Carti Hotel Management System

Use-Case-Realization Specification: Make Payment

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

2.2 Sequence diagram 5

2.3 Class diagram 6

3. Derived Requirements 6

Use-Case-Realization Specification: Make Payment

# Introduction

## Purpose

This document outlines the collaborating objects used to fulfill the Make Payment 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 use case's design and derived requirements are covered in the following part. More specifically, the use

case's class diagram and sequence diagram are realized in the flow of events-design part and are

supplemented by a succinct textual explanation.

The requirements that must be taken into account when implementing the use case are provided in the next

section by the derived requirements.

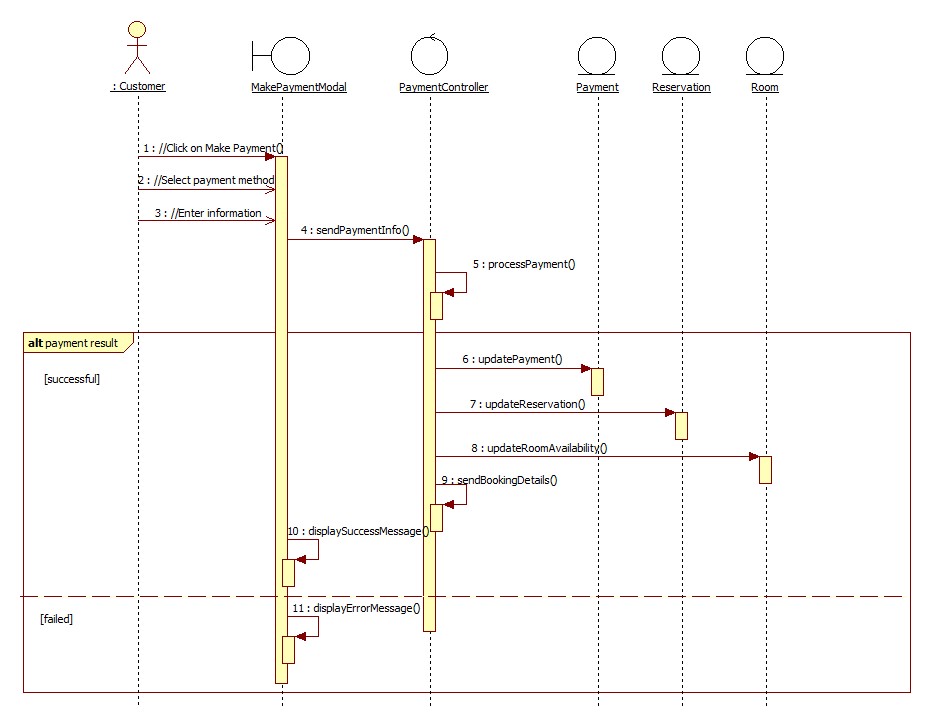
# Flow of Events—Design

## Flow of Events

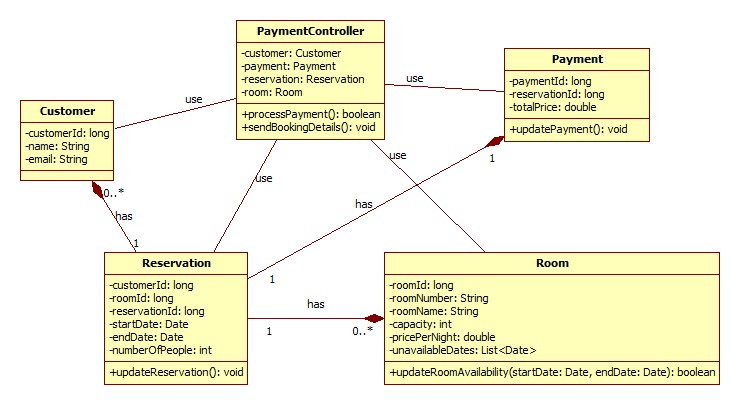
The two diagrams below describe the necessary classes and the sequence of collaborating them to realize the Make Payment use case.

When the customer is booking a room and he/she is on the Confirm Booking Page after checking the information entered for booking, the customer can click the **Make Payment Now** button to process the payment.  
Then the customer will see the Make Payment Modal, where payment method can be chosen and based on the selected method, they will be required information for the transaction. The PaymentController receives the information from senPaymentInfo() and process the payment by invoking processPayment(). If the payment is successful, The PaymentController invokes updatePayment(), updateReservation(), updateRoomAvailability(), and sendBookingDetails() to update the database, the displaySuccessMessage() will be invoke as well. Else, displayErrorMessage() will be invoke to notify the customer.

## Sequence diagram



## Class diagram



# Derived Requirements

None