for different other projects as well. UML diagrams allows us to explore and create a full fledged environment for a system so that it becomes programmable and could be simplified to create. The Hotel Management system consists of different departments that are connected to the user interface. This also includes the Inventory and Kitchen where the chef processes the orders. We have briefly connected those domains to create an expanded and simple software.

MODULE DESCRIPTION

The Hotel Management System Consists of Many Different sub-parts.

Firstly, it has a Booking System where a customer can confirm the rooms that they want to stay in. Then there is a Food Ordering system where the user can connect to the chef and could process their food requests. There is also an Inventory and an Admin management system where the Data for the working gets stored and compiled.

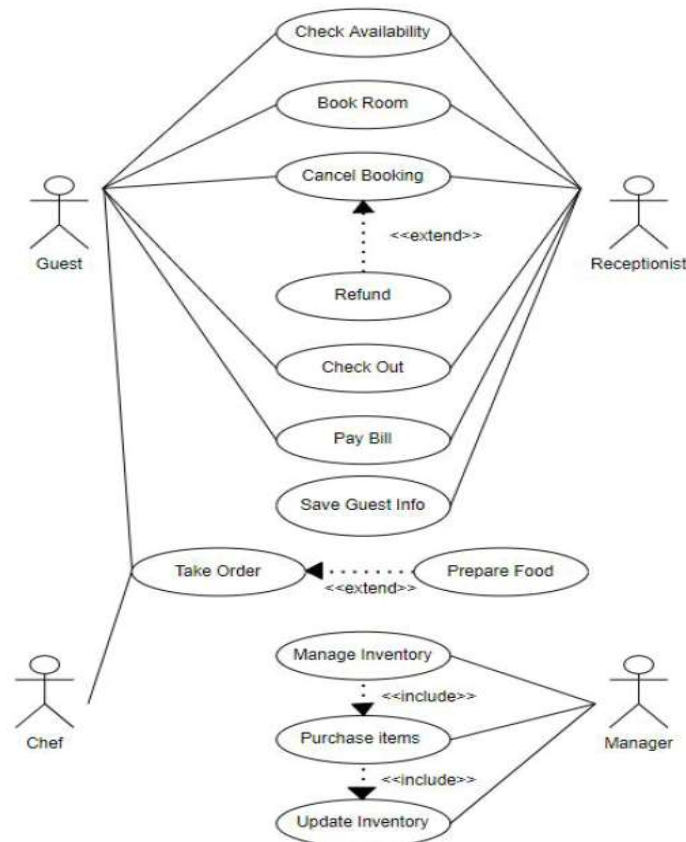
In this UML project, We have worked on the following Diagrams:

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- State Chart Diagram
- Activity Diagram
- Package Diagram
- Component Diagram
- Deployment Diagram

CHAPTER 1

USE CASE DIAGRAM

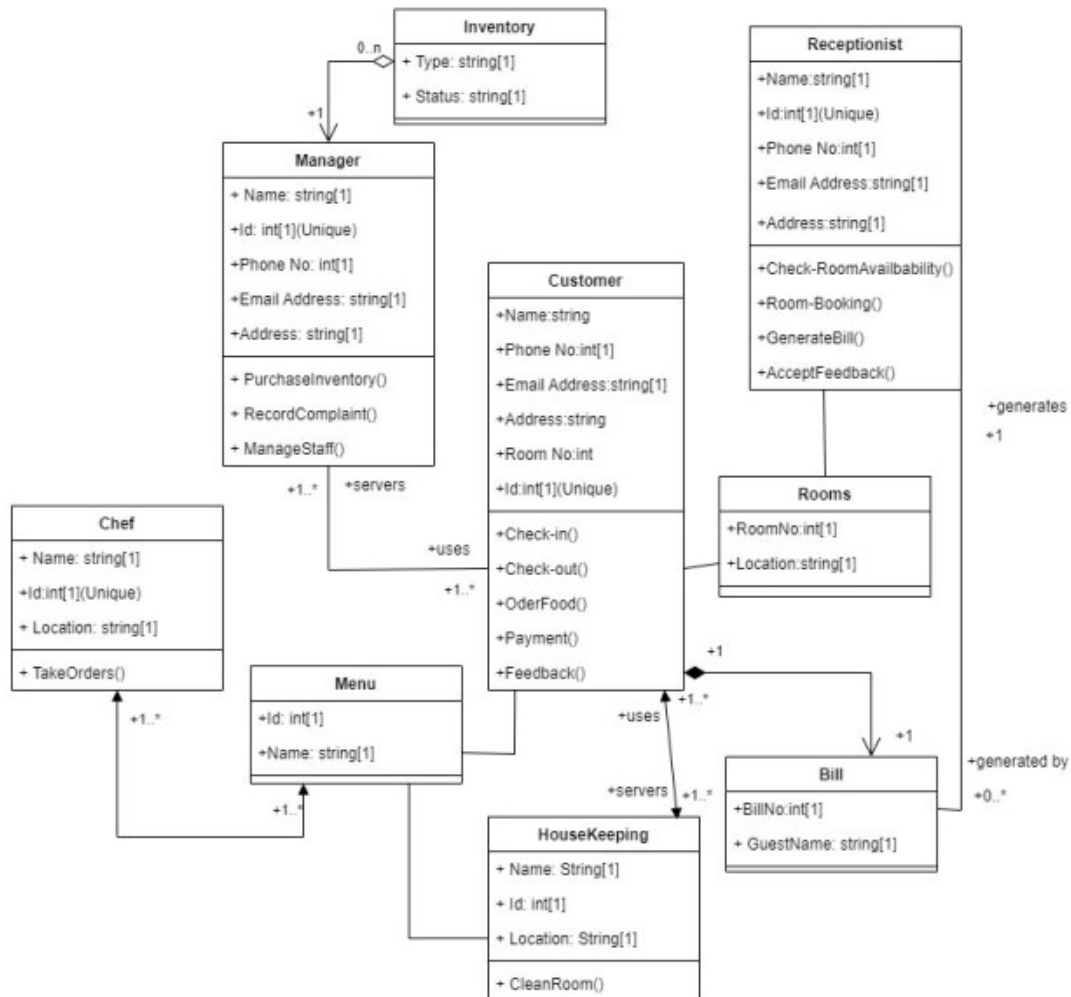
This Use Case Diagram is a graphic depiction of the interactions among the elements of Hotel Management System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Hotel Management System.



The main actors of Hostel Management System in this Use Case Diagram are: Guest, Receptionist, Chef, Hotel Manager, who perform the different type of use cases such as Manage Hotel, Manage Rooms, Manage Payments, Manage Hotel Facility, Manage Users and Full Hotel Management System Operations. Major elements of the UML use case diagram of Hostel Management System are shown on the picture below.

CLASS DIAGRAM

Hotel Management System Class Diagram describes the structure of a Hotel Management System classes, their attributes, operations (or methods), and the relationships among objects.

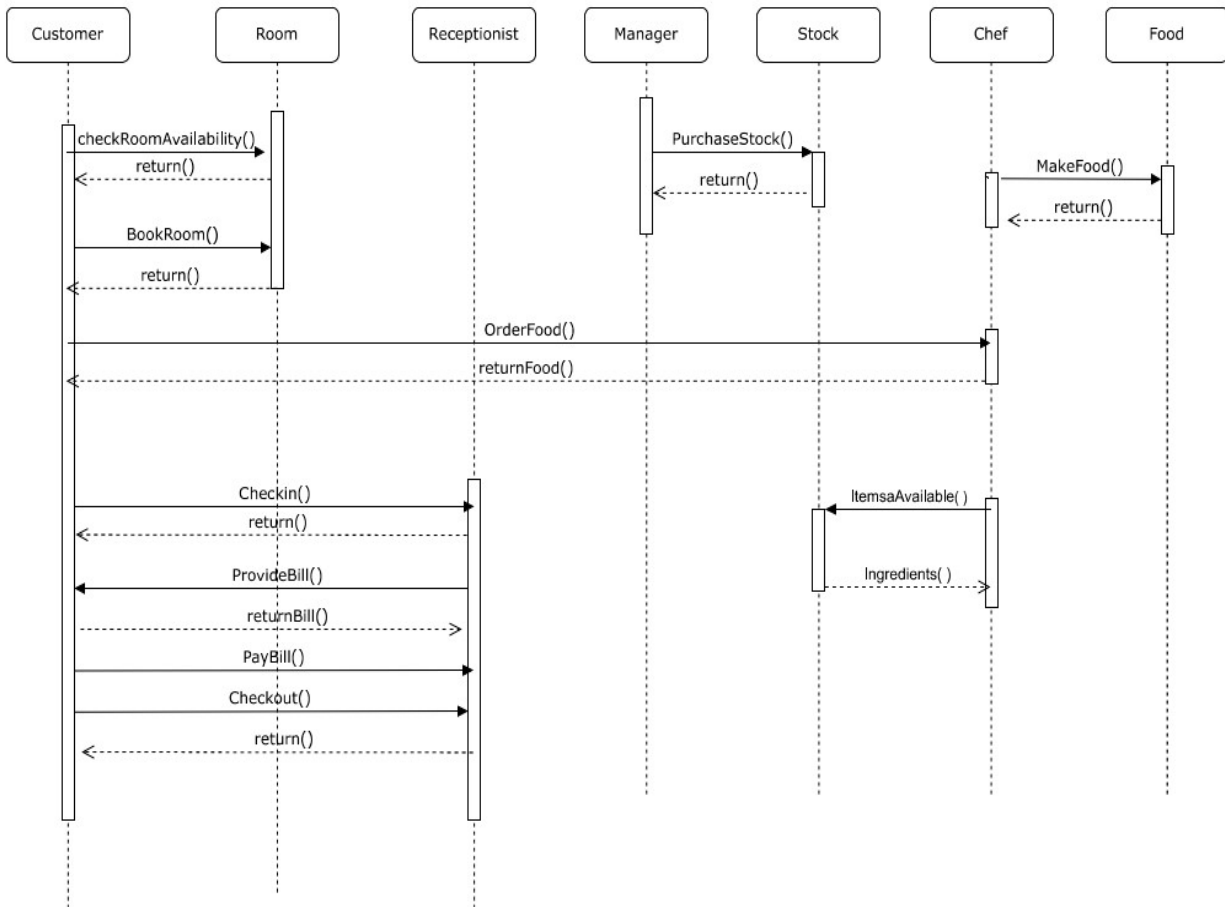


The main classes of the Hotel Management System are Hotel, Rooms, Payments, HouseKeeping, etc.

CHAPTER 2

SEQUENCE DIAGRAM

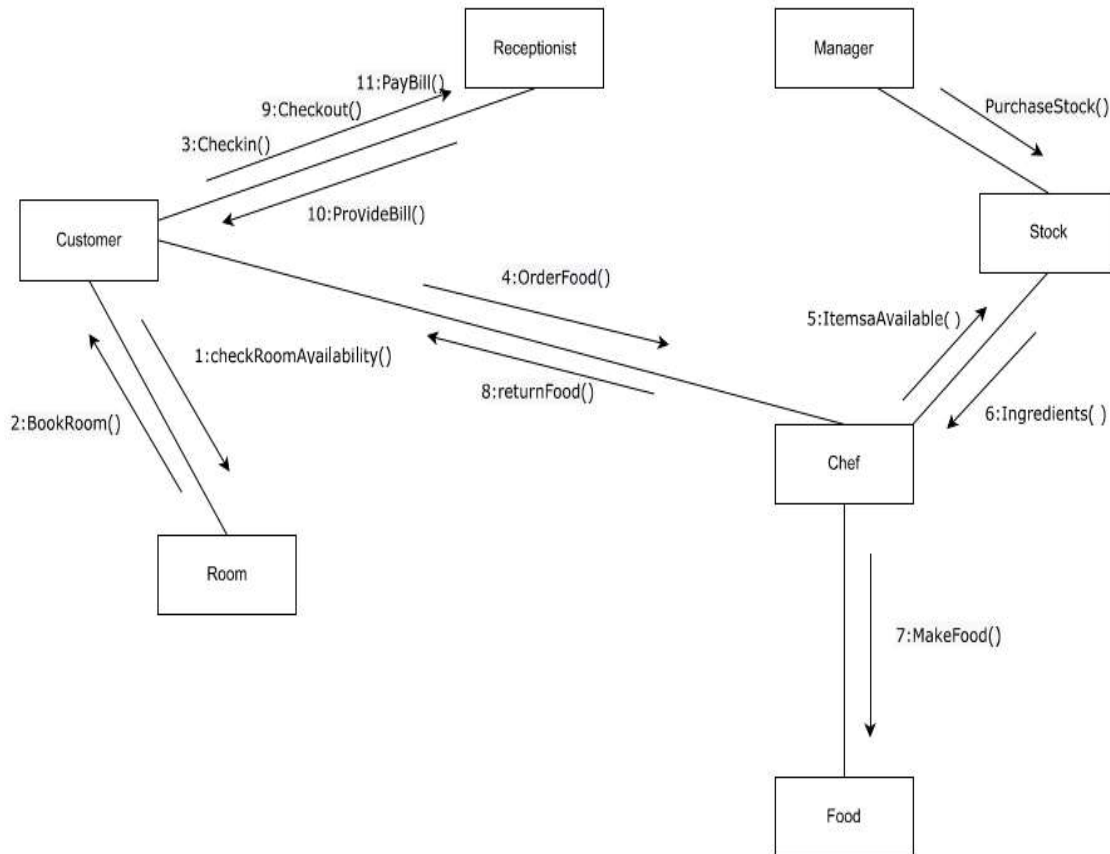
This is the UML sequence diagram of Hotel Management System which shows the interaction between the objects of Rooms, Hotel, Payments.



The instance of class objects involved in this UML Sequence Diagram of Hotel Management System are as follows:

COMMUNICATION DIAGRAM

This is a Communication Diagram example for hotel reservations. Communication diagrams are also interaction diagrams. They convey the same information as sequence diagrams, but they focus on object roles instead of the times that messages are sent. In a sequence diagram, object roles are the vertices and messages are the connecting links.

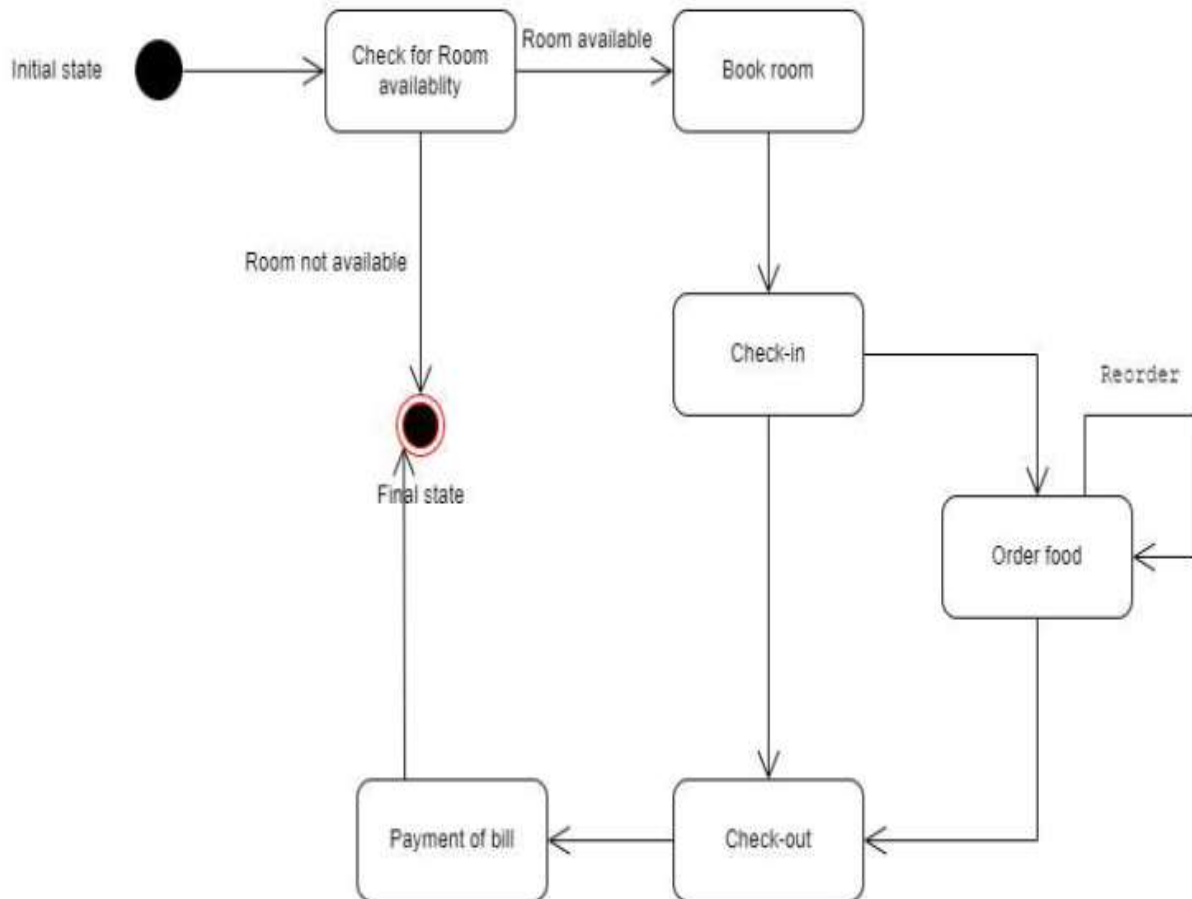


The object-role rectangles are labelled with either class or object names (or both). Each message in a collaboration diagram has a sequence number. The top-level message is numbered 1. Messages at the same level (sent during the same call) have the same decimal prefix but suffixes of 1, 2, etc. according to when they occur.

CHAPTER 3

STATE CHART DIAGRAM

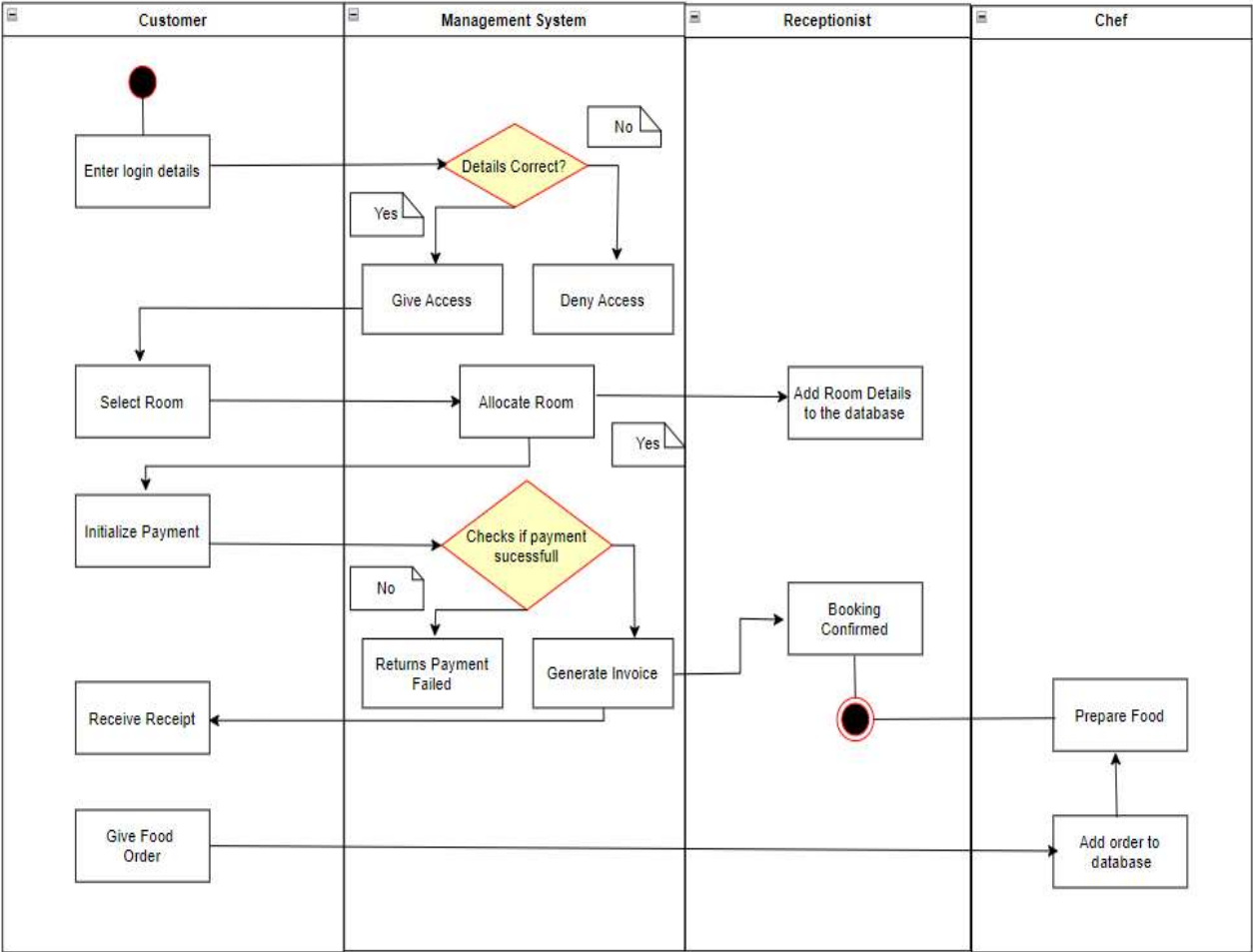
A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioural diagram and it represents the behaviour using finite state transitions.



So simply, a state diagram is used to model the dynamic behaviour of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams. Here we can see the condition of the Hotel Management system.

ACTIVITY DIAGRAM

This is the Login Activity Diagram of the Hotel Management System, which shows the flows of Login Activity, where the admin will be able to log in using their username and password. After login user can manage all the operations on hotels, Rooms, and Payments.

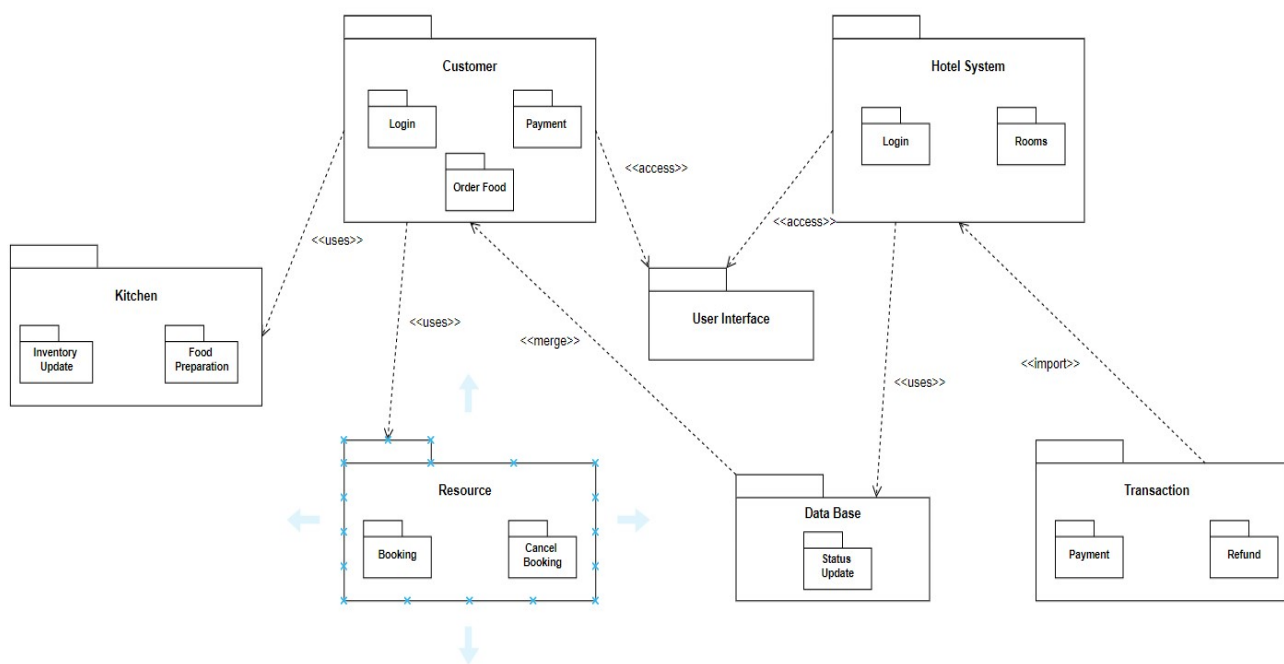


All the pages such as Rooms, and Payments are secure and users can access these pages after login. The diagram below helps demon- state how the login page works in a Hotel Management System. The various objects in the Rooms, Hotel, Hotel Facility and Payments page interact over the course of the Activity, and user will not be able to access this page without verifying their identity.

CHAPTER 4

PACKAGE DIAGRAM

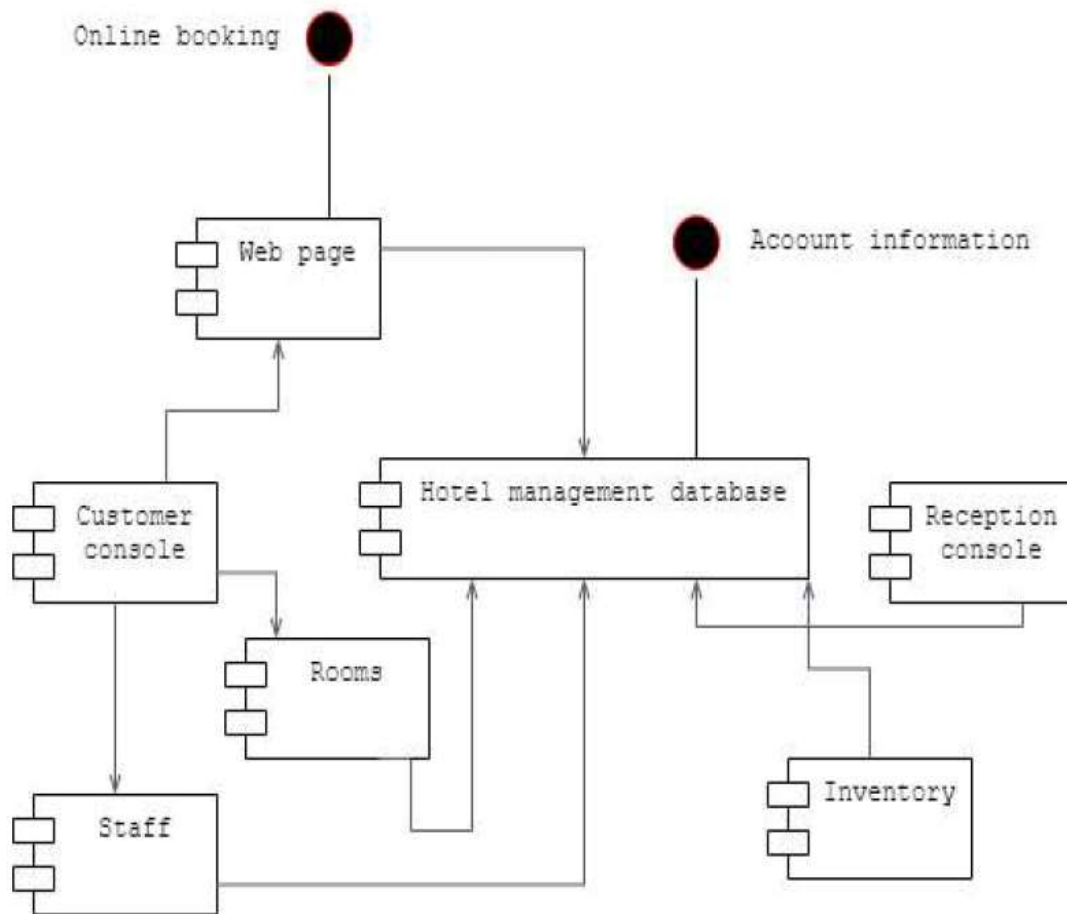
Package diagrams are used to reflect the organization of packages and their elements. When used to represent class elements, package diagrams provide a visualization of the name spaces.



The most common use for package diagrams is to organize use case diagrams and class diagrams, although the use of package diagrams is not limited to these UML elements

COMPONENT DIAGRAM

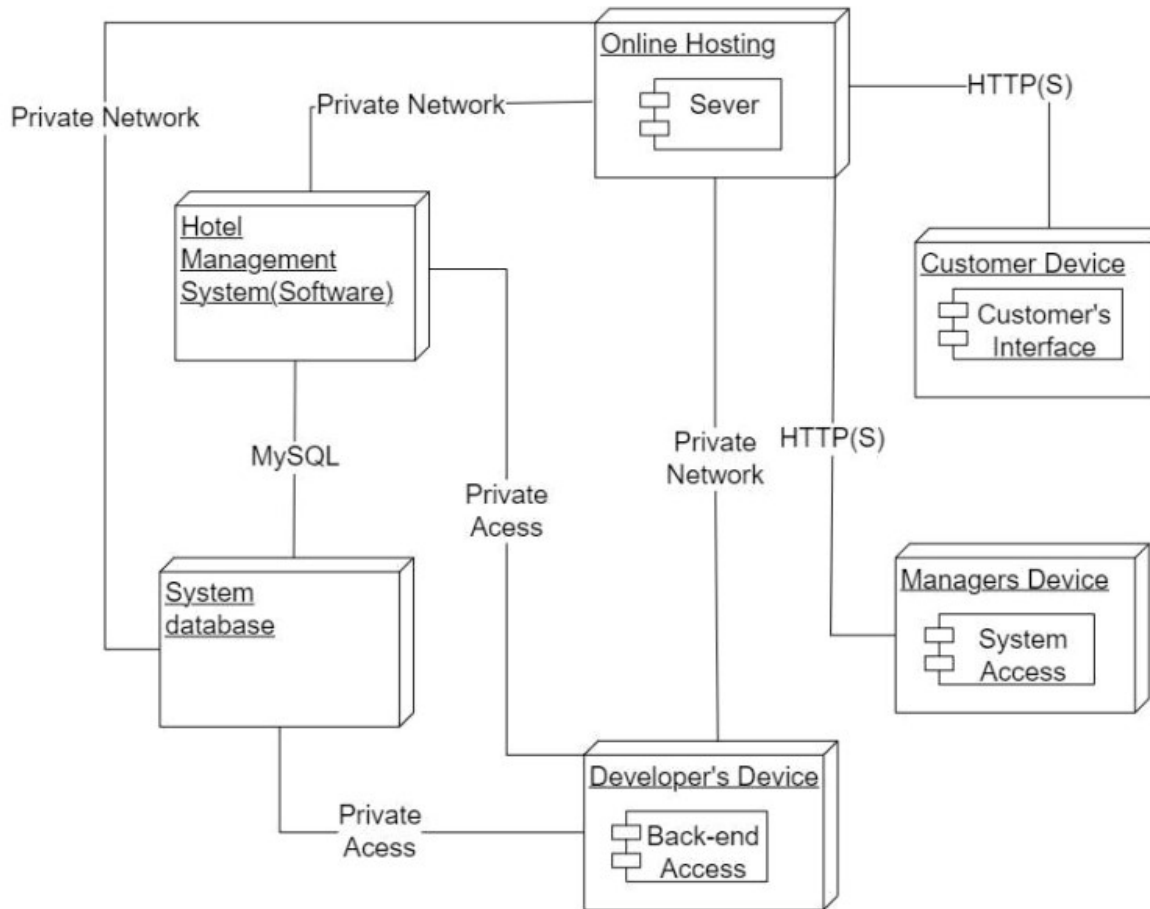
This is a Component diagram of Hotel Management System which shows components, provided and required interfaces, ports, and relationships between the Rooms, Hotel Facility and Bill.



This type of diagrams is used in Component-Based Development (CBD) to describe systems with Service-Oriented Architecture (SOA). Hostel Management System UML component diagram, describes the organization and wiring of the physical components in a system.

DEPLOYMENT DIAGRAM

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware.



It maps software pieces of a system to the device that are going to execute it. Deployment diagrams can be used for, Modeling the network topology of a system. Modeling distributed systems and networks. Forward and reverse engineering processes.

CONCLUSION

As a whole, the UML Diagrams works together to achieve the most desired functions of a Hotel Management Project System. This helped us to design a basic Blueprint of a Working management Model. UML also allows us to easily Program the Software by simplifying the components. By completing all the given Diagrams, the Hotel Management Project System development would be much easier and attainable.

REFERENCES

The Following References are: -

- SRM Lecture PPTs
- Java point
- Lucid Chart
- Etc