Carti Hotel Management System

Software Requirements Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/11/2024 | 1.0 | Final Version | Nguyen Giang Minh |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Introduction 5](#_Toc183545114)

[1.1 Purpose 5](#_Toc183545115)

[1.2 Scope 5](#_Toc183545116)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc183545117)

[1.4 References 5](#_Toc183545118)

[1.5 Overview 5](#_Toc183545119)

[2. Overall Description 5](#_Toc183545120)

[2.1 Use- case Model Survey 5](#_Toc183545121)

[2.2 Assumption and Dependencies 6](#_Toc183545122)

[3. Specific Requirements 6](#_Toc183545123)

[3.1 Functionality 6](#_Toc183545124)

[3.1.1 Manage User 6](#_Toc183545125)

[3.1.2 Manage Room 6](#_Toc183545126)

[3.1.3 Book Rooms 7](#_Toc183545127)

[3.1.4 Manage Services 7](#_Toc183545128)

[3.1.5 Display Revenue Reports 7](#_Toc183545129)

[3.2 Usability 7](#_Toc183545130)

[3.2.1 Training Time 7](#_Toc183545131)

[3.2.2 User-Friendly Interface 7](#_Toc183545132)

[3.3 Reliability 7](#_Toc183545133)

[3.3.1 Availability 7](#_Toc183545134)

[3.3.2 MTBF (Mean Time Between Failures) 7](#_Toc183545135)

[3.3.3 Accuracy 7](#_Toc183545136)

[3.4 Performance 8](#_Toc183545137)

[3.4.1 Response Time 8](#_Toc183545138)

[3.4.2 Throughput 8](#_Toc183545139)

[3.4.3 Capacity 8](#_Toc183545140)

[3.5 Supportability 8](#_Toc183545141)

[3.5.1 Maintenance Access 8](#_Toc183545142)

[3.5.2 Coding Standards 8](#_Toc183545143)

[3.6 Design Constraints 8](#_Toc183545144)

[3.6.1 Prescribed Technologies 8](#_Toc183545145)

[3.7 On-line User Documentation and Help System Requirements 8](#_Toc183545146)

[3.8 Purchased Components 8](#_Toc183545147)

[3.9 Interfaces 8](#_Toc183545148)

[3.9.1 User Interfaces 8](#_Toc183545149)

[3.9.2 Hardware Interfaces 8](#_Toc183545150)

[3.9.3 Software Interfaces 8](#_Toc183545151)

[3.9.4 Communications Interfaces 8](#_Toc183545152)

[3.10 Licensing Requirements 9](#_Toc183545153)

[3.11 Legal, Copyright, and Other Notices 9](#_Toc183545154)

[3.12 Applicable Standards 9](#_Toc183545155)

[4. Supporting Information 9](#_Toc183545156)

Software Requirements Specification

# Introduction

## Purpose

The purpose of this Software Requirements Specification document is to define the development process of the final CHMS software product. It focuses on detailing the software's external behavior while also addressing nonfunctional requirements, design constraints, and other relevant factors.

## Scope

This document pertains to the CHMS and all the particulars outlined herein are established in accordance with the operational principles of The Hotel Center.

## Definitions, Acronyms, and Abbreviations

CHMS – Carti Hotel Management System

## References

None.

## Overview

The following sections of this document cover the general description, specific requirements, and supporting information. While the general description does not define specific criteria, it establishes the context for the requirements. The Overall Description highlights the key factors shaping the product and its needs. The Specific Requirements section provides detailed software requirements, guiding designers in building the system and enabling testers to verify its compliance. Finally, the document concludes with Supporting Information.

# Overall Description

This section highlights the overarching factors influencing the product and its requirements. While it does not specify detailed criteria, it provides a contextual foundation for the requirements elaborated in Section 3, aiding in better understanding.

## Use- case Model Survey

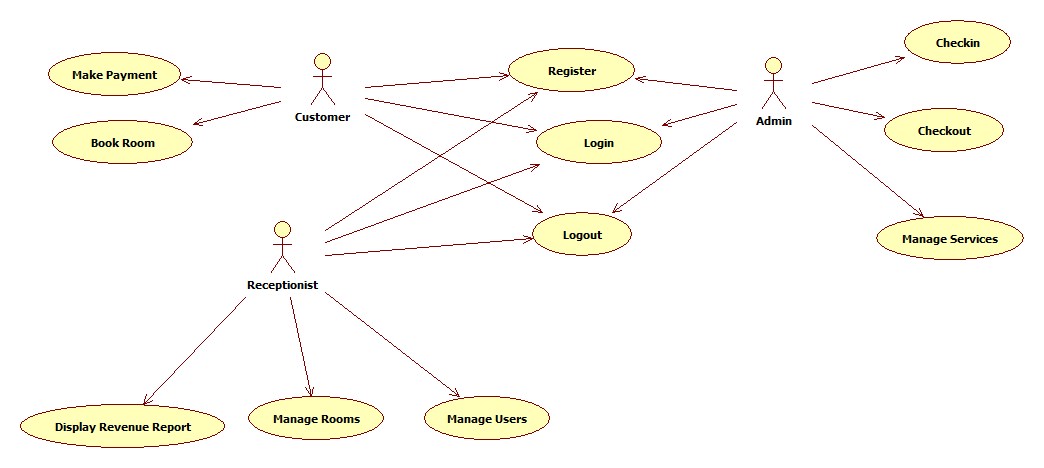
The software's use-case model encompasses three actors and eight primary high-level use cases. The actors in this model are the system administrator, receptionist, and customer. The eight main high-level use cases, along with their sub-units, are as follows: [additional details can be provided here based on the specific use cases and sub-units:

* + - Manage User (Add user, Update user, Delete user): administrator
    - Manage Room (Add room, Update room, Delete room): administrator
    - Display revenue report (Show statistic, export excel file): administrator
    - Manage Services (Add service, Delete service): receptionist
    - Check-in(Search customer, search customer’s reservation, book room for customer): receptionist
    - Check-out (Search customer, search customer’s reservation, print bill): receptionist
    - Book rooms (Search room, Reserve room): customer
    - Make payment (Choose payment type, Make transaction): customer

## Assumption and Dependencies

The project assumes that alumni data will be accessible at some point during its development. Until completion, demo presentations will utilize test data. Users are expected to have familiarity with desktop applications, as well as proficiency in using a keyboard and mouse. Given that the application is desktop-based, users are assumed to have adequate database connectivity skills.

# Specific Requirements



## Functionality

### Manage User

**Description**: The system allows administrators to manage users.

**Functional Requirements**:

* 1. Add new user accounts (username, password, role).
  2. Update existing user information.
  3. Delete user accounts.
  4. Restrict user access based on roles.

### Manage Room

**Description**: The system enables administrators to manage room data.

**Functional Requirements**:

* 1. Add new rooms with attributes such as room number, type, and rate.
  2. Update room information (e.g., availability, features).
  3. Remove rooms from the system.

### Book Rooms

**Description**: Customers can book rooms using the system.

**Functional Requirements**:

* 1. Search available rooms based on date, type, or price.
  2. Reserve a room for a specific duration.
  3. Generate a booking confirmation.

### Manage Services

**Description**: Receptionists can manage hotel services.

**Functional Requirements**:

* 1. Add or remove services (e.g., spa, laundry).
  2. Update service details (name, cost).
  3. Associate services with specific customer reservations.

### Display Revenue Reports

**Description**: Administrators can view financial reports.

**Functional Requirements**:

* 1. Display daily, monthly, or yearly revenue.
  2. Export reports as Excel or PDF.

## Usability

### Training Time

Normal User: A customer should be productive in using the booking system after 1 hour of training.

Power User: Receptionists and administrators should be productive after 3 hours of training.

### User-Friendly Interface

The interface must comply with all most GUI standards.

## Reliability

### Availability

The system must be operational 99.9% of the time, excluding scheduled maintenance.

### MTBF (Mean Time Between Failures)

The system should achieve at least 1000 hours of uninterrupted operation.

### Accuracy

Reservation calculation and reports must be accurate within a 0.01% margin of error.

## Performance

### Response Time

The system must respond to user actions within 2 seconds for 95% of all requests.

### Throughput

Handle up to 500 concurrent users without performance degradation.

### Capacity

Support up to 10,000 room bookings per day.

## Supportability

### Maintenance Access

The system must provide admin-level access tools for database backup and restoration.

### Coding Standards

Follow industry best practices for naming conventions, modular design, and reusable code.

## Design Constraints

### Prescribed Technologies

Programming Language: Use Java for backend and Angular for frontend.

Database: Must use MySQL.

Framework: Adhere to RESTful API design principles.

## On-line User Documentation and Help System Requirements

* Provide an integrated Help Center accessible from the software.
* Include interactive tutorials, FAQs, and step-by-step guides for common tasks.

## Purchased Components

Use licensed Chart.js for graphical reports and PDFKit for document generation.

## Interfaces

### User Interfaces

* Provide an interactive dashboard for administrators and receptionists.
* Ensure the UI is responsive across devices (desktop, tablet, mobile).

### Hardware Interfaces

Support printers for invoice generation and barcode scanners for customer check-in.

### Software Interfaces

Integrate with third-party payment systems (e.g., PayPal, Stripe).

### Communications Interfaces

Use HTTPS for secure data transmission and SMTP for sending emails.

## Licensing Requirements

Enforce licensing through **activation keys** valid for specific versions.

## Legal, Copyright, and Other Notices

Include copyright information and trademark disclaimers for third-party components.

## Applicable Standards

None.

# Supporting Information

TBD