



# **OBJECT-ORIENTED** **PROGRAMMING AND DESIGN**

## **PROJECT REPORT:** **Hotel Management System**

**By:**

**RA2111030010078 - Pranav**

**RA2111030010064 – Keshav**

**T-1 Section**

# INDEX

S.No.	Content	Page No.
1	Problem Description	2
2	Class Diagram	3
3	Use Case Diagram	4
4	Sequence Diagram	5
5	Communication Diagram	6
6	State Chart Diagram	7
7	Activity Diagram	8
8	Component Diagram	9
9	Package Diagram	10
10	Deployment Diagram	11
11	Conclusion	12
12	References	13

## **PROBLEM DESCRIPTION:**

The Hotel Management System is used for maintaining the information for each and every customer and to ensure a smooth stay for them. Each and every customer has their own personal details and the services are transferred through the hotel staff.

Before providing the booking and services to the customer, the administrator/receptionist first checks room availability and gathers necessary information about the customer. Meanwhile the invoice is also raised for the customer. The customer also has the liberty to cancel or alter their room booking and the hotel receptionist can do the necessary changes if the circumstances allow for it.

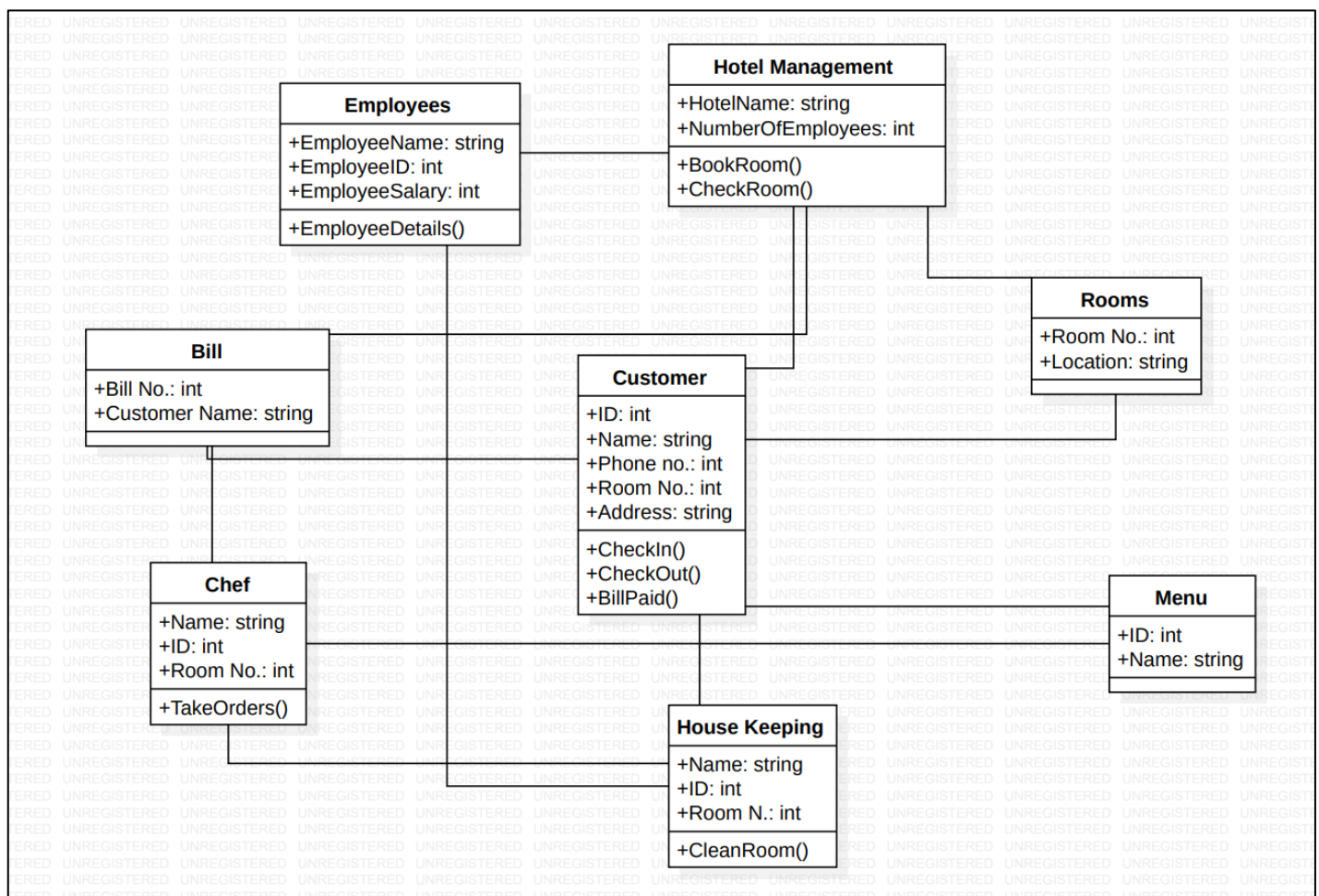
After checking into the room(s), the customer can also avail the various facilities of the hotel through the system such as room cleaning via housekeeping, food services based on the menu prepared by the hotel Chef and staff and so on. From the start of the room booking right up until check out, the invoice of the customers stay is updated and the final bill to be paid is presented at check out to the customer.

The goal of the hotel management system is to offer a system for running a hotel that has a vast number of rooms. The management of the hotel has grown cumbersome without automation. This system will significantly simplify the day-to-day tasks of maintaining a hotel. It will be able to manage numerous services to quickly take care of all consumers. The system should be user-friendly, simple to use, easy to recover from errors, and have a high level of overall end-user subjective satisfaction.

# 1. CLASS DIAGRAM:

The objective of a UML use case diagram is to show the interactions of numerous items called actors with the use case and to capture fundamental functionalities of the E Commerce website.

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. Here, the Class Diagram shows the various objects and functions linked in the due process of a customer checking in until they check out after their stay.

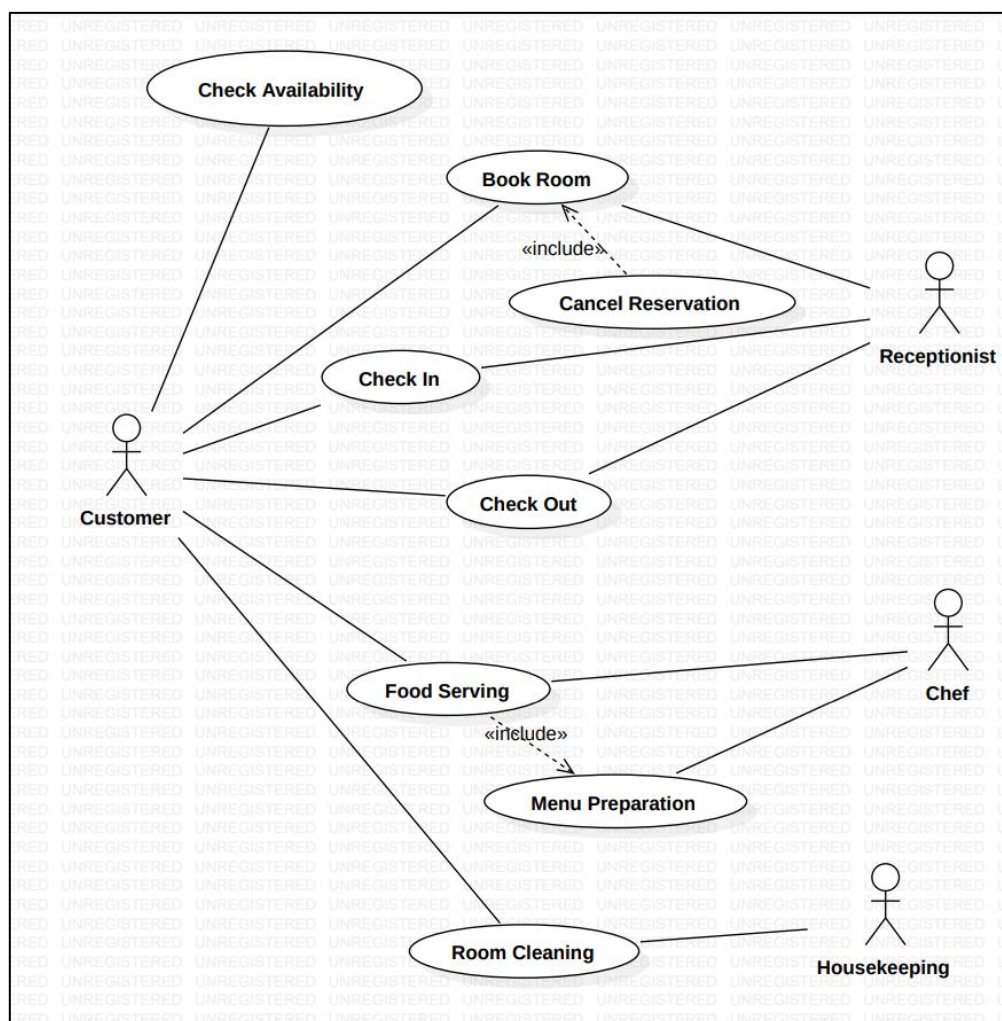


## 2. USE CASE DIAGRAM:

The objective of a UML use case diagram is to show the interactions of numerous items called actors with the use case and to capture fundamental functionalities of Hotel Management.

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

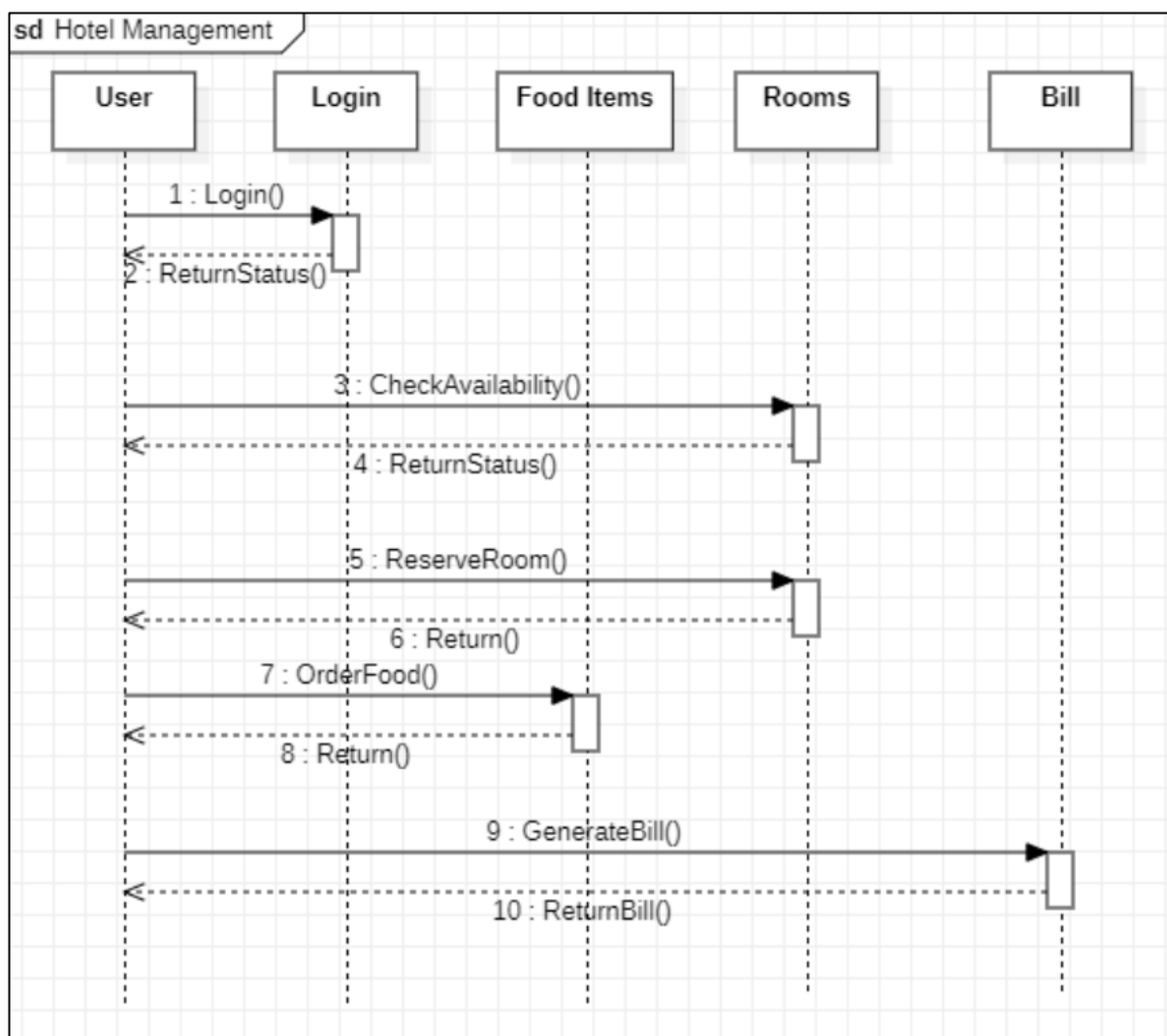
The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.



### 3. SEQUENCE DIAGRAM:

Sequence diagrams object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out Hotel Management and the Hotel booking process.

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.

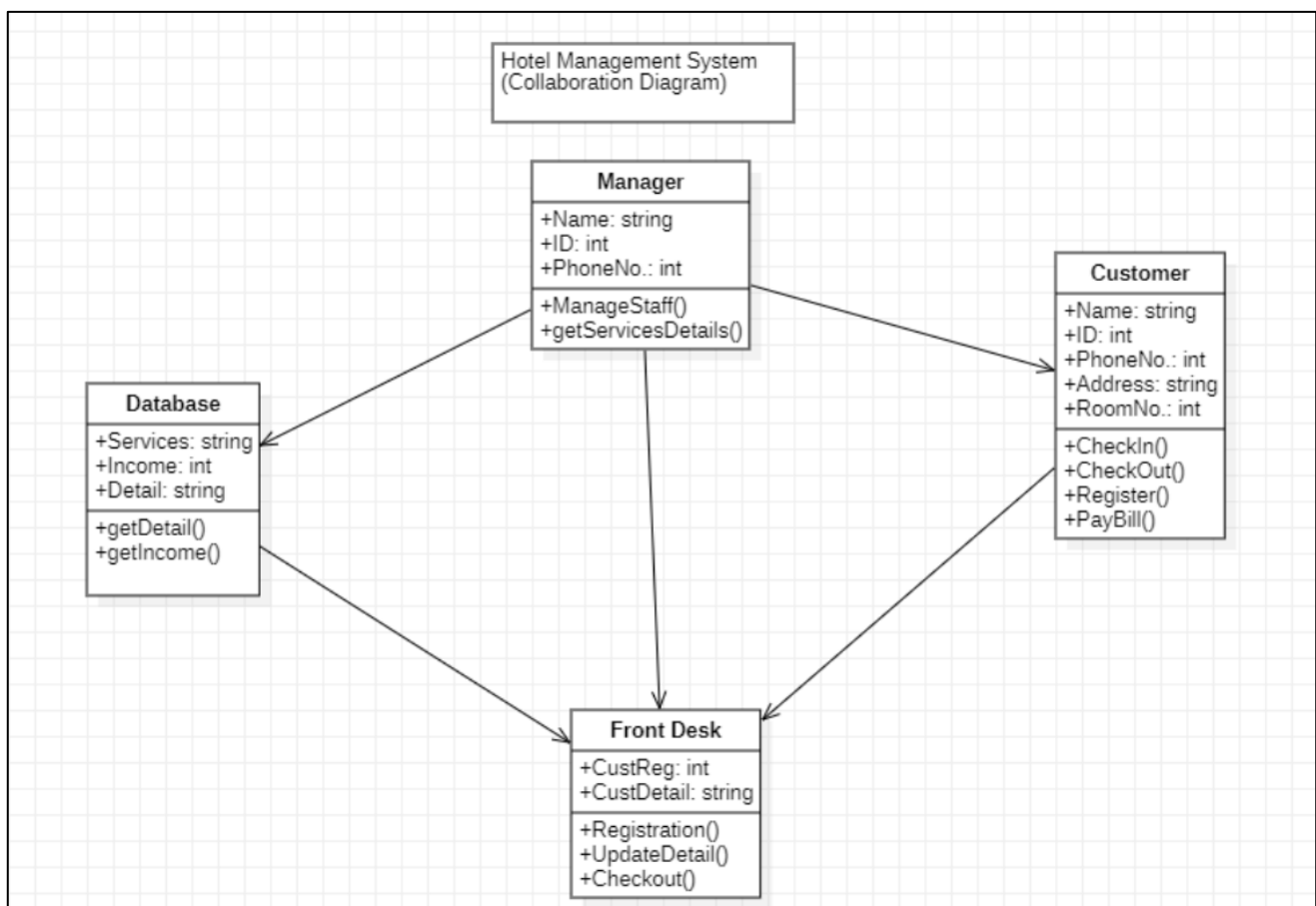


## 4. COMMUNICATION DIAGRAM:

A communication diagram is an extension of object diagram that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, communication diagram shows the messages the objects send each other.

Communication diagrams, formerly known as collaboration diagrams, are almost identical to sequence diagrams in UML, but they focus more on the relationships of objects—how they associate and connect through messages in a sequence rather than interactions. With our UML diagramming tool, refer to this guide on everything you need to know when planning and creating a communication diagram in UML.

UML communication diagrams, like the sequence diagrams - a kind of interaction diagram, shows how objects interact. A communication diagram is an extension of object diagram that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, communication diagram shows the messages the objects send each other.

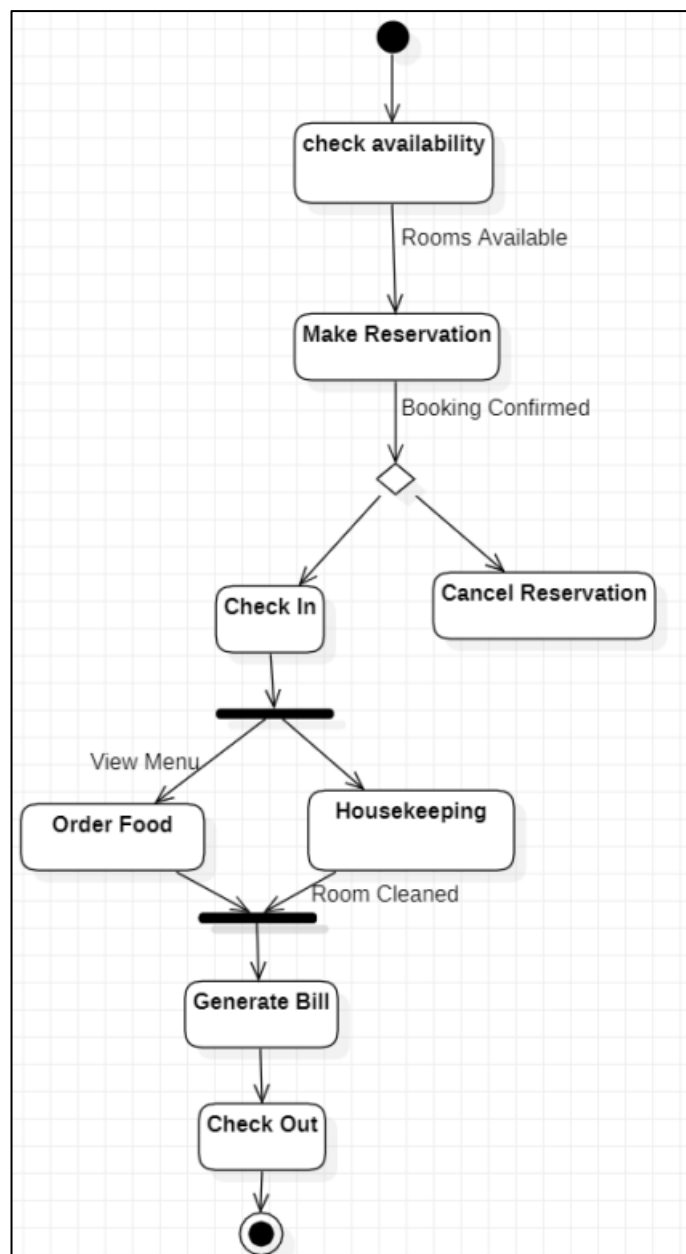


## 5. STATE CHART DIAGRAM:

State transition diagrams provide a way to model the various states in which an object can exist. While the class diagram shows a static picture of the classes and their relationships, state transition diagrams model the dynamic behaviour of a system in response to external events (stimuli).

Statechart diagrams describe a state machine. State machine can be a machine which defines different states of an object and these states are controlled by external or internal events.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of this diagram is to model lifetime of object from creation to termination.

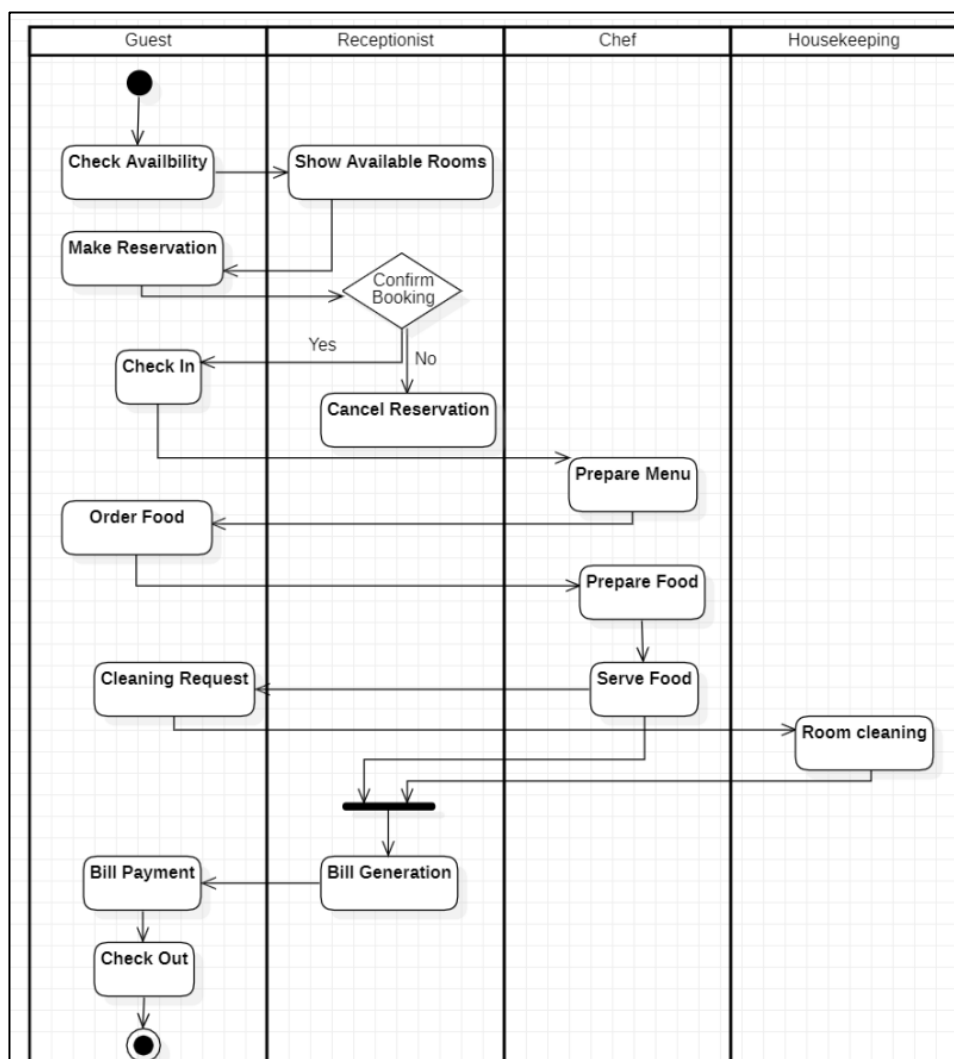




## 6. ACTIVITY DIAGRAM:

Activity diagrams describe the activities of a class. They are similar to state transition diagrams and use similar conventions, but activity diagrams describe the behaviour/states of a class in response to internal processing rather than external events. Activity diagram is basically a flowchart to represent the flow from one activity to another. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Here we use the Swimlane form of Activity Diagrams.

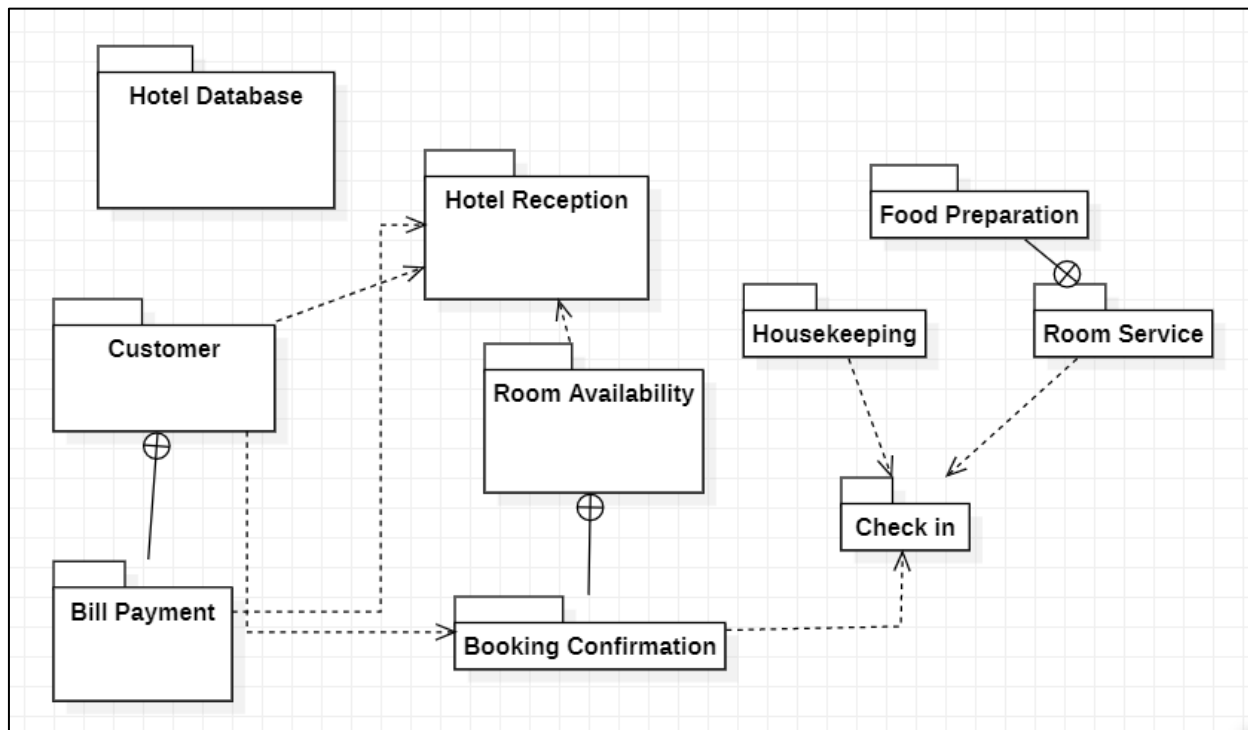
Swimlanes are used to show which activities are done by which organisation in the diagram. The lanes are boundaries and the activities of a particular organisation are drawn in the same lane as that of the organisation. They're good in that they combine the activity diagram's depiction of logic with the interaction diagram's depiction of responsibility. Here, it clearly shows the relation & chain of order between the customer and various functionalities in the Hotel.



## 7. PACKAGE DIAGRAM:

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages. A package is a grouping of related UML elements, such as diagrams, documents, classes, or even other packages. Each element is nested within the package, which is depicted as a file folder within the diagram, then arranged hierarchically within the diagram. Package diagrams are most commonly used to provide a visual organization of the layered architecture within any UML classifier, such as a software system.

Package diagram, a kind of structural diagram, shows the arrangement and organization of model elements in middle to large scale project. Package diagram can show both structure and dependencies between sub-systems or modules, showing different views of a system, for example, as multi-layered (aka multi-tiered) application - multi-layered application model. Below, We see the dependency and structure of the Hotel Management Database; From the Customer to Room Service to the final Bill payment.

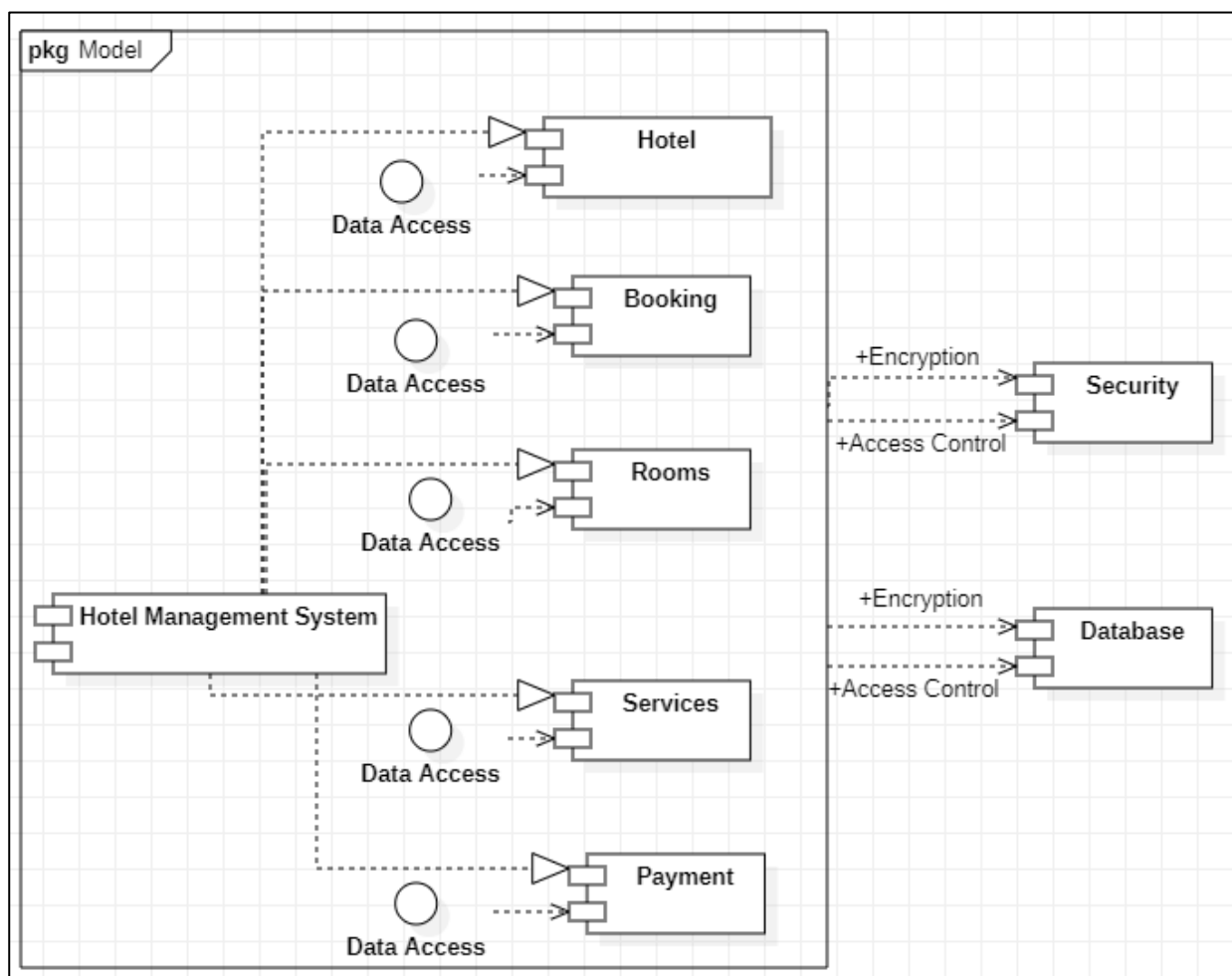


## 8. COMPONENT DIAGRAM:

A component diagram in the Unified Modelling Language depicts how components are wired together to form larger components and or software systems. Components diagrams can be used to illustrate the structure of arbitrarily complex systems.

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behaviour is explained by the provided and required interfaces.



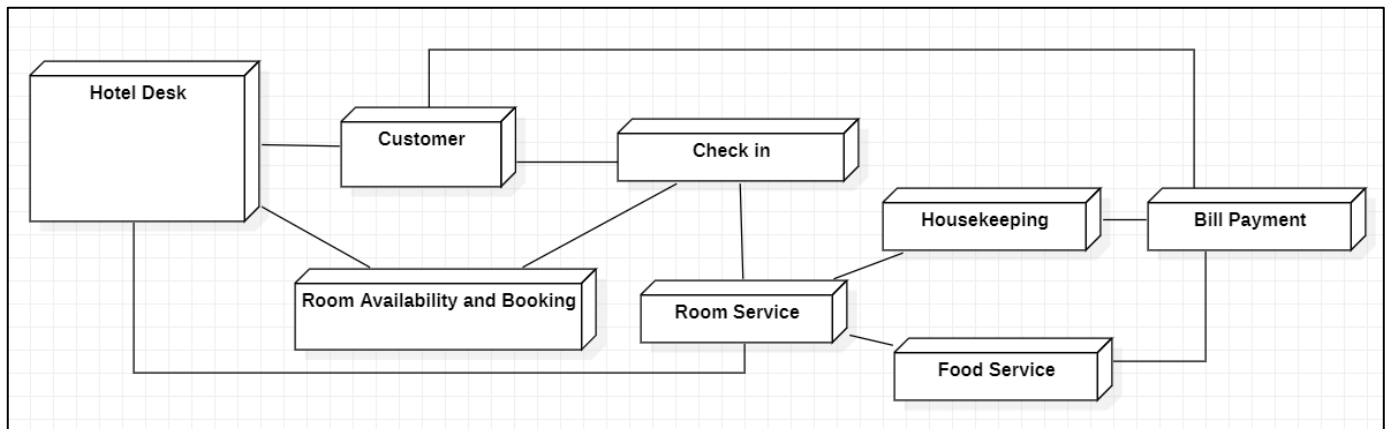
## 9. DEPLOYMENT DIAGRAM:

A deployment diagram in the Unified Modelling Language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show the allocation of Artifacts to Nodes according to the Deployments defined between them. Deployment of an artifact to a node is indicated by placing the artifact inside the node.

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system. The diagram below shows how the hotel management system will be deployed during physical implementation.



## **CONCLUSION:**

Hotel Management is a managing structure or system that allows customers to book hotel rooms based on availability. They also have the flexibility to cancel their booking or change it as per their desire. The customer can also avail the various services offered in the hotel, such as housekeeping and food services. The main purpose of Hotel Management is hospitability and making each customer's stay at the hotel feel like a home away from home.

With this system, Hotel Management is made significantly easier as it reduces manual labour and thereby also reduces the risk of human error. Thereby, ensuring a smooth stay for the customer and success in the running of the Hotel

## **REFERENCES:**

<https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>