

HavenInn

Hotel Management System

Low-Level Design (LLD)

BY
GROUP 2

Avu Naveen Kumar
Padala Ruchitha
Shruti Pravin Satam
Arfin Zainuddin Sayyed
Anisha Rajesh Dubey
Prerana Mahesh Waghela
Janhavi Sunil Gangan
Swati Ramdas More
Shama Ravindra Anpat
Asavari Vijay Akshekar

iTransform || BULT July 2022
Capgemini Services India Pvt Limited

INDEX

SR. NO.	CONTENT	PAGE NO.
CHAPTER 1	INTRODUCTION	4-6
1.1	Introduction	4
1.2	Documentation Purpose	4
1.3	Intended Audience	5
1.4	Purpose, Scope & Applicability	5-6
CHAPTER 2	REQUIREMENTS AND ANALYSIS	7-8
2.1	Problem Definition	7
2.2	Planning and Scheduling	7
2.3	Hardware Requirements	7
2.4	Software Requirements	8
CHAPTER 3	SYSTEM DESIGN & UML DIAGRAMS	9-38

3.1	Use Case Diagram	9-10
3.2	Class Diagram	11-12
3.3	Database Diagram	13
3.4	ER Diagram	14-15
3.5	Activity Diagram	16-25
3.6	Sequence Diagram	26-32
CHAPTER 4	IMPLEMENTATION AND TESTING	
4.1	HTTP Codes	
4.2	Database Design	
4.3	Classes and Functions	
4.4	Unit Testing	
CHAPTER 5	RESULT AND DISCUSSION	
5.1	Screen Layouts and Visual Implementation	
CHAPTER 6	CONCLUSION AND REFERENCE	

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

HavenInn is an online Hotel Management System that automates the major operations of the hotel. The Reservation Module keeps track of room reservations and checks availability. The Room Module is for managing all room types and room services. The Inventory Control Module will keep track of all inventories of the hotel and guest details will be handled by Guest Module. There are three End Users for HavenInn. The End Users are the Owner, Manager, and Receptionist. The owner can access all system functionalities without any restrictions. The Manager can access all system functionalities with limited restrictions. The Receptionist can only access the Reservation Module. To keep restrictions for each End User level, HavenInn also provides Role Level Authorization.

1.2 DOCUMENTATION PURPOSE

This document describes the solution architecture for HavenInn - Online Hotel Management System. This will provide a detailed description of the requirements for the Hotel Management System (HavenInn). This will allow for a complete understanding of what is to be expected from the newly introduced system. A clear understanding of the system and its functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project.

The document would finally provide a clear idea about the system. A brief outline of the document is

1. Overall Description
2. System Features
3. External Interface Requirements
4. Non-Functional Requirements

1.3 INTENDED AUDIENCE

The intended audience of this document would be the owner and specific employees like the Manager and Receptionist of the Hotel, and the Project Team with the objective to refer and analyze the information. This document can be used in any case regarding the requirements of the project and the solutions that have been taken.

Role	Intended Audience
Stake Holders	Owner, Manager, and Receptionist
Developers	Group 2

1.4 PURPOSE, SCOPE AND APPLICABILITY

1.4.1 PURPOSE

The purpose of the Hotel Management System is to simplify the day-to-day processes of the hotel. The system will be able to handle multiple tasks in an efficient manner. As a solution to a large amount of file handling happening at the hotel, this software will be used to overcome those drawbacks. Ease of use and most importantly the efficiency of information retrieval are some benefits of the system. The system will be user-friendly, provide easy recovery of errors, and have an overall end-user high subjective satisfaction.

1.4.2 SCOPE AND APPLICABILITY

1.4.2.1 Proposed System Features

The system will be able to provide a number of services to clients in a fast and efficient manner.

There are three user levels in the Hotel Management System:

- ★ Owner
- ★ Manager
- ★ Receptionist

Owner: Hotel owner has the privilege of Monitoring and authorization of all the tasks handled by the system. He can access every function performed by the system. As the main authorized person of the hotel, the owner gets the ability to manage the other users including their user roles and privileges. Meanwhile, he will be able to take all the kinds of reports available in the system. As the owner of the system and the hotel, he has the power to set room rates as well.

Manager: The manager also has most of the privileges mentioned above, except the things regarding payment handling. The reason for using a manager is to reduce the workload done by the owner that cannot be assigned to the receptionist, as those tasks seem much more

responsible. At the user level, the Manager has the authority to take all the reports available in the system but here also except the reports related to financial stuff and hotel income.

The manager has other abilities that the receptionist doesn't have. Such as, adding a new staff member to the system, modifying them or removing them, adding new inventory to the system, modifying them, and removing them. Adding new room types to the system, modifying them, and removing them.

Receptionist: As a hotel receptionist, his or her role will be to attain the goals of bookings and to ensure that all guests are treated with a high standard of customer service. Hierarchically receptionist role has the least accessibility to the system functions. He or she can perform limited functions such as registering new guests to the system, making reservations, and sending email reminders to clients for booking confirmation along with Bill and Payment Links.

CHAPTER 2: REQUIREMENTS AND ANALYSIS

2.1 PROBLEM DEFINITION

To manage all kinds of hotel operations, hoteliers used to manage data either by using Excel sheets or by maintaining files. The processes of manually recording reservations and assigning rooms to guests were all done on paper without tracking the status of the rooms, such as how many rooms were reserved, and how many rooms were available. In response to this practice, hoteliers had to shift through numerous documents to find a single piece of information. Hence, leads to issues like overbooking or double-booking followed by guest dissatisfaction.

As a solution to a large amount of file handling happening at the hotel, Hotel Management System will be used to overcome those drawbacks.

2.2 PLANNING AND SCHEDULING

A schedule is your project's timetable, consisting of sequenced activities and milestones that have to be delivered under a given deadline. Having a project plan you know exactly what should be delivered in what order. Resource allocation helps you find and assign the right employees. Then, a schedule tells you exactly when all of that should happen. With the right scheduling techniques, you can also adjust some activities and tasks in case of a project runs late or if any changes to the scope occur.

2.3 HARDWARE REQUIREMENTS

The stakeholders can use HavenInn both on PC/ Laptops. As it is a web-based application, a user requires an active internet connection.

Component	Minimum	Recommended
Processor	1.9 gigahertz (GHz) x86- or x64-bit dual-core processor with SSE2 instruction set	3.3 gigahertz (GHz) or faster 64-bit dual-core processor with SSE2 instruction set

Memory	8-GB RAM (Developer) 4-GB RAM (Stakeholder)	16-GB RAM or more (Developer) 8-GB RAM (Stakeholder)
Display	Super VGA with a resolution of 1024 x 768	Super VGA with a resolution of 1024 x 768

2.4 SOFTWARE REQUIREMENTS

Browser

The stakeholders should have a browser that supports JavaScript.

The Browsers can be:

- **Microsoft Edge** (latest publicly-released version) running on Windows 10, Windows 8.1, Windows 8, Windows 7
- **Google Chrome** (latest publicly-released version) running on Windows 10, Windows 8.1, Windows 8, Windows 7.
- **Apple Safari** (latest publicly-released version) running on the two latest publicly-released Mac OS versions, or Apple iPad.

For Developers:-

Database: Microsoft SQL Server Management Studio 18.

IDE: Visual Studio 2022 and Visual Studio Code.

Language Used

- C#
- HTML
- CSS
- Javascript/ES6
- React JS

Version Control System: Github

Framework: Microsoft .NET Core Framework 3.1 or above

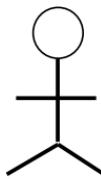
CHAPTER 3: SYSTEM DIAGRAM & UML DIAGRAMS

UML DIAGRAMS

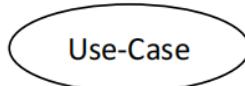
3.1 USE CASE DIAGRAM

- A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

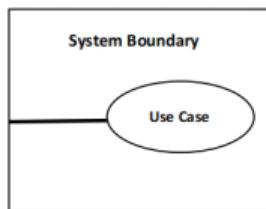
❖ **Actors:** An actor portrays any entity (or entities) that perform certain roles in a given system.



❖ **Use-Case:** A use-case in use case diagram is a visual representation of distinct business functionality in a system.



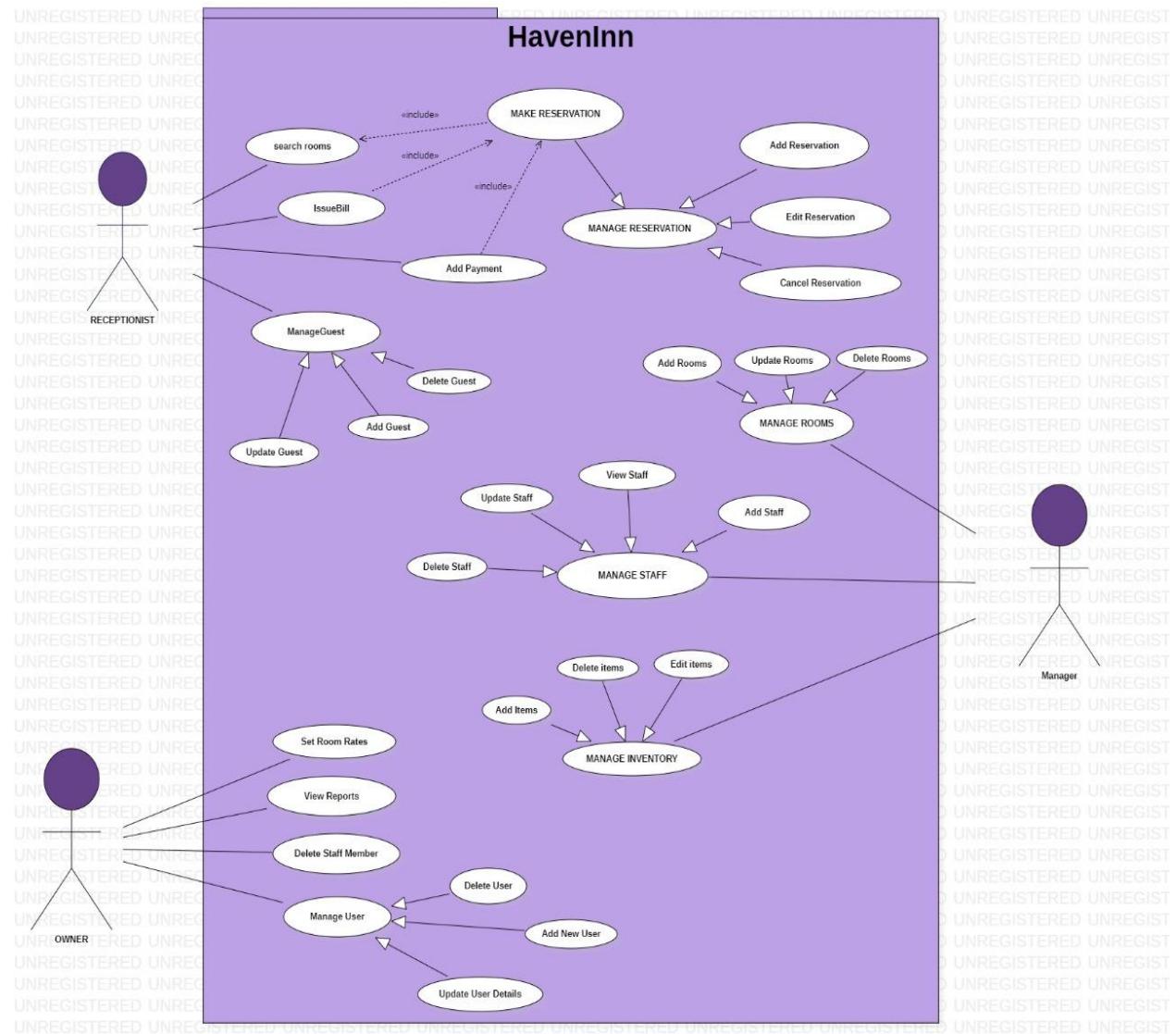
❖ **System Boundary:** A system boundary defines the scope of what a system will be.



❖ **Include:** When a use case is depicted as using the functionality of another use case in a diagram, this relationship between the use cases is called an *include relationship*.

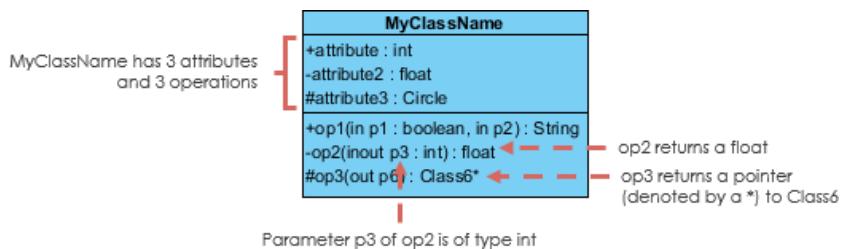


USE CASE DIAGRAM

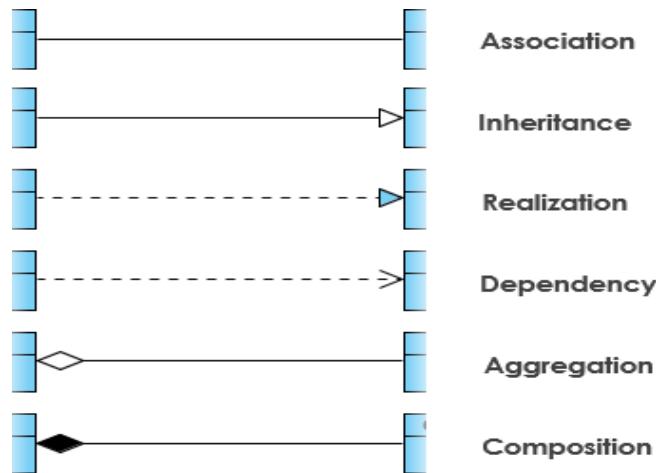


3.2 CLASS DIAGRAM

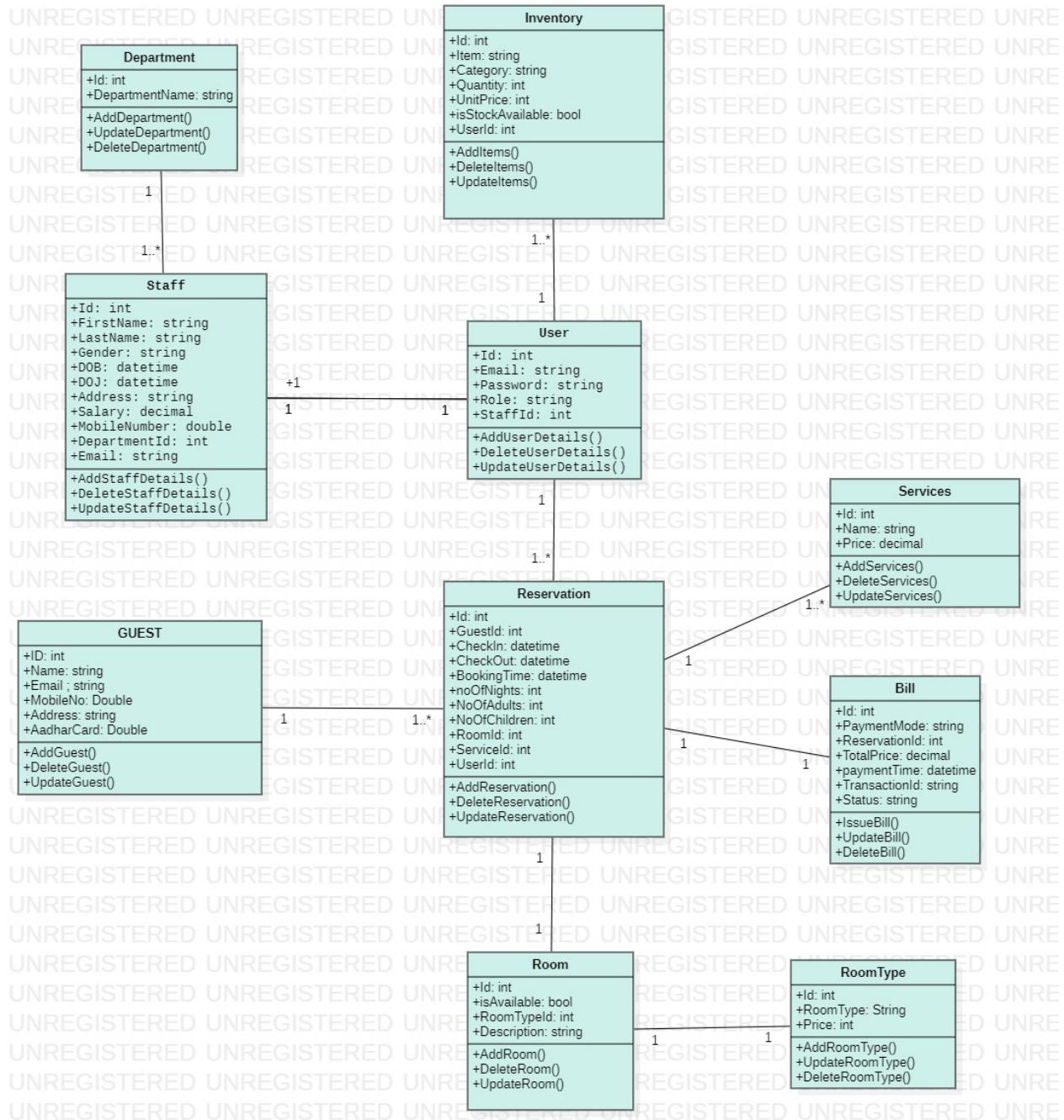
- **Class Name** The name of the class appears in the first partition.
- **Class Attributes:** Attributes are shown in the second partition. The attribute type is shown after the colon.
- **Class Operations (Methods):** Operations are shown in the third partition. They are services the class provides.
- **Class Visibility:** The +, -, and # symbols before an attribute and operation name in a class denote the visibility of the attribute and operation.



- **Relationships between classes:** A class may be involved in one or more relationships with other classes. A relationship can be one of the following types:

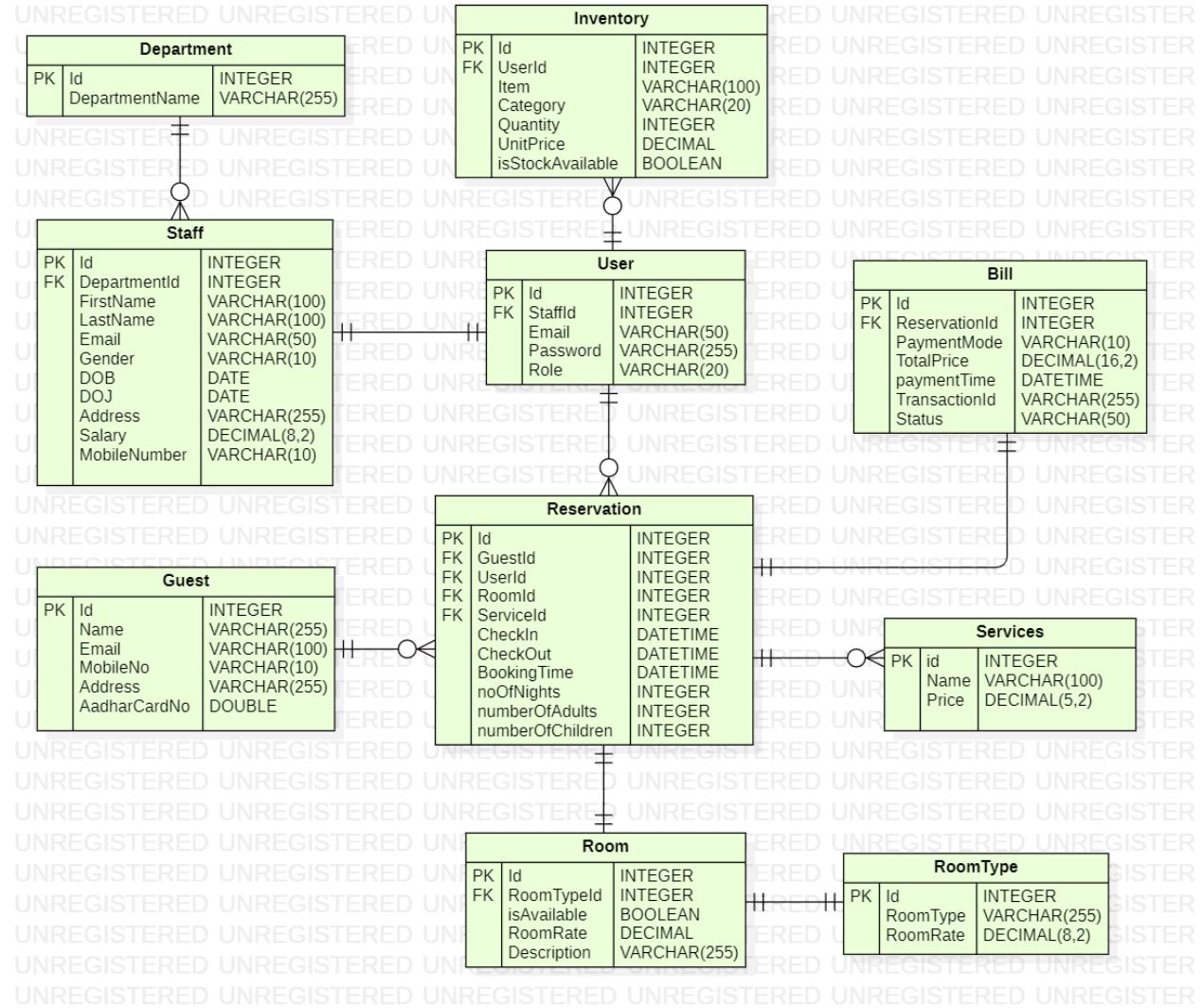


CLASS DIAGRAM



3.3 DATABASE DIAGRAM

Database diagrams graphically show the structure of the database and relations between database objects.

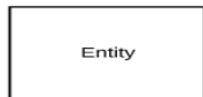


3.4 ENTITY RELATIONSHIP DIAGRAM

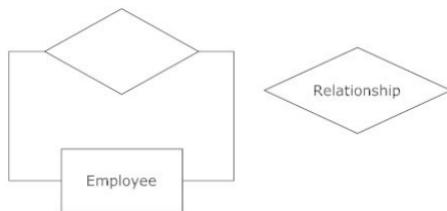
This model is used to define the data elements and relationships for a specified system. ERD displays the relationship of entity sets stored in a database.

Entity Relationship Diagram Symbols

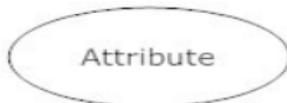
- 1. Entity:** An Entity is an object of Entity Type and a set of all entities is called an entity set. An entity is represented by a rectangle.



- 2. Actions:** Actions that are represented by diamond shapes, show how two entities share information in the database.



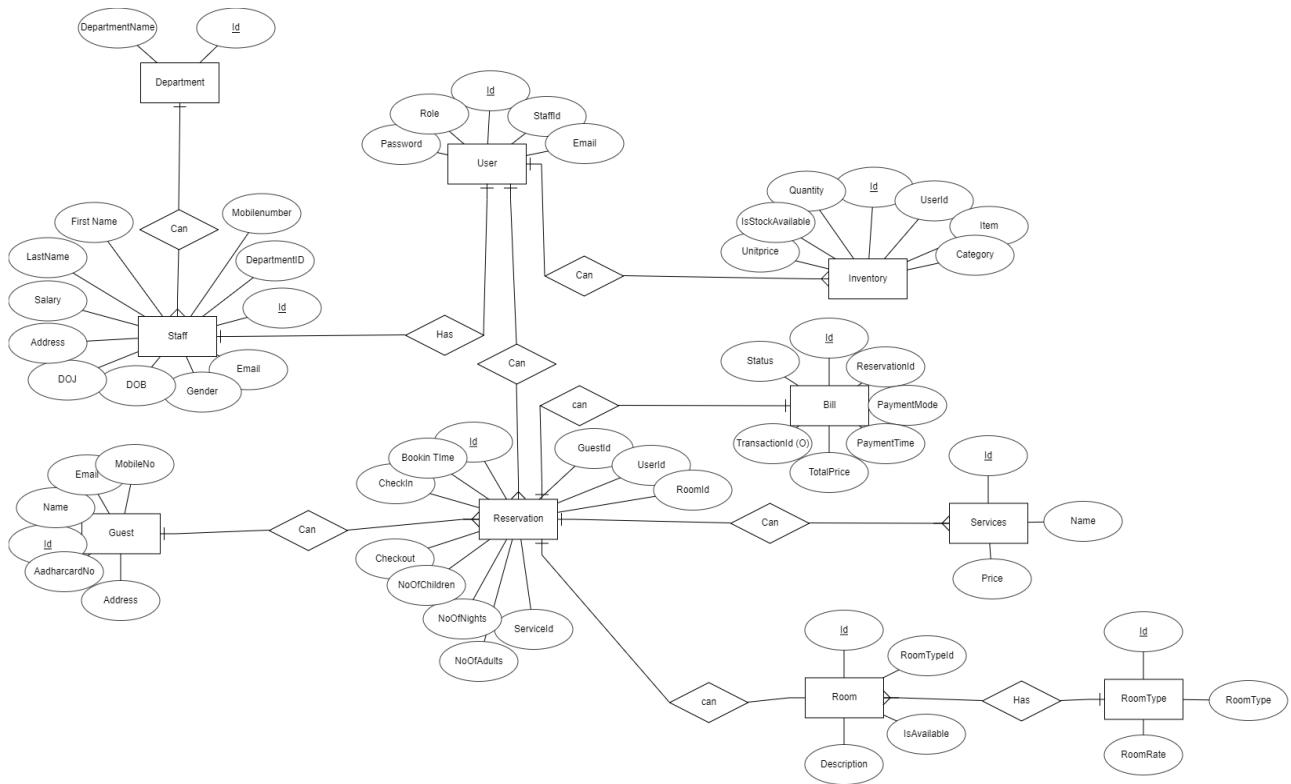
- 3. Attributes:** Attributes are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.



- 4. Connecting lines:** Solid lines that connect attributes to show the relationships of entities in the diagrams.

- 5. Cardinality:** It specifies how many instances of an entity relate to one instance of another entity.

ENTITY RELATIONSHIP DIAGRAM



3.4 ACTIVITY DIAGRAM

Activity Diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency.

- ❖ **Initial State or Start Point:** A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram.



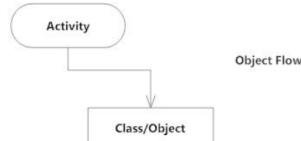
- ❖ **Activity or Activity State:** An action state represents the non-interruptible action of objects.



- ❖ **Action Flow:** Action flows, also called edges and paths are usually drawn with an arrowed line.



- ❖ **Object Flow:** Object flow refers to the creation and modification of objects by activities.



- ❖ **Decisions and Branching:** A diamond represents a decision with alternate paths.

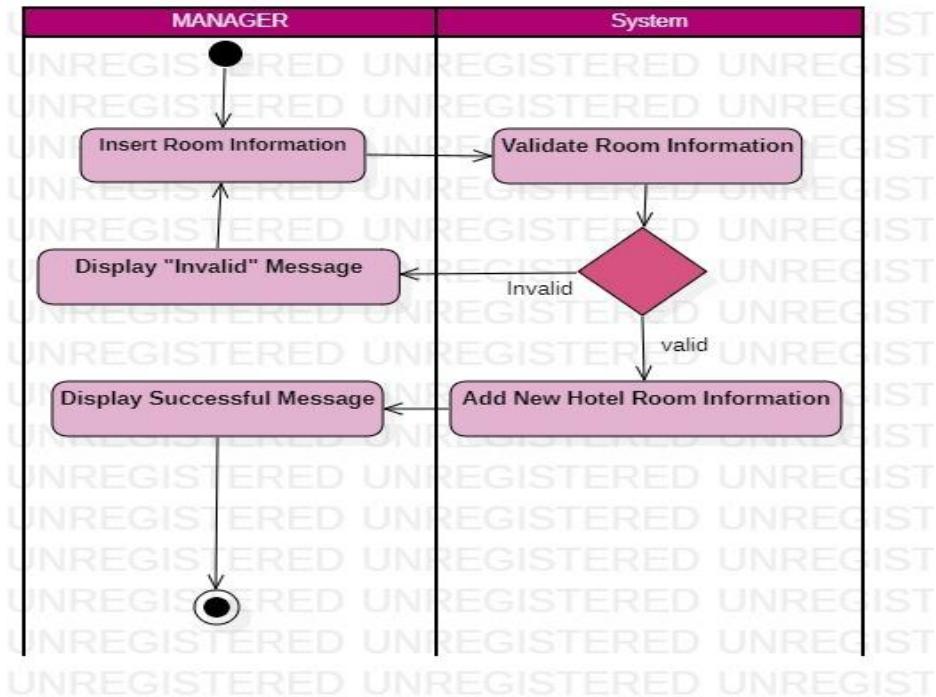


- ❖ **Final State or End Point:** An arrow pointing to a filled circle nested inside another circle represents the final action state.

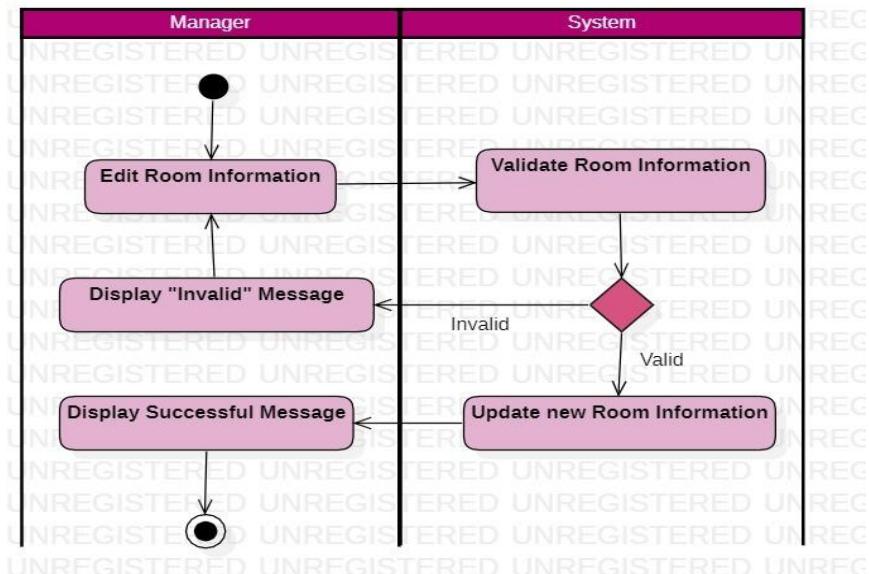


3.4.1 MANAGER

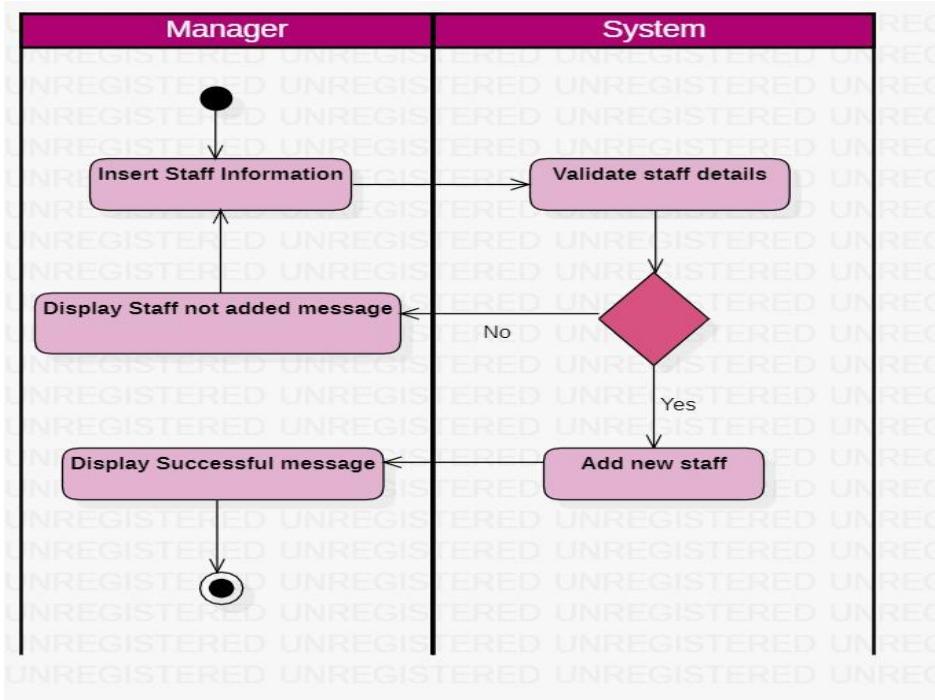
- Add Hotel Room



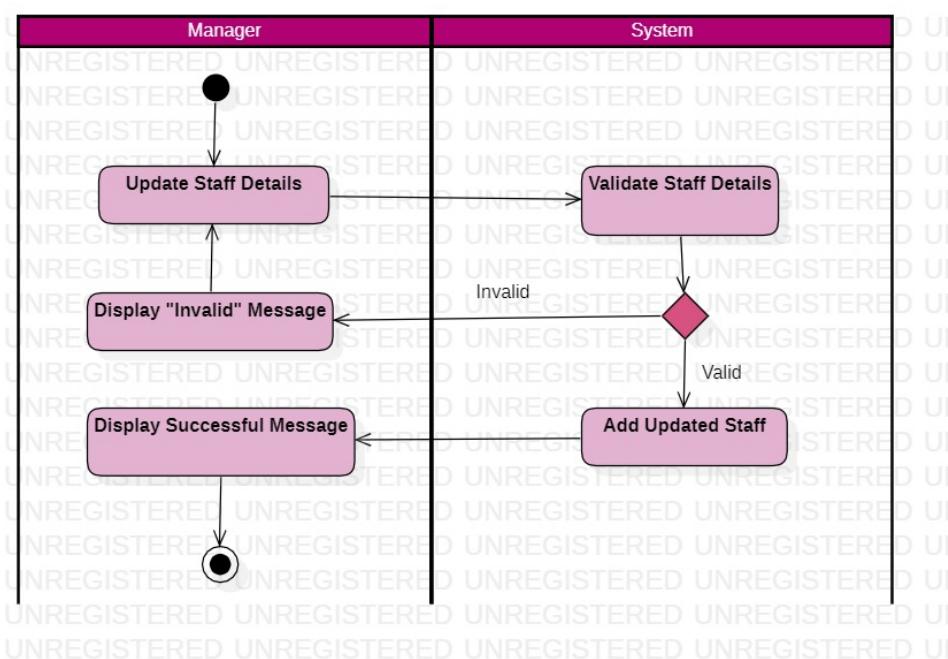
- Edit Hotel Room



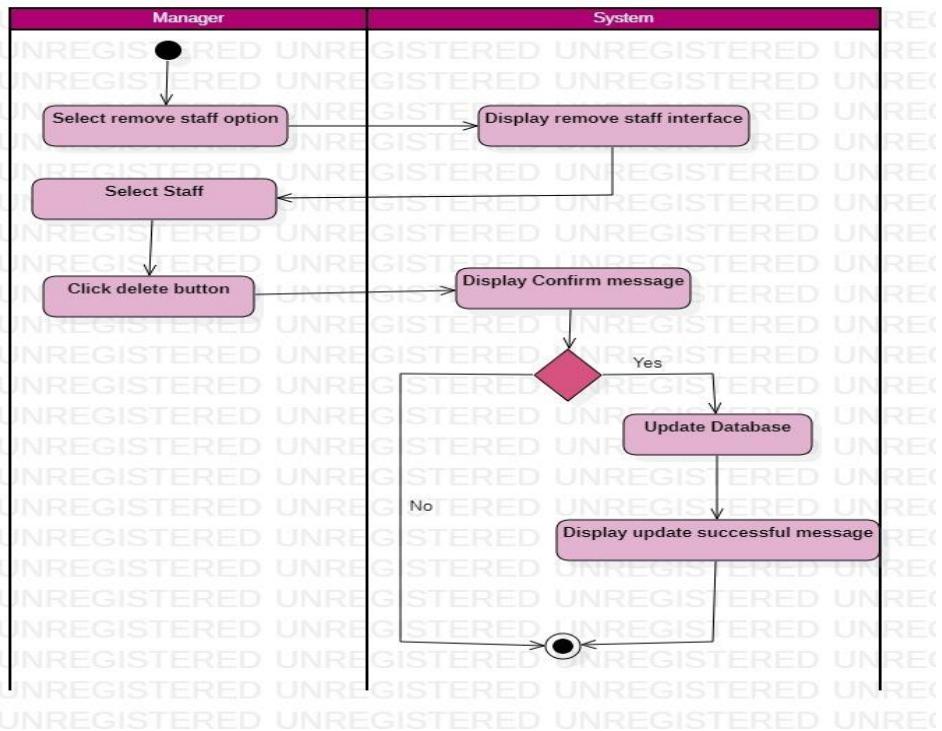
- Add Staff Details



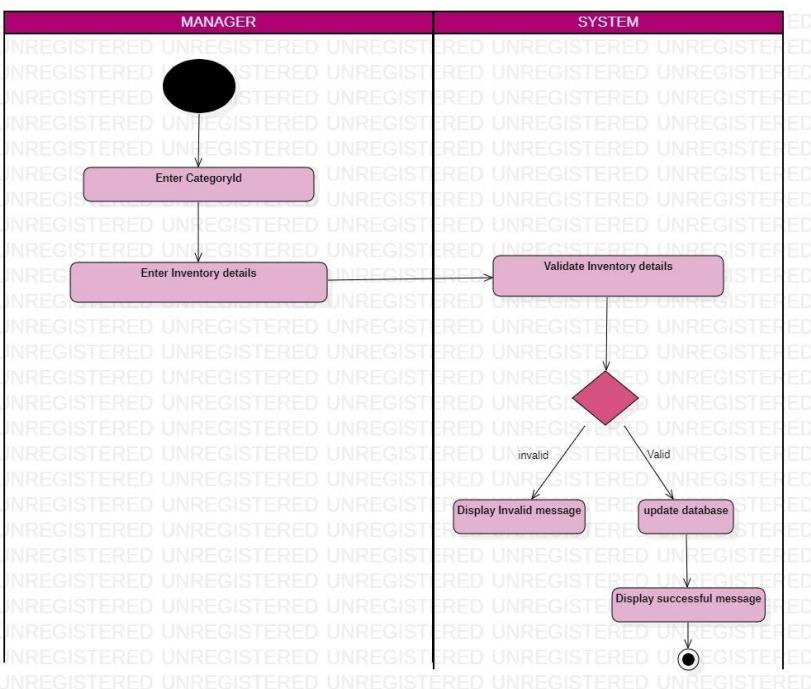
- Update Staff Details



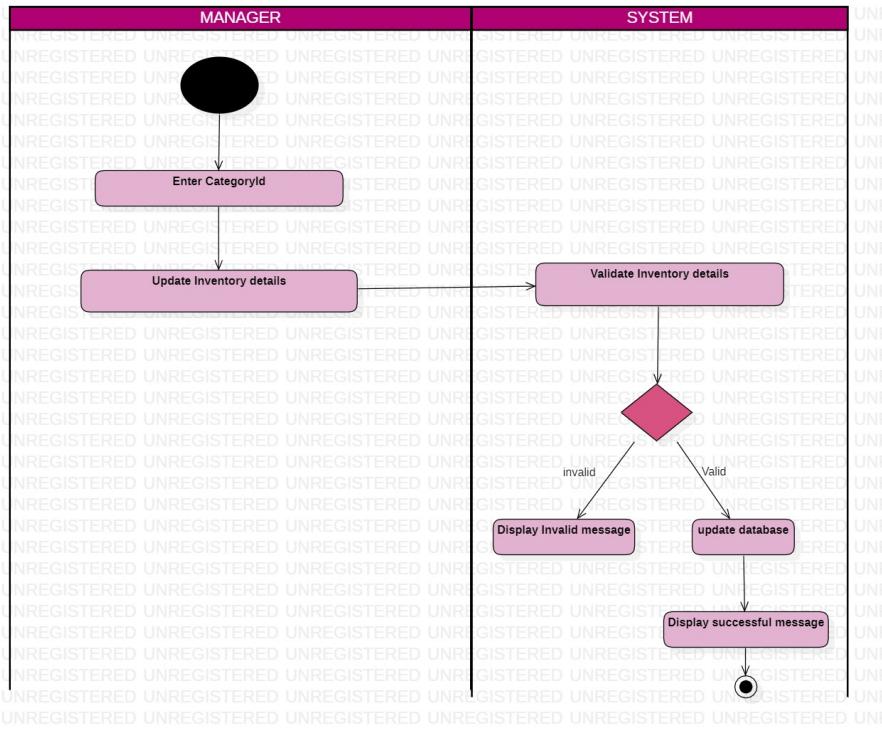
- Delete Staff Details



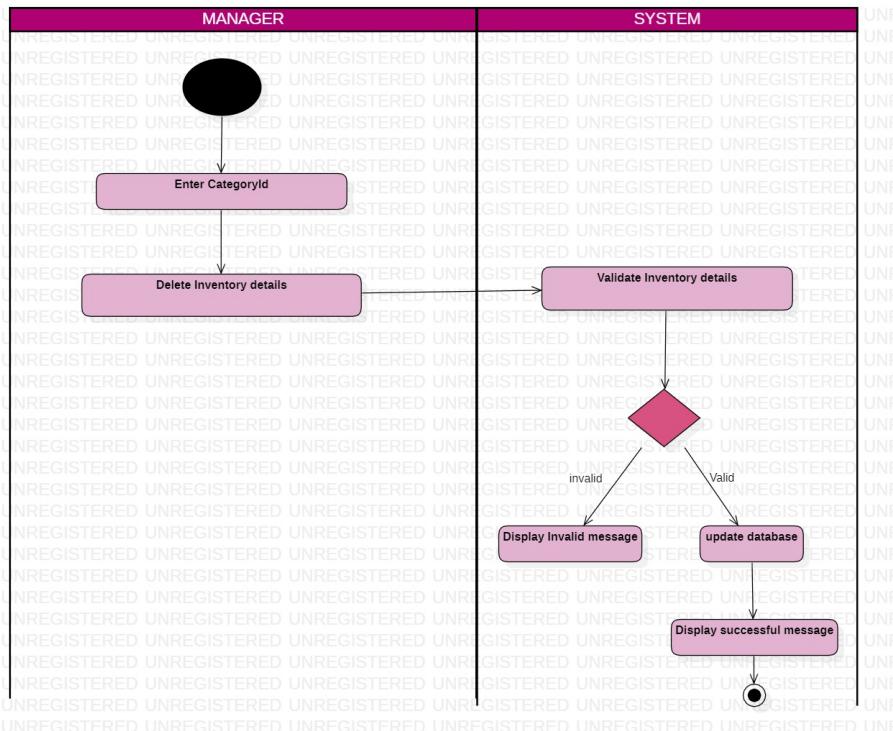
- Add Inventory



● Update Inventory

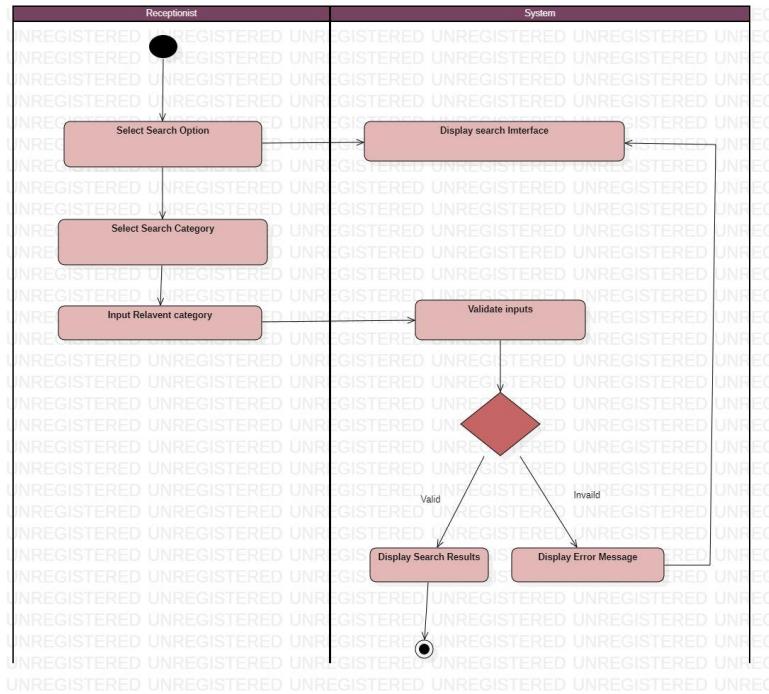


● Delete Inventory

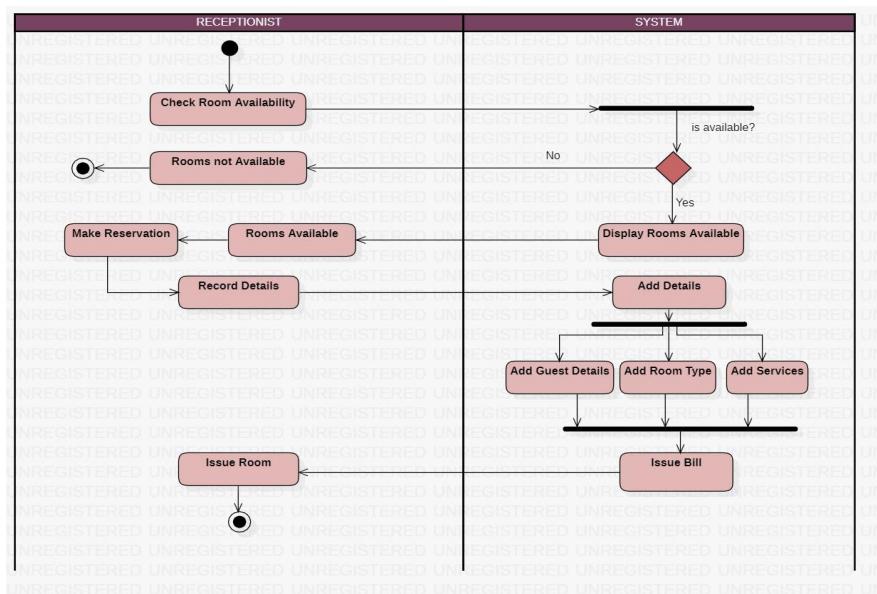


3.4.2 RECEPTIONIST

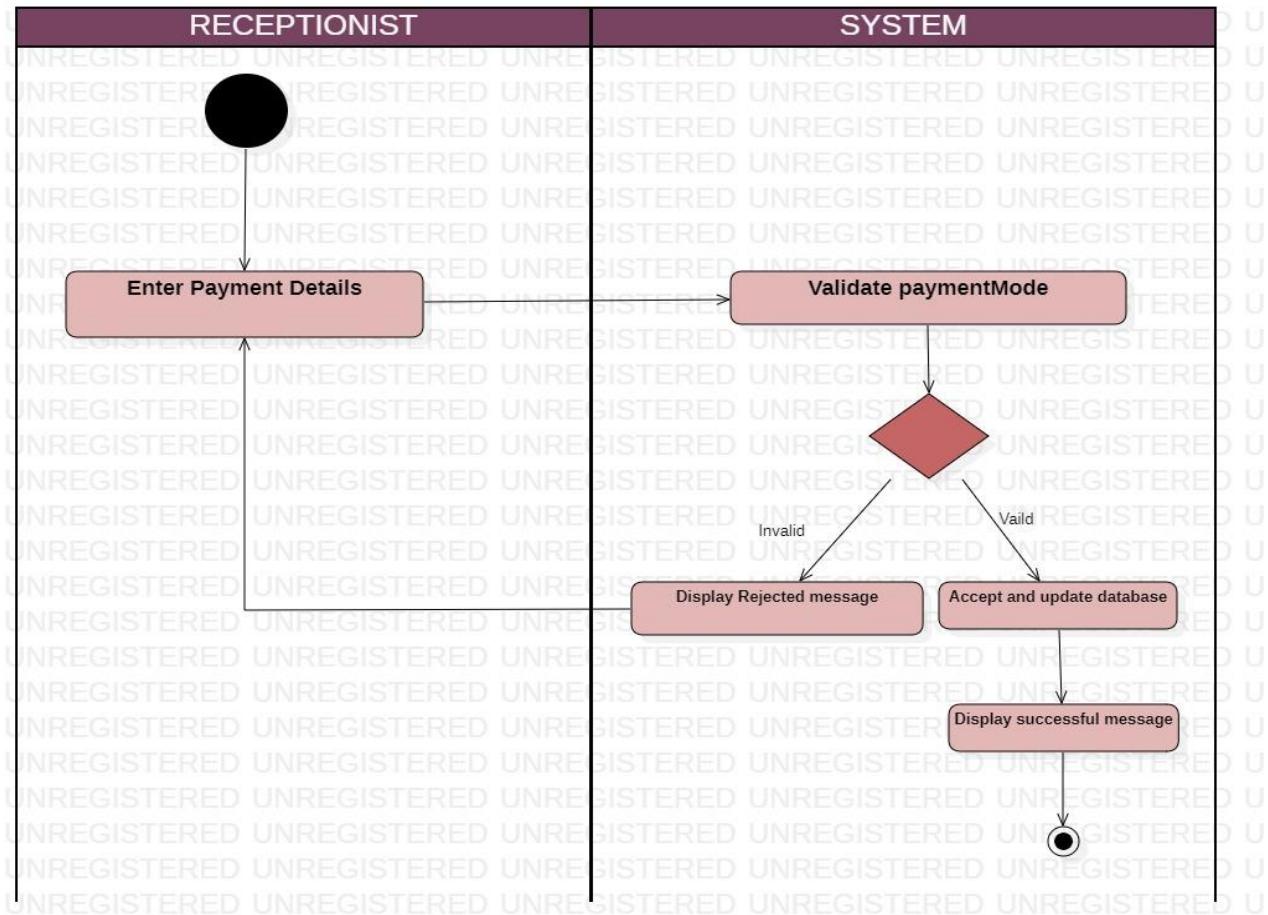
- Search Rooms



- Make Reservation

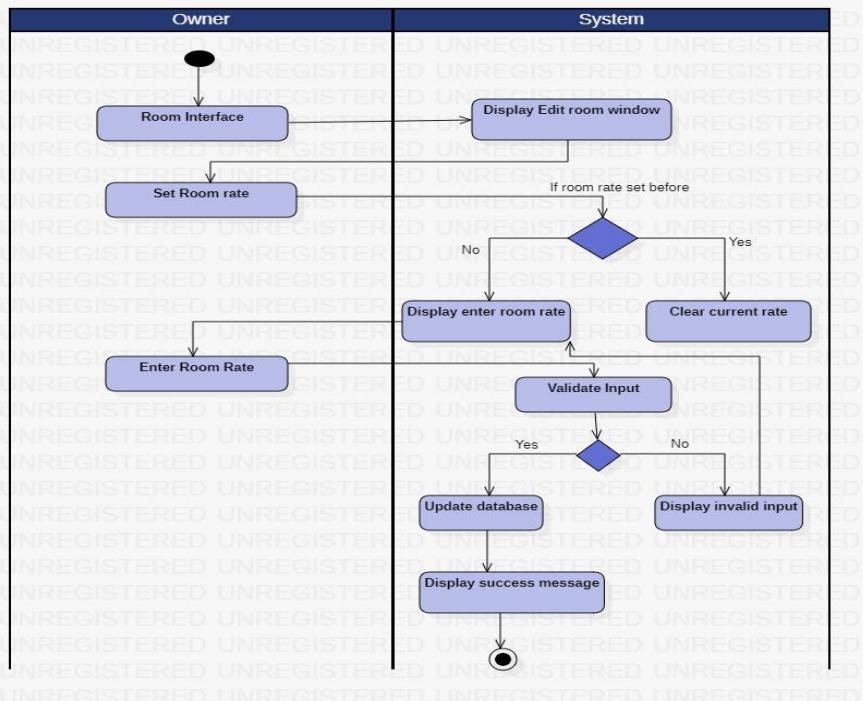


- Payment

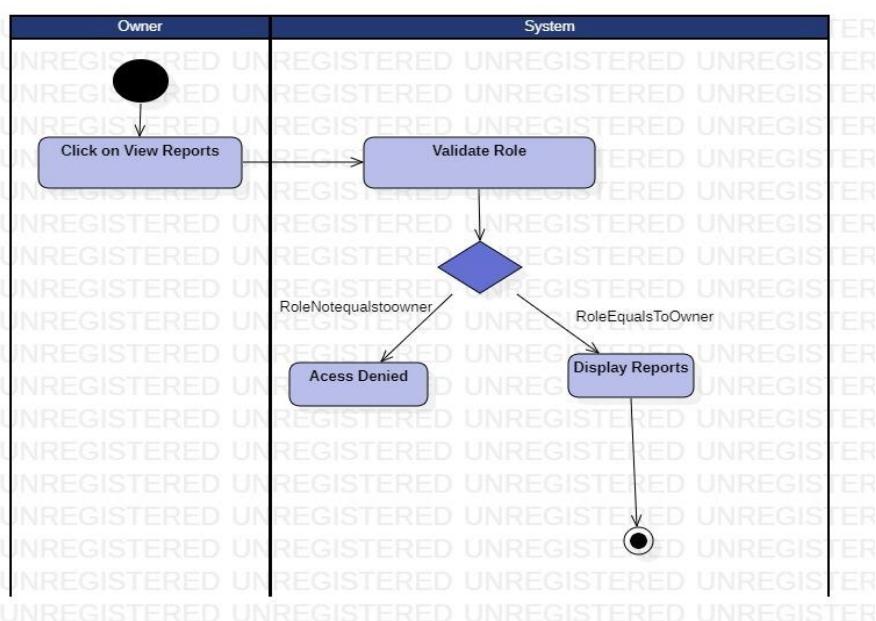


3.4.3 OWNER

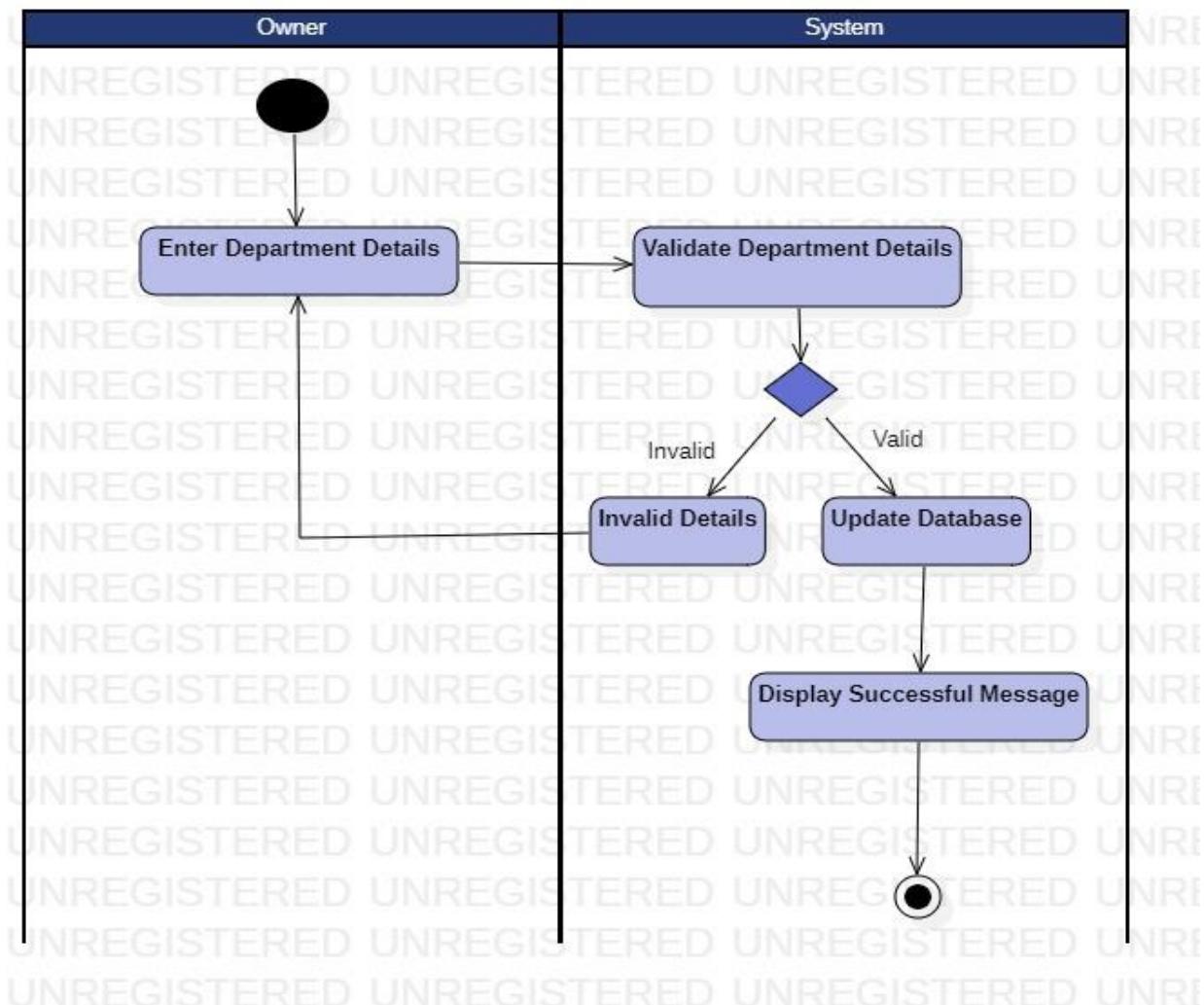
- Set Room Rate



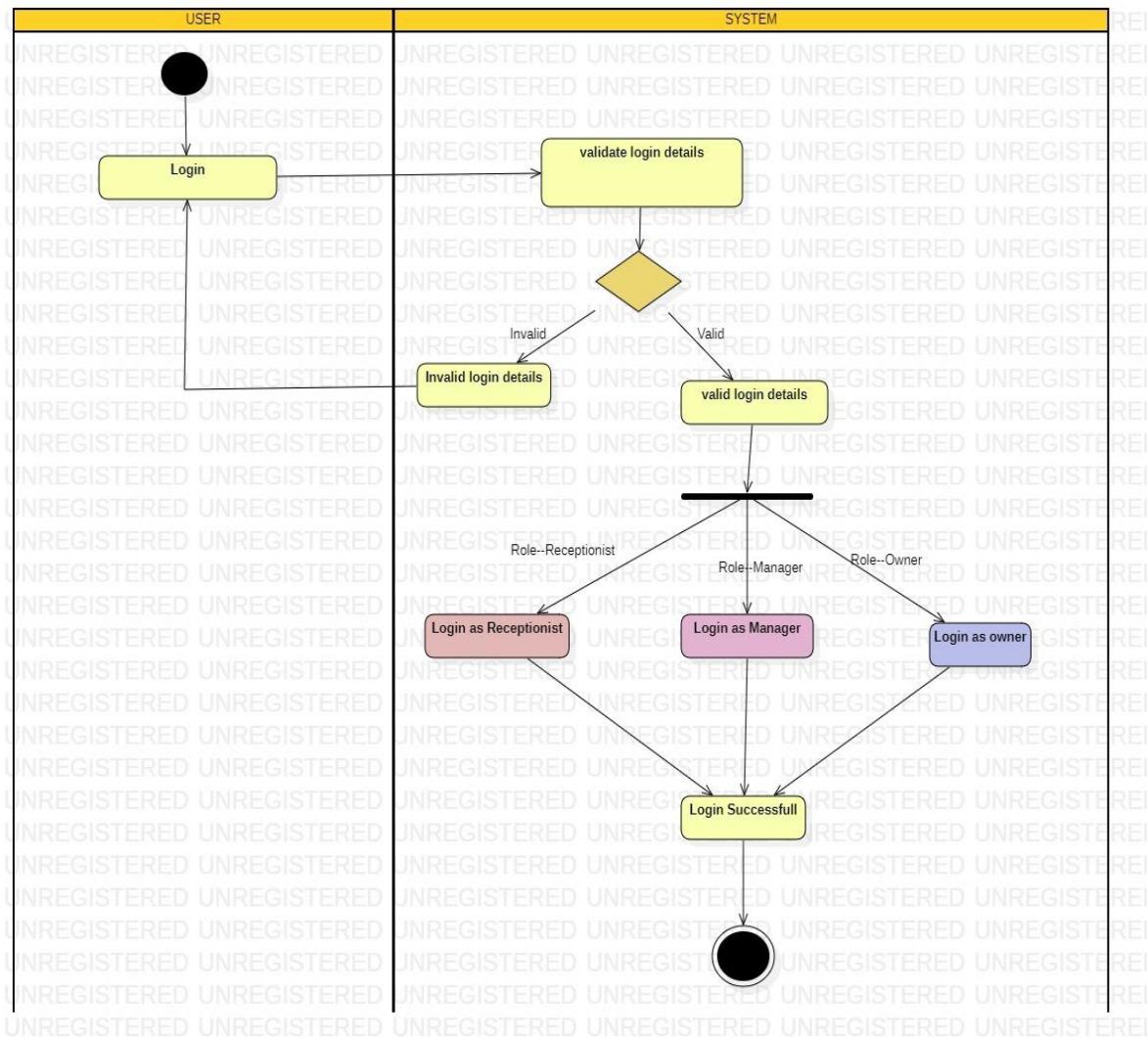
- **View Reports**



- Manage Department



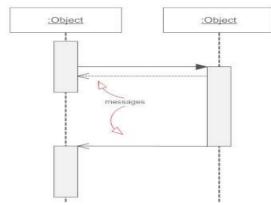
- **Login**



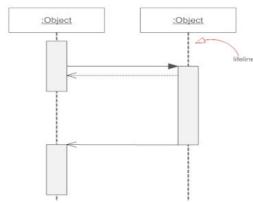
3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence.

- ❖ **Messages:** Messages are arrows that represent communication between objects.



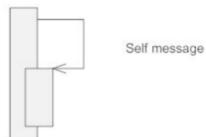
- ❖ **Lifelines:** Lifelines are vertical dashed lines that indicate the object's presence over time.



- ❖ **Reply or Return Message:** A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline.

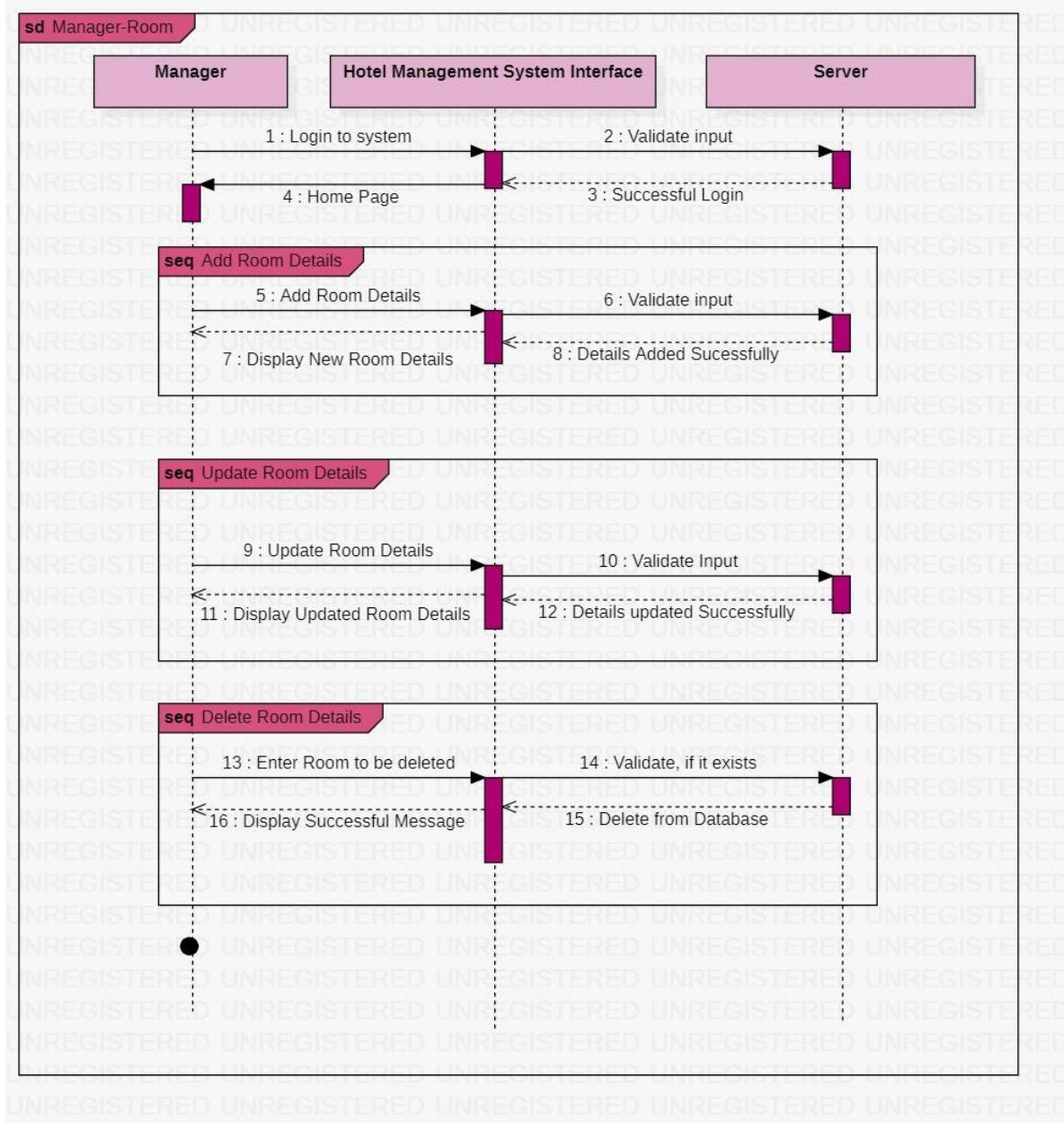
← Reply or return message

- ❖ **Self-Message:** A message an object sends to itself.

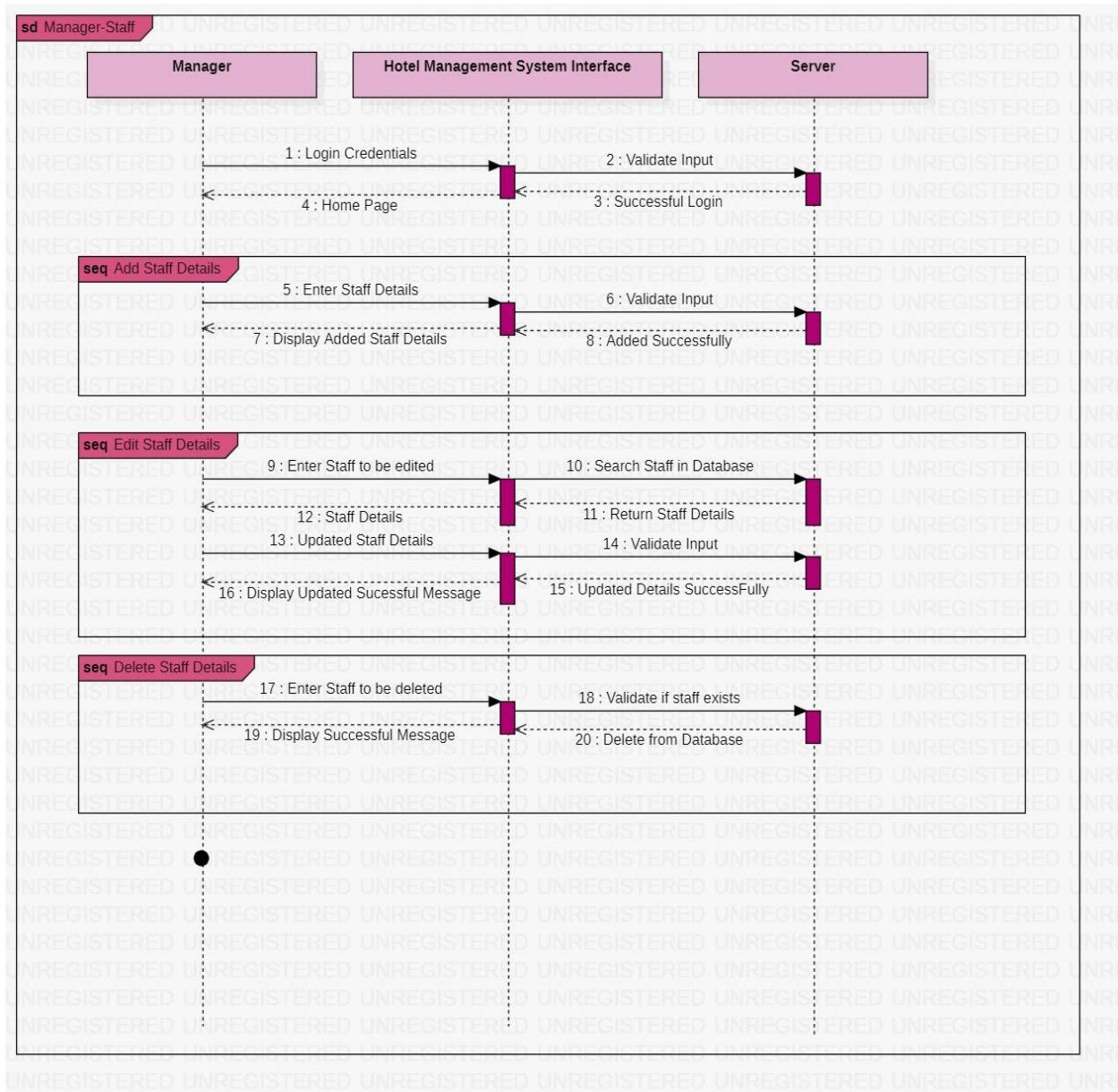


3.5.1 MANAGER

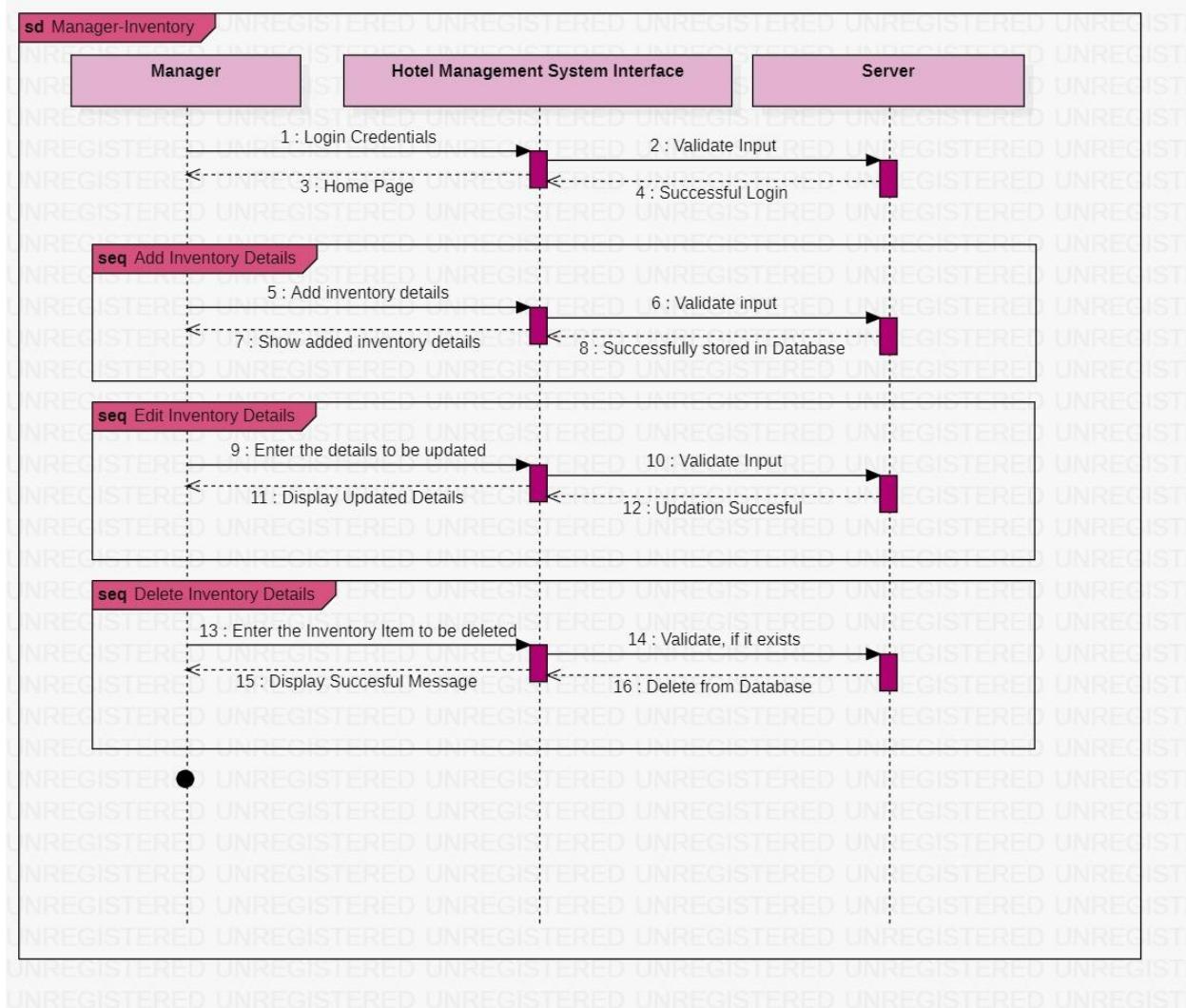
- Room Sequence Diagram



● Staff Sequence Diagram

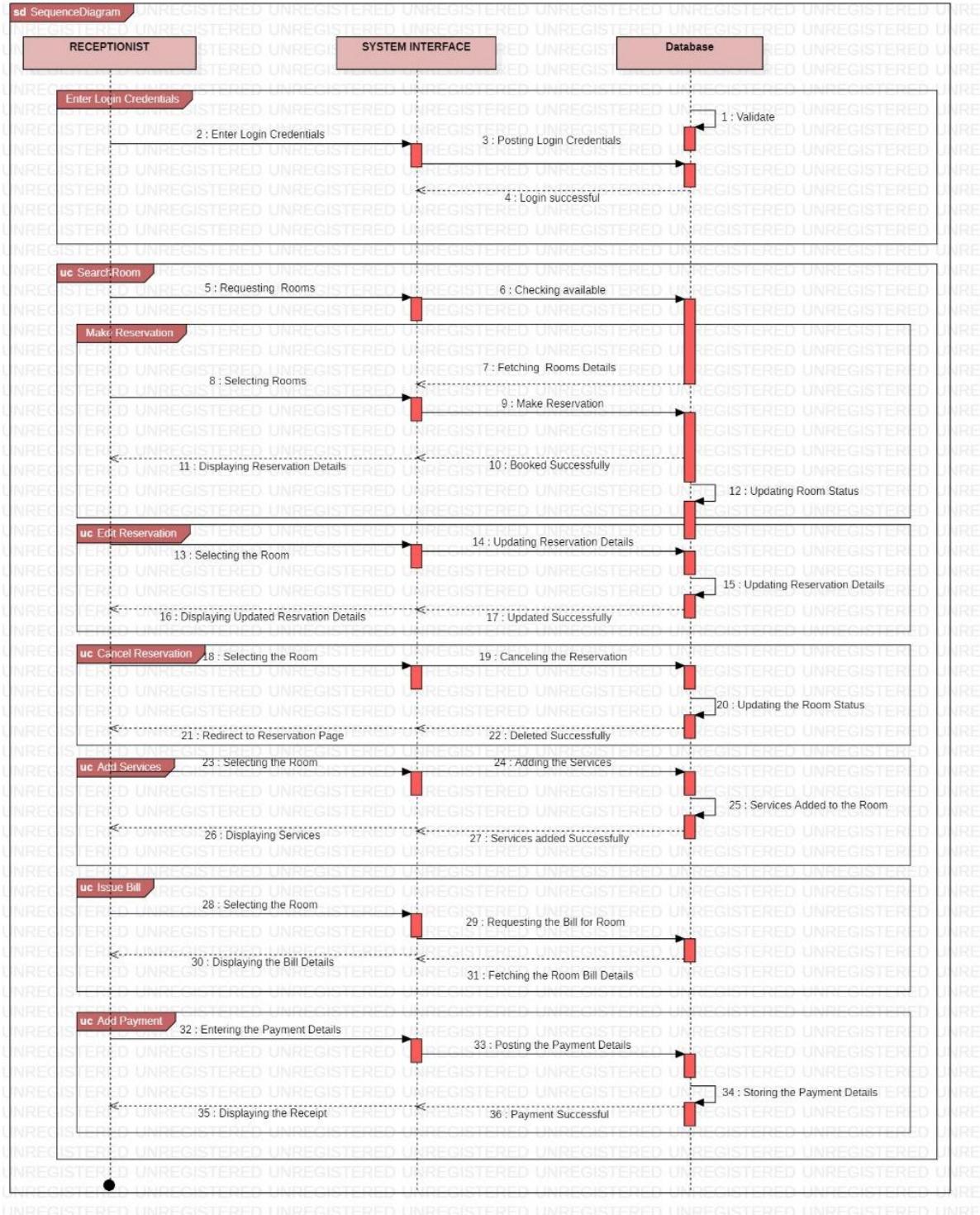


● Inventory Sequence Diagram

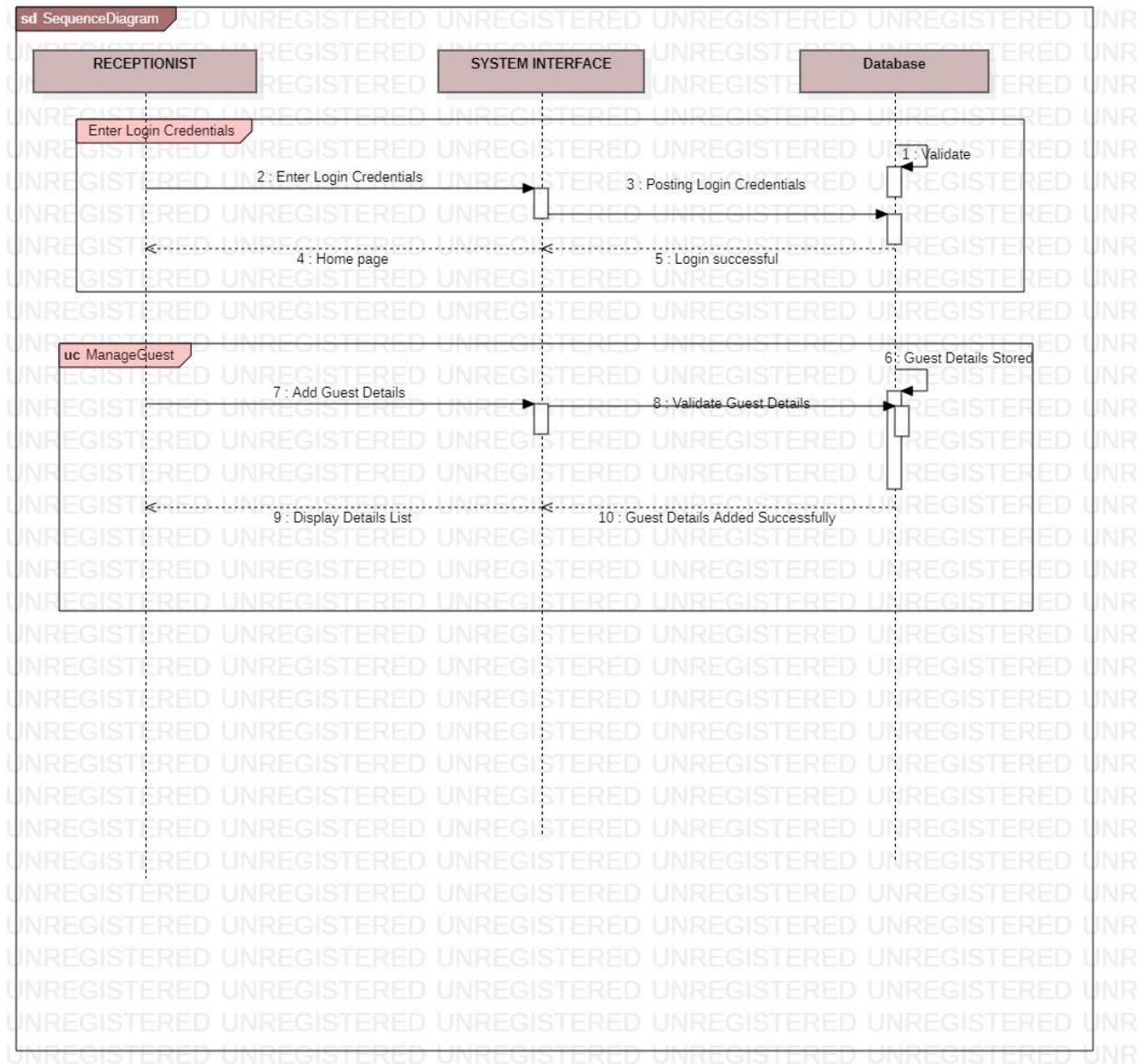


3.5.2 RECEPTIONIST

- **Reservation Sequence Diagram**



● Guest Sequence Diagram



3.5.3 OWNER

- **Owner Sequence Diagram**

