Check-in

Version <1.0>

Revision History

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

Table of Contents

1. Introduction 2

1.1 Purpose 2

1.2 Scope 2

1.3 Definitions, Acronyms, and Abbreviations 2

1.4 References 2

1.5 Overview 2

2. Flow of Events 2

3. Derived Requirements 2

Check-in

# Introduction

## Purpose

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

## Scope

This document applies to the Carti Hotel Management System which will be developed by FIT Company.

## Definitions, Acronyms, and Abbreviations

None

## References

None

## Overview

The design and derived requirements of the use case are addressed in the following sections. Specifically, the class diagram and sequence diagram for the use case are developed in the flow-of-events design section, accompanied by a concise textual explanation. The subsequent section outlines the derived requirements that must be considered during the implementation of the use case.

# Flow of Events—Design

The diagrams below illustrate the essential classes and their interactions required to implement the Check-in use case. The process begins on the Receptionist Page, where the receptionist enters the customer’s name into the search bar and clicks the Search button. The Search Controller processes the input, retrieves all matching customer records, and displays them as a list. The flow then diverges into two cases:

Case 1: At least one result is found

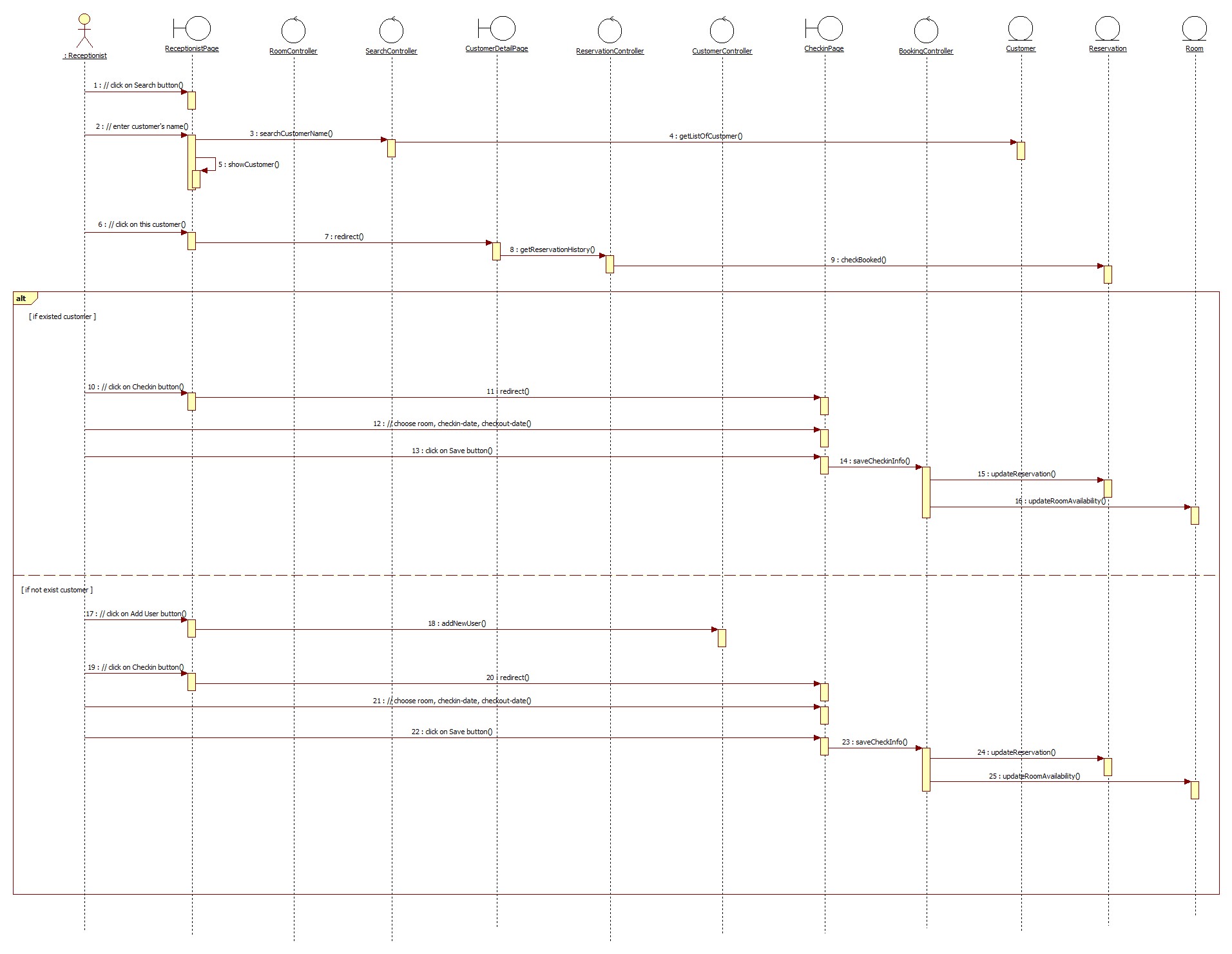
* The receptionist selects a customer from the list, navigating the system to the Customer Detail Page.
* The Booking Controller fetches the customer's reservation history and displays it. It also requests the Reservation Service to verify whether the customer has an active booking.
* If a booking exists: No further action is required.
* If no booking exists: The receptionist can proceed by clicking the Check-in button to navigate to the Check-in Page.
  + On this page, the receptionist selects a room, check-in date, and check-out date, then clicks Save.
  + The Booking Controller updates the Room and Reservation data. If the operation is successful, a confirmation message is displayed.

Case 2: No results are found

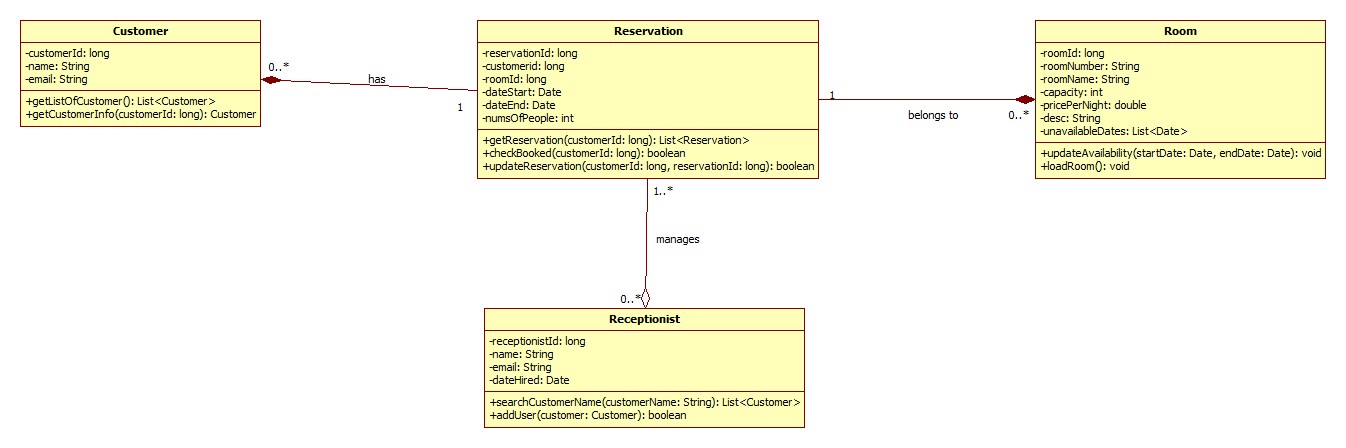
* The receptionist clicks the Add User button to create a new account for the customer, triggering the Add User class.
* After the account is created, the receptionist can click the Check-in button to proceed to the Check-in Page, following the same steps outlined in Case 1.

This workflow ensures seamless customer check-in, covering scenarios for both existing and new customers.

## Sequence diagram



## Class diagram



# Derived Requirements

None