Name	**Vyom Doshi (2020300009)** **Shubham Golwal (2020300015)**
Class:	**T.E. C.O.M.P.S.**
Experiment No.	2

AIM:	Design UML diagram -Use Case & Description

ASSIGNMENT 2

Problem Statement:

Resort Property Management System is designed to manage managers with the ability to control their company's rental business. All aspects of short-term rental management are considered including reservations, accounting, maintenance, housekeeping, booking, and more.

Theory:

Purpose of Use Case Diagrams:

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

Where to Use a Use Case Diagram?

As we have already discussed there are five diagrams in UML to model the dynamic view of a system. Now each and every model has some specific purpose to use. Actually, these specific purposes are different angles of a running system.

To understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

These diagrams are used at a very high level of design. This high level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering. Use case diagram is used differently to make it suitable for reverse engineering.

In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.

Use case diagrams can be used for −

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.

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.

UML 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.

Where to Use Class Diagrams?

Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system.

Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

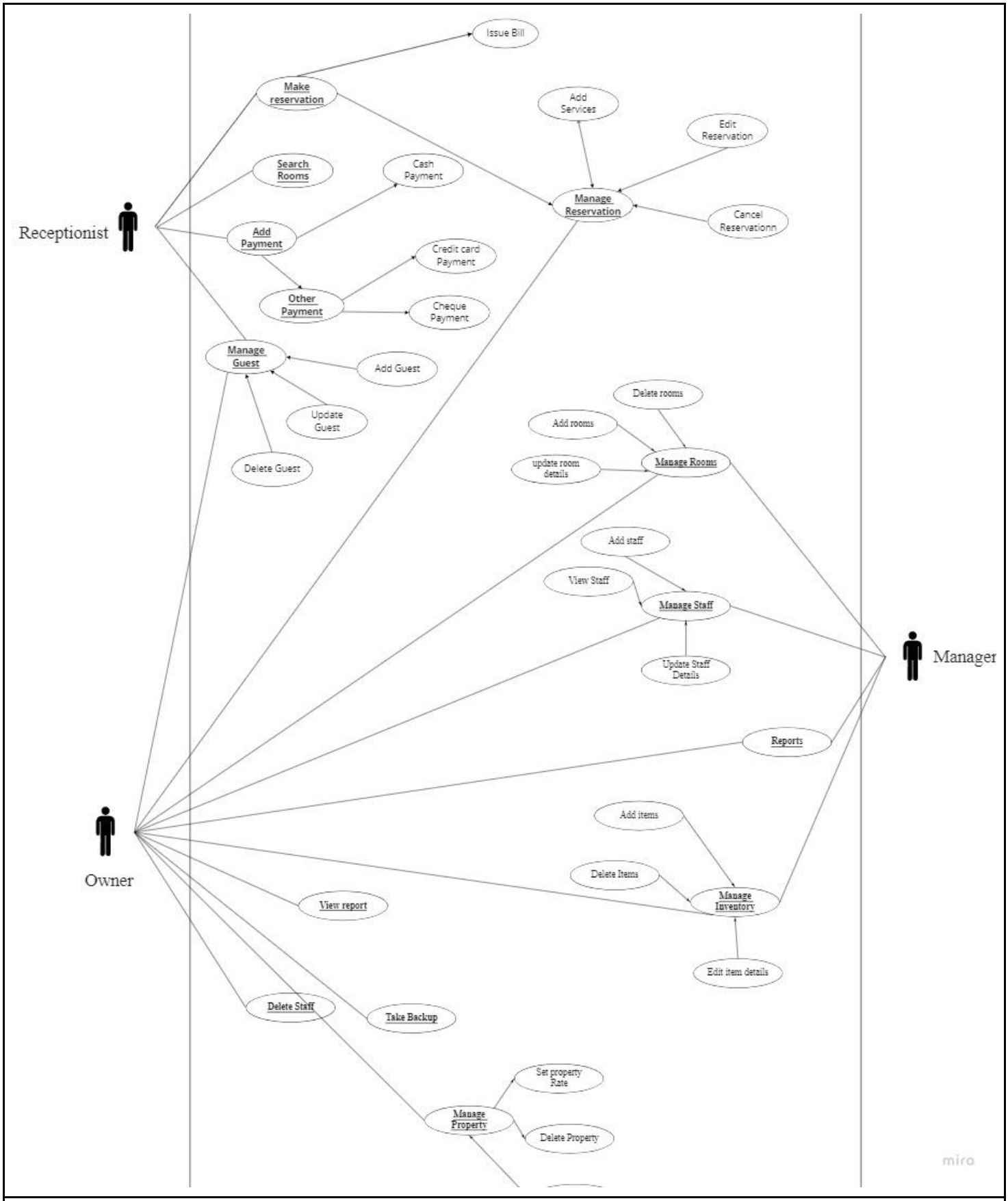
Generally, UML diagrams are not directly mapped with any object-oriented programming languages but the class diagram is an exception.

Class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, class diagram is generally used for construction purpose.

In a nutshell it can be said, class diagrams are used for −

- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionalities performed by the system.
- Construction of software applications using object oriented languages.

Use Case Diagram:



Use Case Description:

Use case	**Make reservation**	
Goal	Add a new reservation	
Description Actor	Receptionist, Manager, Owner	
Precondition	Same guest shouldn't already exist	
Normal Flow	**Steps**	**Actions**

	1.	Receptionists enter the guest details and their room requirement
	2.	System searches for room details
	3.	System presents room types and tariffs
	4.	System records customer's name and address
	5.	Receptionist confirms booking on system
	6.	System generates confirmation receipt
Alternate Flow	3.1 There are no rooms available in that category 3.2 The receptionist then has to change the category until rooms are found and then proceed	
Exception	The database somehow is not able to accept new entries, there is critical system failure	
Postcondition	Hotel guest detail updated to current hotel list	
Special requirement	**N.A.**	

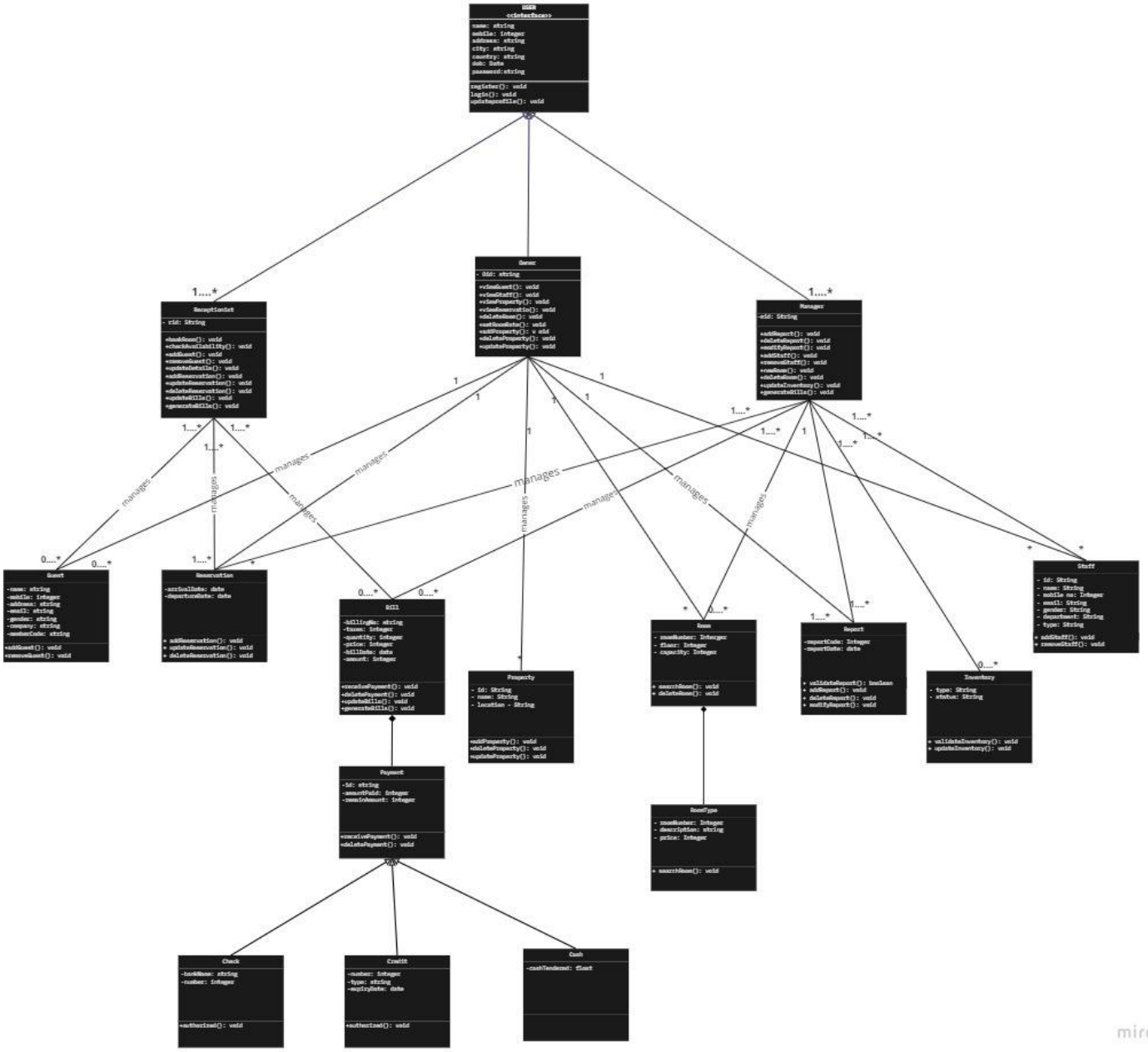
Use case	**Manage Staff (Add, Update, Delete, View)**	
Goal	Create, update delete or view staff members	
Description Actor	Owner	
Precondition	The user/actor has verified himself to be the owner	
Basic Flow	**Steps**	**Actions**
	1.	If the actor wants to create a new profile, then he clicks on the create option. He then enters the details of the new employee
	2.	if he wishes to see the employee profile then he enters the employee id or name
	3.	The system shows the search results from the database
	4.	The actor can select the profile required
	5.	He can click edit/remove shown on the top right
	6.	If edit is clicked the preexisting details are loaded on the screen in editable mode
Alternate Flow	2.1 The employee doesn't exist	
Exception	The database somehow is not able to accept new entries, there is critical system failure, the filter system is not working accurately due to database update problems	
Postcondition	The details of the employee are updated successfully	
Special requirement	**N.A.**	

Use case	**Taking Backups**	
Goal	Take a backup of the system	
Description Actor	Owner	
Precondition	User should login to the system/ the backup trigger should be invoked	
Basic Flow	**Steps**	**Actions**
	1.	Display user interface
	2.	Select backup option or restore option
	3.	Display backup interface
	4.	Select create backup option
	5.	Backup version created and named automatically
	6.	Stop all updating processes on the database

	7.	Creating backup process is not successful, display “Unsuccessful” message.
Alternate Flow	2.1 if the user selects restore option, then it asks for the version of backup to be restored 2.2 after selecting the version number the system automatically restores the system to that state	
Exception	Some processes updating the database still remain on.	
Postcondition	The backed-up data is stored on a remote storage device.	
Special requirement	The storage device should not be touched while backup is being performed	

Use case	**Get financial reports**	
Goal	View a financial report for specific time	
Description Actor	Owner	
Precondition	Log in to the system	
Basic Flow	**Steps**	**Actions**
	1.	Display user interface
	2.	System prompts the owner to select two dates
	3.	Enter the required dates
	4.	System will display the revenue for that specific time
Alternate Flow	2.1 If invalid details entered Display “unsuccessful” message and reenter interface.	
Exception	The user enters one of the dates which is in the future/an invalid date.	
Postcondition	The financial reports are displayed in case of valid details entered.	
Special requirement	**N.A.**	

Class Diagram:



CONCLUSION:	From this experiment we were able to understand the difference Use case diagram and class diagram and we were also able to understand how to create and implement these forms of representation.