**FIT Company**

**Carti Hotel Management System Use-case-Realization Specification: Manage Users**

**Version <1.0>**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24th Nov 2024 | Version 1.0 | Final version | Tran Quang Huy |
|  |  |  |  |

**Table of Contents**

[1. Introduction 4](#_Toc153891218)

[1.1 Purpose 4](#_Toc153891219)

[1.2 Scope 4](#_Toc153891220)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc153891221)

[1.4 References 4](#_Toc153891222)

[1.5 Overview 4](#_Toc153891223)

[2. Flow of Events—Design 4](#_Toc153891224)

[2.1. Flow of events 4](#_Toc153891225)

[2.1.1. Add User 4](#_Toc153891226)

[2.1.2 Update User 4](#_Toc153891227)

[2.1.3. Delete User 5](#_Toc153891228)

[2.2. Sequence Diagram 5](#_Toc153891229)

[2.2.1. Add User 5](#_Toc153891230)

[2.2.2. Update Use 6](#_Toc153891231)

[2.2.3. Delete User 7](#_Toc153891232)

[2.3 Class Diagram 8](#_Toc153891233)

[3. Derived Requirements 8](#_Toc153891234)

**Use-Case-Realization Specification: Manage Users**

# Introduction

## Purpose

This document outlines the realization of the "Manage Users" use case within the design model of the Carti Hotel Management System project, specifically focusing on the collaboration of objects.

## Scope

This document applies to the Carti Hotel Management System.

## Definitions, Acronyms, and Abbreviations

None

## References

None

## Overview

The next section thoroughly covers the design of the use case and its related requirements. In particular, the flow of events-design portion incorporates the development of both the class diagram and sequence diagram for the use case, along with brief written explanations.

After presenting the design, the following section details the derived requirements that are vital for the successful execution of the use case. These requirements act as key guidelines to guarantee the proper implementation and functionality of the use case.

# Flow of Events—Design

## Flow of events

The diagrams below illustrate the key classes and the order in which they interact to implement the Manage Users use case.

After a user logs in successfully as an admin, they are taken to the Admin Page. A function is triggered, followed by the Room Controller, which retrieves all room numbers and their availability statuses, displaying this information sequentially on the Admin Page. This setup enables the admin to effectively manage all users, including adding, updating, and deleting user accounts.

## Add User

When the admin selects the "Users" button and then chooses the "Add Users" option, they are redirected to the Add User Page. There, the admin enters the required information to create a new user and clicks save. If the information entered is valid, the User Controller inserts the new user's details into the User database and returns a success message to the admin. If the information is invalid, the admin receives an error message.

## Update User

When the admin selects the "Users" button and then chooses the "Update Users" option, the system invokes a function to fetch the list of users from the database and display it on the screen. The admin must click on a specific user, which will redirect them to the Update User Page, resembling the Add User Page.

Following this, the User Controller class activates a function that navigates the admin to the Update User Page for the selected user.

On the Update User Page, the admin enters the required information to modify the user's details and clicks the "Save" button. If the information provided is valid, the User Controller updates the User database and returns a success message to the admin. If the information is invalid, the admin receives an error message.

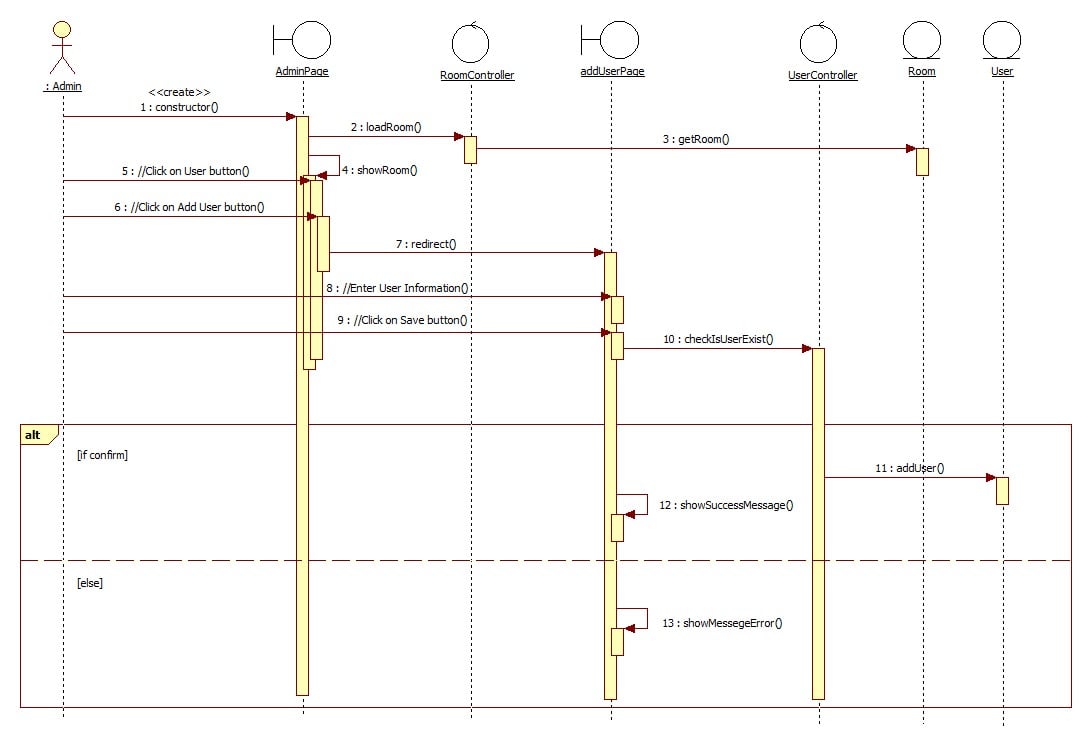
## Delete User

When the admin selects the "Users" button and then chooses the "Delete Users" option, the system triggers a function that redirects the admin to the Delete User Page, displaying a list of users on the screen. The admin must click on a specific user they wish to delete.

Upon clicking a particular User row, a confirmation dialog appears. If the admin confirms the deletion, the User Controller invokes a function to remove the user, updates the User database, and returns a success message to the admin. If the admin decides to cancel the deletion, the User Controller aborts the operation, and no further action is taken.

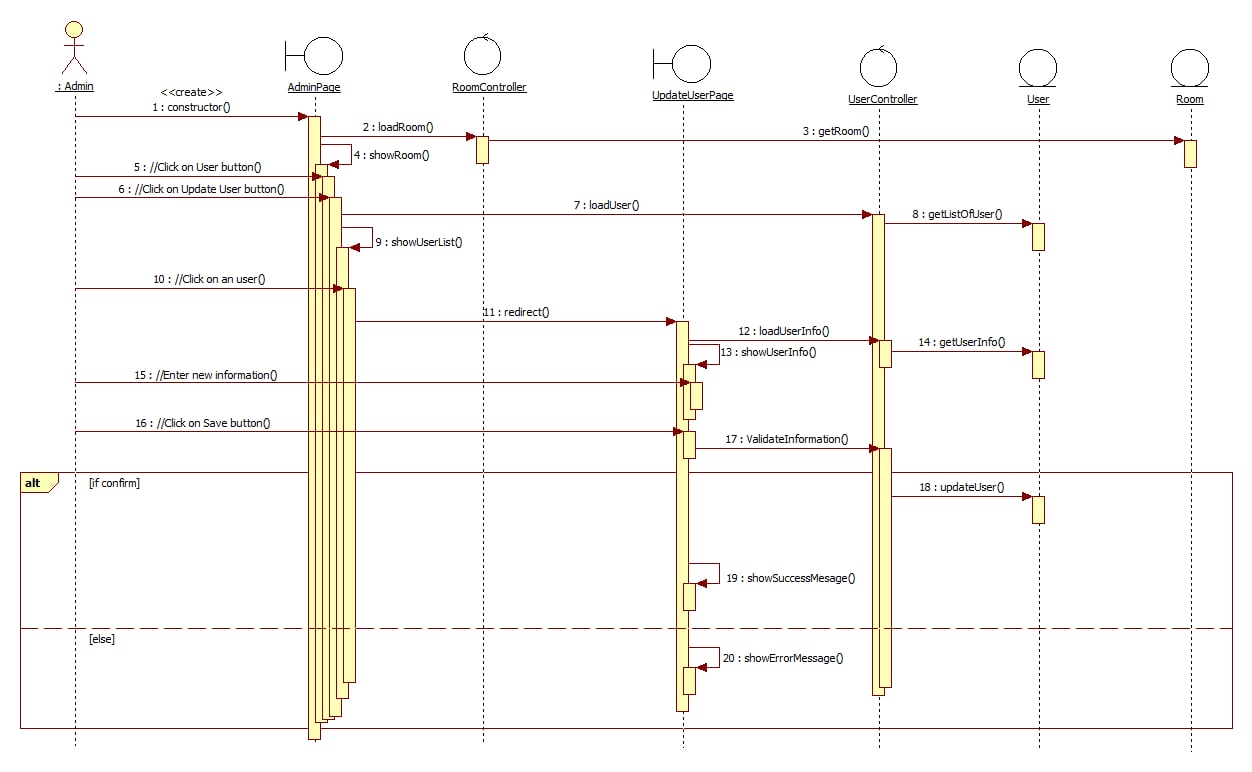
## Sequence Diagram

## Add User

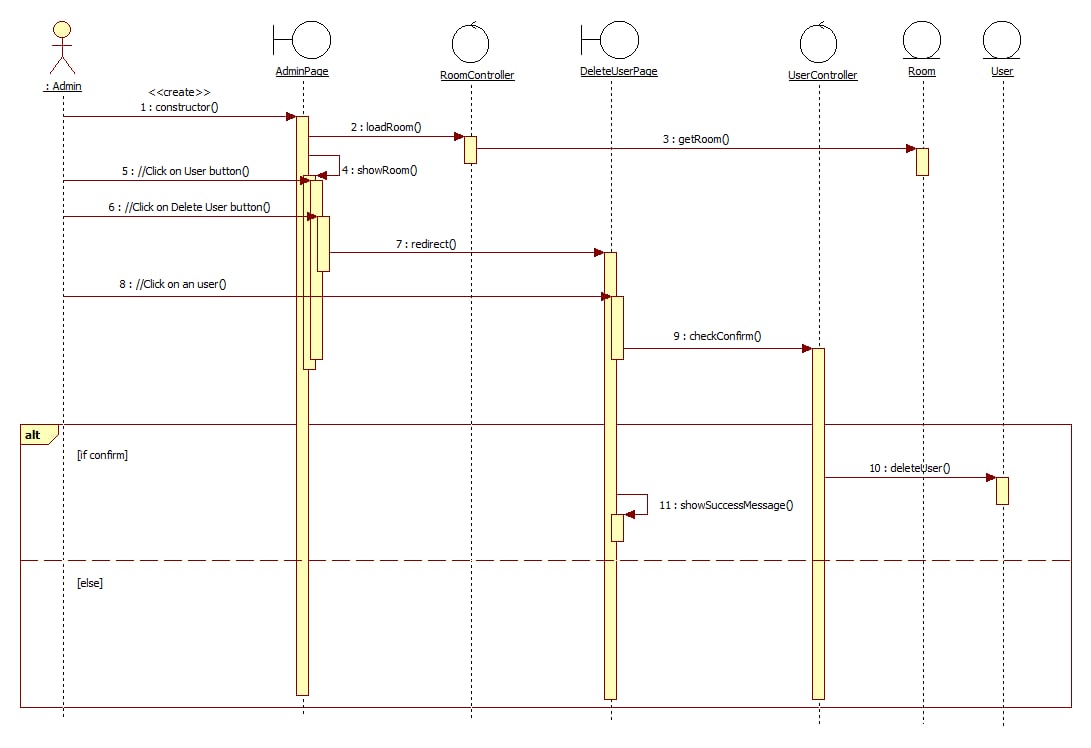
****

## Update Use

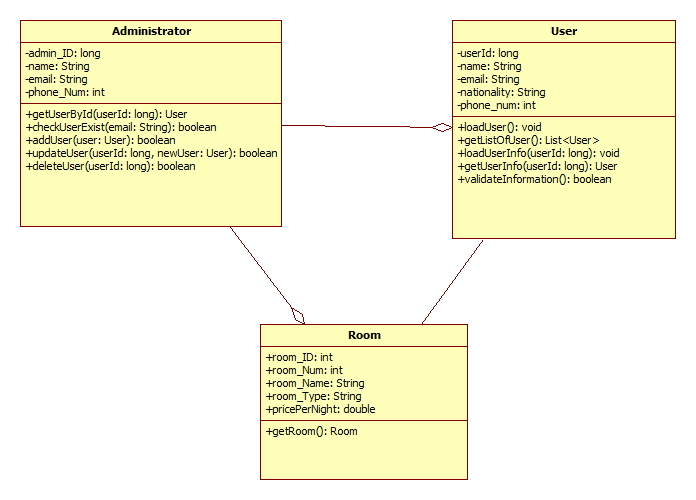
## 



## Delete User



## Class Diagram

****

# Derived Requirements

None