Carti Hotel Management System

Use-Case-Realization Specification: Manage Rooms

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24/11/2024 | 1.0 | Final version | Nguyen Huu Dang |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Flow of Events—Design 4

2.1 Flow of Events 4

2.1.1 Add room 4

2.1.2 Update room 4

2.1.3 Delete room 5

2.2 Sequence diagram 5

2.3 Class diagram 8

3. Derived Requirements 8

Use-Case-Realization Specification: Manage Rooms

# Introduction

## Purpose

This document outlines the collaborating objects used to fulfill the Manage Rooms use case inside the Carti Hotel Management System project's design model.

## Scope

This document applies to the Carti Hotel Management System.

## Definitions, Acronyms, and Abbreviations

None.

## References

None.

## Overview

The subsequent section comprehensively addresses the design of the use case along with its associated requirements. In particular, the flow of events-design section includes the implementation of the class diagram and sequence diagram for the use case, accompanied by concise textual explanations.

Following the design presentation, the subsequent section outlines the derived requirements crucial for the successful implementation of the use case. These requirements serve as essential guidelines to ensure the effective realization and functionality of the use case.

# Flow of Events—Design

## Flow of Events

The following diagrams outline the essential classes and the sequence of their collaboration to bring about the realization of the Manage Rooms use case.

Upon the successful login of the user as an admin, they are directed to the Admin Page. The admin selects **Manage Rooms** button on the left-hand side, invoking loadRoomInfo() to see the room list. Subsequently, the admin is able to manage all rooms, encompassing tasks such as adding, updating, and deleting rooms.

2.1.1 Add room

When the admin clicks the **Add Room** button, displayAddRoomModal() is called, displaying an Add Room Modal. The admin inputs the room number, room type, floor, and the price of the room type to create a new room, followed by clicking the **Save** button at the bottom of the modal, which triggers RoomController’s addRoom() function.

The system will then verify the existence of the room and validate other information by invoking checkValidRoom(). If the room does not already exist and the information is valid, the Room Controller proceeds addRoom() to add the room information to the Room database and display a success message by calling displaySuccessMessage(). If the room already exists or any information is invalid, displayErrorMessage() shows an error message is presented to inform the admin of the situation.

2.1.2 Update room

On the room list, the admin clicks the edit icon at the right side of each room to see **Update Room** and **Delete Room** options. Clicking on the **Update Room** button, the admin sees the Update Room Modal displaying the current rooms in the form of a table, which is displayed by invoking displayUpdateRoomModal().

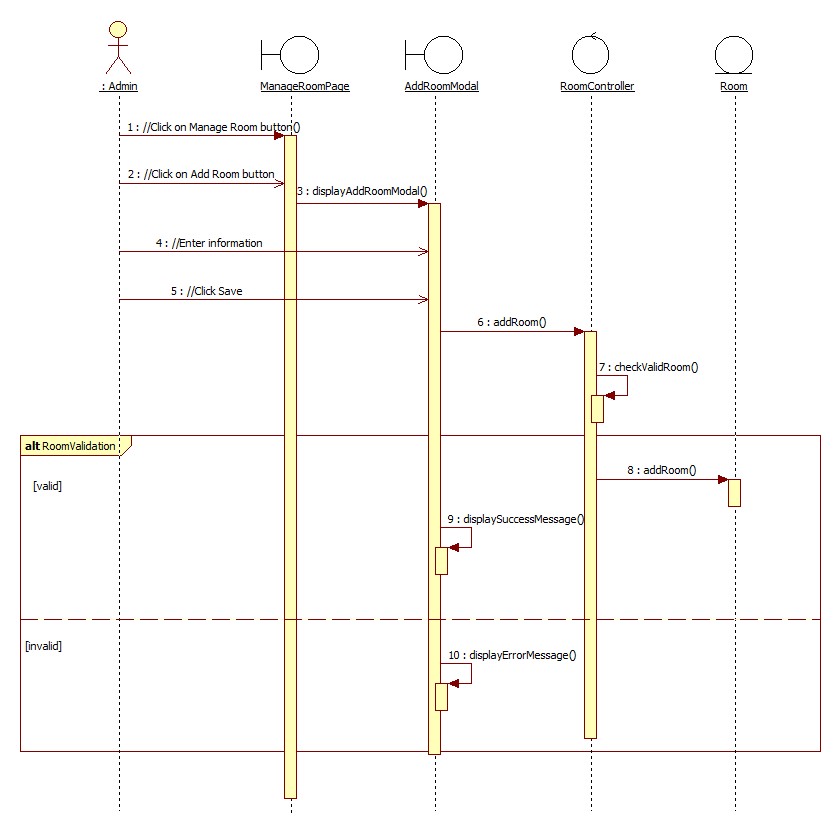
On the Update Room Modal, the admin modifies the information of the room and clicks the **Save** button to invoke updateRoom(). If the checkValidRoom() for new information is valid, the Room Controller updates the Room database with updateRoom() and calls displaySuccessMessage() to display a success message back to the user. Else, displayErrorMessage() is called to notify the admin.

2.1.3 Delete room

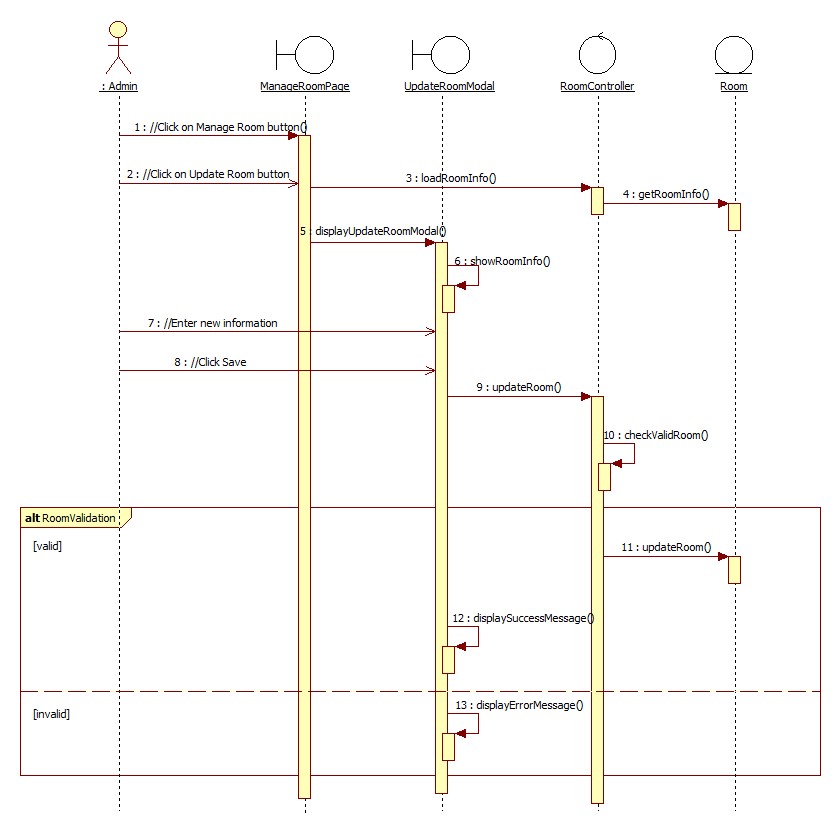
On the room list, the admin clicks the edit icon at the right side of each room to see **Update Room** and **Delete Room** options. The admin clicks on the **Delete Room** button, showDeletePrompt() is invoked, displaying a prompt to confirm the deletion. The admin then selecting **Sure**, the Room Controller deletes the room by calling deleteRoom() and display a notification by calling displaySuccessMessage().

## Sequence diagram

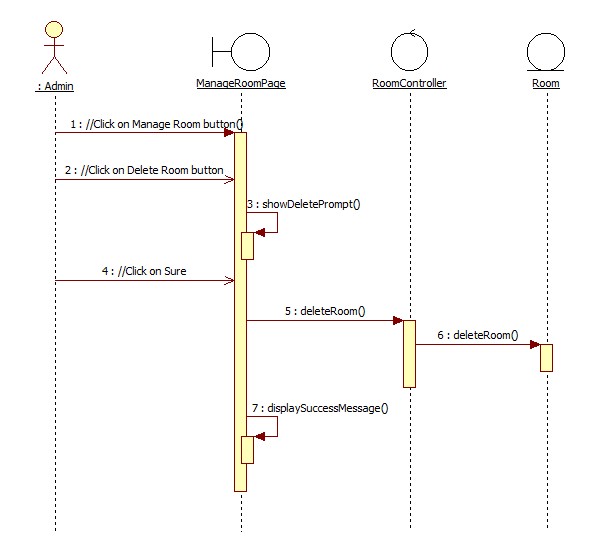
2.2.1 Add Room



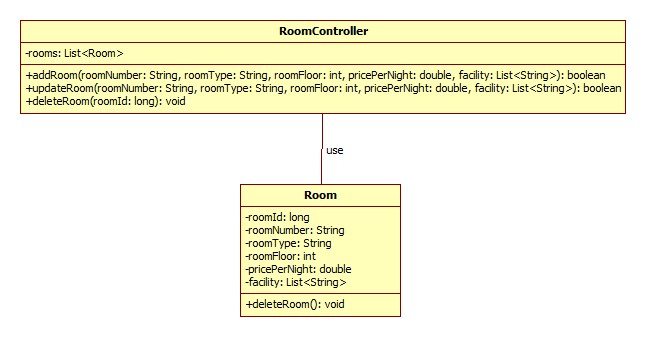
2.2.2 Update Room



2.2.3 Delete Room



## Class diagram



# Derived Requirements

None