SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT (Software Requirements Specification – SRS)

Build Hotel Management Web Application

Composed by: Hoang Dang Khai – 22028022
Tran Duy Tuan Anh – 22028228
Vu Viet Hung – 22028124
Le Van Luong – 22028040
Nguyen Duy Anh Quoc - 22028161

Revision History

Date	Version	Description	Author
5/5/2024	1.0	Add Revision History	Khai
10/5/2024	2.0	Refractor document	Khai

Table of Contents

1. Introduction

1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the user requirements and constraints for the Hotel Management System (HMS). This SRS will enable a complete understanding of what is expected from the newly introduced system to be built. A clear understanding of the system and its functionality will enable the correct software development for the end user and will be used to develop future phases of the project. From this SRS, System Hotel management can be designed, built and finally tested.

This SRS will be used by the system development team building the HMS. Hotel end users will be able to use this SRS as a “test” to see if the construction team has built the system according to their expectations. If it is not as expected, the end user can specify how it is not to their liking and the team will change the SRS to fit the end user's needs. Hotel managers and receptionists can also use this document to better understand the system and how to use it.

1.2 Scope

The software product that will be produced is a Hotel Management System that will automate key hotel operations.

The first subsystem is the Room Viewing and Reservation System to track the number of rooms booked, rooms in use, booked rooms and rooms about to check out. The second subsystem is the Room and Room Categories System Allows viewing hotel room types, adding, editing, deleting rooms and room types. The system also helps us see the list of available rooms for each room type. The functions of these subsystems will be described in detail in part 2 - Overall description.

There are two end users for HMS. The end users are hotel receptionists (customer service representatives) and hotel Managers. Both types of users can access the Viewing and Booking System. Only Managers can access the Room and Room Catalog System. The objective of the Hotel Management System is to provide a system to manage a hotel that has increased in size to a total of 1000 rooms. Without automation, hotel management becomes a difficult task. The end user's daily hotel management work will be greatly simplified through the automated system. The system will be able to handle multiple services to take care of all customers quickly. The system must be user-friendly, easy to use, easy to troubleshoot, and provide high subjective satisfaction to the end user.

1.3 Definitions, acronyms, and abbreviations

SRS: Software Requirements Specification
HMS: Hotel Management System
HTML: HyperText Markup Language
CSS: Cascading Style Sheets

1.4 References

Internet References:

- Best hotel Management Software - 2024 Reviews & Pricing (softwareadvice.com)
- Hotel Management - Case Study | PDF | Database Transaction | Software (scribd.com)
- Our hotel Management System Software - High Level Software (high-level-software.com)

Book References:

- Sommerville-Software-Engineering-10ed
- IEEE Recommended Practice for Software Requirements Specifications

1.5 Overview

This document is the Software Requirements Specification (SRS) for Hotel Management Systems (HMS). The purpose of this SRS is to provide detailed information on the functional and non-functional requirements of the HMS.
This document is organized into the following main sections:
Introduction: This section will provide an overview of the entire SRS.
Overall description: This part of the SRS describes the general factors affecting the product and its requirements. This section does not state specific requirements. Instead, it provides a foundation for those requirements, described in detail in Part 3 of the SRS, and makes them easier to understand.

Specific requirements: This section of the SRS includes all software requirements at a level of detail sufficient to allow designers to design a system that meets those requirements and that testers can test to see if the system meets those requirements or not. Throughout this section, any stated requirements will be externally perceivable to users, operators, or other external systems. These requirements shall include, at a minimum, a description of all inputs (stimuli) to the system, all outputs (responses) from the system, and all functions performed by the system in response with input or output support.

2. Overall description

2.1 Product Perspective:

This software is specially developed for hotel room management. It provides a simple database and a beautiful graphical user interface that is simple and easy for even new users.

2.2 Product Functions:

General functions of the product:

- Add, edit, delete rooms, room types and information about rooms, room types
- Book rooms, cancel rooms
- Display room information: room number, room type, price, booked Booked or not...
- Payment
- Management of booking data
- Check in, check out.

2.3 Users Characteristics

There are 2 user levels in our Hotel Management System:

A. Hotel manager

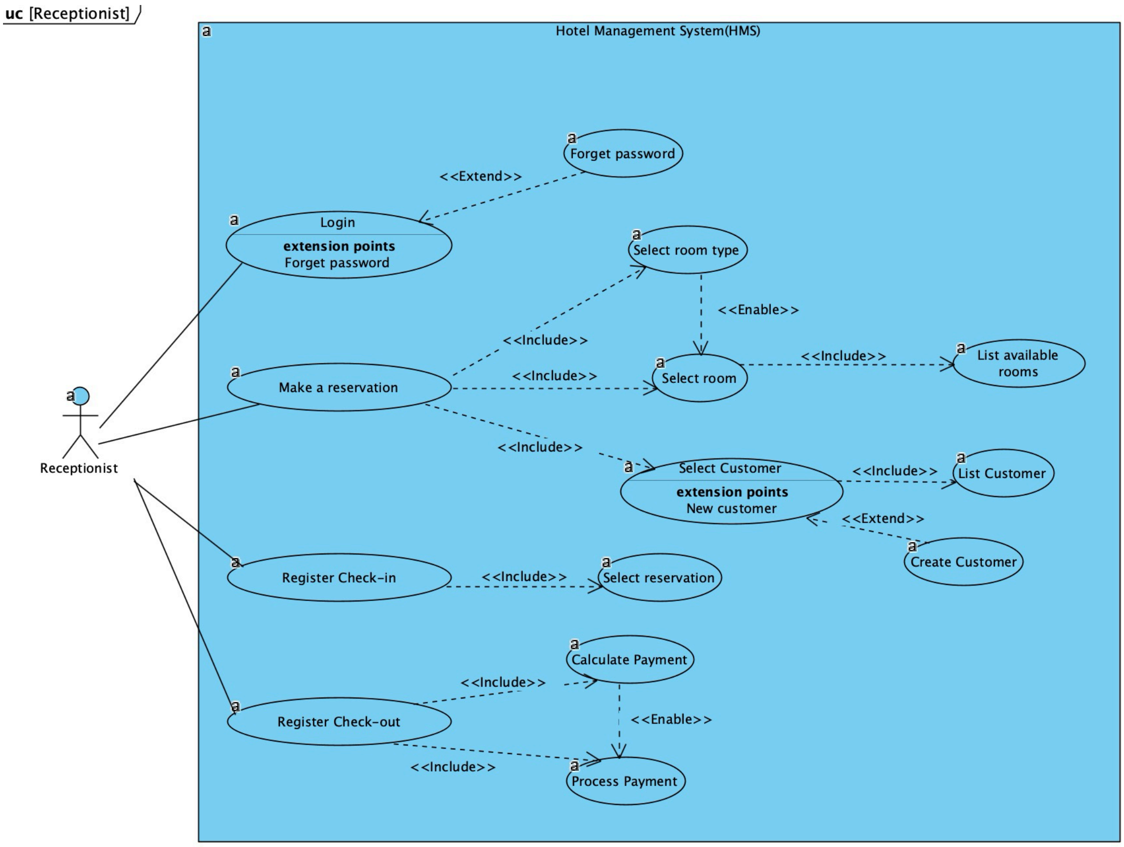
The manager has all access to the hotel system. The manager is responsible for managing the hotel's resources and staff. Managers can view any reports such as financial reports, customer information, booking information and room information, analyze them and make appropriate decisions. Managers can add, edit, and delete rooms, room types, and related information. Managers are required to have previous hotel management experience and basic knowledge of databases and application servers.

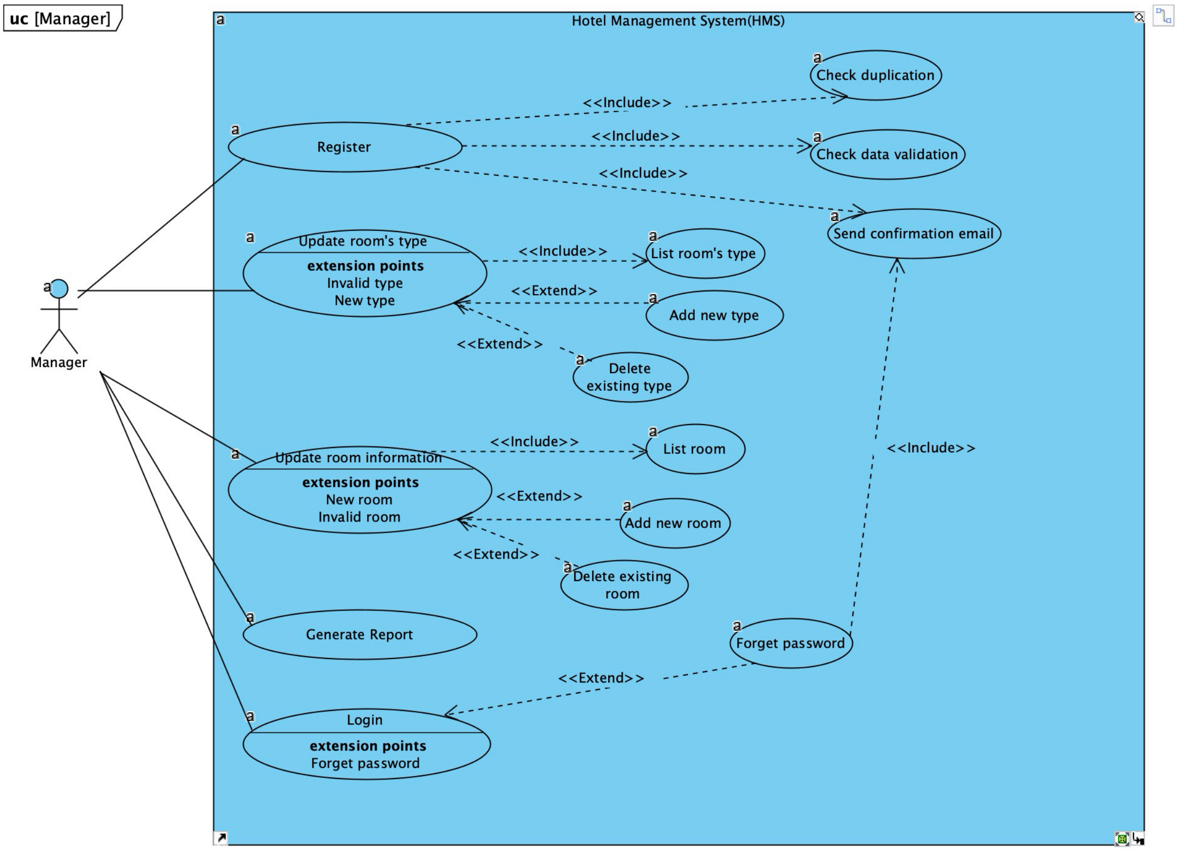
B. Receptionist

The sole purpose of a hotel reception is to provide quality service to customers. The receptionist has less access than the manager. Reception can manage reservation details. Receptionists can search for room availability, add guests, confirm reservations, and update reservation details. Receptionists should have experience in providing services, communicating with customers, and operating computers.

User stories:

- As a registered user, I can log in, so that I can access the system.
- As a forgetful user, I can request a password reminder so that I can log in if I forget mine.
- As a receptionist, I want to check room status, so that I know which rooms are empty, in use, or need cleaning.
- As a receptionist, I want to make reservations, so that I can help customers reserve rooms.
- As a receptionist, I want to be able to check-in customers, so that they can start their stay.
- As a receptionist, I want to be able to check-out customers, so that they can finish their stay.
- As a receptionist, I want to be able to see my customers' booking history, so that I can provide better service.
- As a manager, I want to update room information, so that I can provide accurate information on the system about our hotel.
- As a manager, I want to see revenue reports, so that I can monitor the hotel's business situation.
- As a manager, I want to manage customer information, so that I can provide better service to customers.


uc [Receptionist]
Hotel Management System(HMS)
Receptionist
Forget password
<<Extend>>
Login
extension points
Forget password
Select room type
<<Enable>>
<<Include>>
Make a reservation
<<Include>>
Select room
<<Include>>
List available rooms
<<Include>>
Select Customer
extension points
New customer
<<Include>>
List Customer
<<Extend>>
Create Customer
Register Check-in
<<Include>>
Select reservation
Calculate Payment
<<Include>>
<<Enable>>
Register Check-out
<<Include>>
Process Payment


uc [Manager]
Hotel Management System(HMS)
Manager
Check duplication
<<Include>>
Register
<<Include>>
Check data validation
<<Include>>
Send confirmation email
Update room's type
extension points
Invalid type
New type
<<Include>>
List room's type
<<Extend>>
Add new type
<<Extend>>
Delete existing type
Update room information
extension points
New room
Invalid room
<<Include>>
List room
<<Extend>>
Add new room
<<Extend>>
Delete existing room
<<Include>>
Generate Report
Forget password
<<Extend>>
Login
extension points
Forget password

2.4 Constraints

- Memory: The system will only have 10GB of data server capacity.
- Language requirements: Software must be written in English or Vietnamese.
- Budget constraints: Due to budget constraints, HMS is designed to be very simple and for basic functions only. The user interface will be very simple.
- Implementation constraints: The application should only be based on Java, JavaScript, HTML, CSS, frameworks like Spring Boot, React.
- Reliability requirements: The system must regularly synchronize with the backup server to avoid data loss when errors occur and can be recovered.

3. Specific requirements

3.1 External interfaces

3.1.1 User interfaces

- Login interface
  Hotel Management interface with functions of adding, editing, deleting rooms and room types.
- Reception interface with room and customer management functions.
- The system's user interface must be compatible with all types of web browsers like Google Chrome, Safari, CocCoc, Firefox and Internet Explorer …

3.1.2 Hardware Interfaces

- Server Side
  - Monitor: Resolution: 1024x768
  - Processor: Intel or AMD 2GHZ
  - RAM: 4GB
  - Disk Space: 10GB
- Client Side:
  - Monitor: Resolution: 1024x768
  - Processor: Intel or AMD 1GHZ
  - RAM: 512MB

    - Disk Space: 2GB

3.1.3 Software Interfaces

- Web Server: Apache Tomcat Server
- Database Server: MySQL Server
- Development End: J2EE, Java, JSP, Servlet, HTML, XML, JavaScript.
- Operating System: Windows, MacOS.

3.1.4 Communication Interfaces

The system will use HTTP/HTTPS to communicate over the Internet and to communicate within the internal network, the system will use TCP/IP protocol.

3.2 Functional Requirements

3.2.1 All user:

- Log in
    - The system shall verify the customer's email and password against the member database when logging in
    - After logging in, members shall be redirected to Home screen

3.2.2 Receptionist

- Book room
    - The system shall allow the Receptionist to check room availability
    - The system shall display prices and some information of all rooms
    - The system shall allow the Receptionist to confirm or cancel reservations
    - The system shall records booking information into the database
    - The system shall allow the Receptionist to update, add, and delete reservation information
    - The system shall provide customer portal access for receptionists to answer customer inquiries
- Payment
    - The system shall allow customers to pay bills online using a credit or debit card

    - The system shall provide a file pdf of the invoice to print for customers paying in cash.

3.2.3 Manager

- Register
    - Customers can register with their details
    - The system shall record the following customer information into the member database: Email, Password, Name, Address, Date of Birth.
    - The system shall send a verification message to email
- The system shall generate financial and customer reports
- The system shall allow managers full access to modifications to customer, booking and room information
- The system allows Managers to update, add, and delete room types and room information.

3.3 Performance requirements

- Static numerical requirements
    - Supports 2000 devices
    - Supports 200 concurrent users
    - Type of information processed: text, image
- Dynamic numerical requirements
    - Data in the database must be updated within 2 seconds
    - Query results must return within 3 seconds
    - UI loading time is no more than 1 second
    - Login authentication must be performed within 2 seconds
    - Answering customer inquiries must be done within 6 minutes

3.4 Design constraints:

The design constraints for a hotel management system are:

- ***Standards compliance*:** The system must comply with information security standards such as ISO/IEC 27001 or GDPR (General Data Protection Regulation) not only protects the organization's asset information but can also protect customer personal information.
- ***Hardware limitations:***

    - The system must operate on devices with enough resources to run web applications, including a strong enough CPU, large enough memory, and a stable network connection.
    - Internet bandwidth usage to be able to operate smoothly in limited internet environments.
- ***Legal regulations:***
    - Ensure compliance with legal regulations regarding the use of React and Spring Boots.
    - The system needs to comply with legal regulations related to the storage and processing of personal data, as well as regulations on information security in the hotel industry.
- ***Specific technical requirements:***
    - Programming language: The system must be developed using the JavaScript programming language. React will be used to develop the user interface (front-end). Spring Boot will be used for the backend to handle business logic and interact with the database.
    - Use SQL database to store information about rooms, reservations, payments, and other services.
- ***User interface (UI) requirements*****:** The user interface must be designed to be easy to use and end-user friendly, including flexibility across mobile and tablet devices.
- ***Performance and reliability*****:** The system must be able to handle multiple requests from users simultaneously and ensure high reliability without encountering waiting time or system failure issues.
- ***Scalability*****:** The system should be designed to be easily scalable as needed, including adding new features and support for multiple hotels and branches.

These constraints provide a framework for developing a hotel management system, ensuring that it complies with key requirements and meets the specific needs of the hotel business.

3.6 Software system attributes

3.6.1 Reliability

- The system accurately responds to common hotel management activities to meet end user goals through system tests with a high level of reliability.

3.6.2 Availability

- Have enough resources to deploy to achieve the specific task effectively without any trouble. Additionally, the system is flexible enough to provide space to add new features and handle them conveniently. Besides, the system also provides usage instructions for all user levels.

3.6.3 Security

- **Use encryption techniques:**
    - Apply encryption to databases, such as using data encryption for user passwords to protect them from unauthorized access.
    - TLS/SSL encryption techniques can be applied.
- **Keep specific log or historical data sets:**
    - Record all user activities, including login information, data changes, and other activities to check and monitor system performance.
    - Store access history to sensitive data to detect and correct untrustworthy behavior.
- **Assign functions to different modules:**
    - In particular, dividing the user authentication function into a separate module minimizes the risk of security errors in the entire system.
    - Place sensitive data management functions, such as password management and access management, into a standalone module for increased security.
- **Restrict communication between some areas of the program**
    - Set up brute force tools that can conduct communication between system components, minimizing the risk of attacks from malware.
    - Use prevention to control network storage and prevent unauthorized access to parts of the system.
- **Check data integrity for critical variables**
    - The system must perform comprehensive data checks against critical variables such as payment information to prevent illegal modifications or erroneous data.

3.6.4 Maintainability

- The system is designed specifically for separate and independent components, to easily replace, modify or upgrade one part without affecting other parts.

- Modules are designed so that their functions and responsibilities are clear and distinct. Interfaces between components or modules are simplified and standardized to minimize dependencies and enhance component performance when performing maintenance.
- Code source complexity is minimized for ease of understanding and editing.
- Algorithms and logic are developed clearly and efficiently to minimize risks and enhance maintainability.
- The system is designed for ease of testing, with independent and reusable test units.