Software Engineering Software Requirements Specification (SRS) Document

[HCK Hotels Management Software]

[14 FEB 2023]

[1.0]

By: [Christopher Jacobs, Hilda Ramirez, Kachif Adefalou]

[We agree to abide by the UNCG Academic Integrity Policy]

Table of Contents

1. Introduction

1.1. Purpose

The goal of the hotel management application is to help different types of hotel staff keep track of their responsibilities. The app will focus on increasing efficiency by providing essential information to employees by streamlining given tasks.

1.2. Document Conventions`

The purpose of this Software Requirements Document (SRD) is to outline the client-view and developer-view requirements for the hotel management application. The client-oriented requirements will provide a clear understanding of the system from the perspective of the end-users, including guests, employees, and managers. The requirements will describe the features and functionality required to meet the needs of these stakeholders, as well as the different types of users served by the system. The developer-oriented requirements will provide a detailed description of the software requirements from a developer's perspective. These requirements will include a description of the software architecture and the process of building the application.

1.3. Definitions, Acronyms, and Abbreviations

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager.
MySQL	Open-source relational database management system.
.HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
Spring Web	Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
Thymeleaf	A modern server-side Java template engine for our web environment. This is one of the dependencies of our system.
NetBeans	An integrated development environment (IDE) for Java. This is where our system will be created.
API	Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage.
HCK	Acronym for Hilda, Chris, Kachif. It is the company's name.

1.4. Intended Audience

Users: These are the people who will be using the software system once it's complete. They will be interested in the functional requirements,non-functional requirements, user interface design, and usability aspects of the software system.
Developers: These are the people who will be building the software system. They will be interested in the technical requirements, software design, and architecture.
Business analysts: These are the people who help define the requirements for the software system. They will be interested in the functional requirements, use cases, and business rules.
Quality assurance (QA) team: These are the people who will be testing the software system. They will be interested in the test cases, expected results, and acceptance criteria.
Project managers: These are the people who are responsible for managing the project. They will be interested in the project scope, timeline, budget, and risks.

1.5. Project Scope

The goal of the hotel management application is to streamline hotel operations, increase efficiency, and improve customer service. The software aims to provide an easy-to-use interface for all stakeholders, including guests, employees, and managers. By achieving these goals, the application aligns with the overall business goals of a hotel, which prioritize customer satisfaction and operational efficiency.

The benefits of the project to the hotel business include:

- Providing guests with a convenient way to manage their reservations, check-in and check-out, and request additional services. This will improve the overall guest experience and satisfaction.

- Automating many of the manual processes associated with hotel management, such as managing reservations, room assignments, and employee schedules. This will free up staff time and reduce errors, leading to increased operational efficiency.

- Providing managers with real-time data and insights into hotel operations, such as room occupancy rates, revenue, and guest feedback. This will enable managers to make more informed decisions and take corrective action when necessary.

- Facilitating communication between staff members and guests, ensuring that all requests and inquiries are handled in a timely and efficient manner. This will reduce the risk of miscommunication and improve overall service quality.

1.6. Technology Challenges

There aren’t any technological constraints to say, but this is the first project that the development team has worked on. The members do not have much experience.

1.7. References (APA)

Java Tutorial. (n.d.). https://www.w3schools.com/java/default.asp

Ntini, S. (2023). *Lecture Notes [Slides]*. Retrieved from University of North Carolina at Greensboro: https://uncg.instructure.com/courses/115037/files/folder/Lecture_Notes

2. General Description

2.1. Product Perspective

HCK Hotels Hotel Management Software was developed in response to the need for a more efficient and convenient way of managing HCK Hotel operations. The idea originated from the three joint owners of the hotel chain.

2.2. Product Features

The HCK Hotels management application is designed to provide an all-in-one solution for hotel guests, employees, and management. The main features of the application include:

- Different graphical user interface based on the role of the user (customer, employee, or manager)
- Customers can book rooms, open help desk tickets, and send messages to room service
- Employees can receive messages based on their role, such as room service and luggage carriers receiving messages from customers and being able to locate their rooms, and janitors and maintenance workers receiving tasks from management
- Managers have access to chat logs, handle customer complaints, give tasks to employees, and have access to hotel data (such as occupancy rates) to manage operations more efficiently.

2.3. User Class and Characteristics

1. Customers: This user class includes individuals who stay at the hotel as guests. They will use the application to book rooms, request services such as room service, and communicate with hotel staff.

2. Employees: This user class includes hotel staff such as room service attendants, janitors, and maintenance workers. They will use the application to receive messages, access work schedules, and receive tasks assigned to them by their supervisors.

3. Management: This user class includes hotel managers who are responsible for overseeing various operations in the hotel. They will use the application to manage employees, handle customer complaints, access hotel data, and assign tasks to employees.

The user classes will not have access to the parts of that app that are unnecessary. For example, an employee account will not be able to assign tasks, nor will they be able to book rooms.

2.4. Operating Environment

The hotel management application is designed to operate on the web and can be accessed from any device with an internet connection and a web browser. It will be compatible with a wide range of operating systems, web browsers, and devices. The application will require an active internet connection to function.

2.5. Constraints

- Security: Given the sensitive nature of hotel management information, the application will need to protect against data breaches and unauthorized access.
- Training and adoption: Ensuring that all employees can effectively use the new system may be a challenge, particularly for those who are not technologically literate.
- Time and budget constraints: Development and implementation of the new system will need to be completed within a specific timeline, which may require trade-offs between features and functionality.

2.6. Assumptions and Dependencies

Assumptions:

1. The hotel has a stable and reliable internet connection at all times.
2. The hotel has sufficient hardware infrastructure to support the software and its users.
3. The users have basic computer literacy skills and are familiar with using web-based applications.
4. The software will be developed within the designated timeline and budget.
5. The hotel will provide all necessary information for integration with the software.

External Dependencies:

1. The software may be dependent on third-party services for certain features such as payment processing or employee payrolls.
2. Compliance with applicable laws and regulations regarding data privacy and security.
3. Springboot, Apache, Tomcat, and My SQL.
4. The program will be developed within NetBeans.

3. Functional Requirements

3.1. Primary

FR0: The system will allow different users to manage hotel tasks for all the software requirements

FR1: The system will allow the “customer” user to book a room, open helpdesk tickets, and send messages for other users etc, managers, employees.

FR2: The system will allow the “employee” user to have access to work schedules, receive messages from other users like managers.

FR3: The system will allow the “manager” user to display real-time analytics and key performance indicators as well as managing employees, handling customer complaints, accessing hotel data, and assigning tasks to employees.

3.2. Secondary

- The system will provide tools to manage housekeeping tasks, such as cleaning schedules and maintenance requests.
- The system will provide tools to manage employee schedules, performance review or assign task

4. Technical Requirements

4.1. Operating System and Compatibility

HCK hotel management software will be compatible with the most widely used operating systems, including Windows, macOS, and Linux. By ensuring compatibility, the hotel management software can operate seamlessly and provide efficient services to hotel staff and customers.

4.2. Interface Requirements

4.2.1. User Interfaces

The user interfaces for the HCK Hotels management application will vary based on the role of the user. For hotel customers, the interface will include screens where they can input their reservation information and access their personal information. They will have access to their reservation details, the ability to make changes to their reservation, and the option to book additional services such as room service or spa appointments. They will also be able to open help desk tickets and send messages to room service. For employees, the interface will be tailored to their specific roles. Room service and luggage carriers will receive messages from customers and will be able to locate their rooms. Janitors and maintenance workers will receive tasks from management and be able to update the status of those tasks. Managers will have access to chat logs and will handle customer complaints. They will also give tasks to the employees, such as assigning a janitor to clean up a spill. Managers will also have access to hotel data, such as the occupancy rate or stock status.

The user interface will have a clean and modern design that is easy to navigate. All screens will include a header with the HCK Hotels logo, and a footer with links to important information such as the hotel's terms and conditions and contact information. Buttons and functions will be clearly labeled, with descriptive text that is easy to understand. Error messages will be displayed in a user-friendly manner, with clear instructions on how to rectify the issue.

4.2.2. Hardware Interfaces

The software will be accessible and be able to run from any device that can view or interact with traditional web pages. Access to the internet would be required to run this software and It will also be compatible with the hardware and other software applications that the hotel is currently using.

4.2.3. Communications Interfaces

It is essential to ensure that the software is compatible with the various communication interfaces used by the hotel to ensure smooth communication and data transfer between systems. This includes but is not limited to Property Management System, Point Of Sale interface, Customer Relationship Management as well as Housekeeping and Maintenance interface.

4.2.4. Software Interfaces

We will use Spring Boot and java to help build the frontend in addition to MySQL to help with the database management. Spring Boot and java will also be used to link and connect both the front and backend.

5. Non-Functional Requirements

5.1. Performance Requirements

[The performance requirements need to be specified for all the functional requirements.]

- NFR0(R): The local copy of the hotel database will consume less than 20 MB of memory
- NFR1(R): The system (including the local copy of the hotel database) will consume less than 50MB of memory
- NFR2(R): The novice user will be able to create and print a ticket in less than 5 minutes.
- NFR3(R): The expert user will be able to create and print a ticket in less than 1 minute.

5.2. Safety Requirements

- The software should be designed to protect the privacy of guests and staff. Any information collected should be used only for legitimate business purposes, and the software should comply with data privacy regulations.
- The software should have strong authentication and authorization mechanisms in place to ensure that only authorized users can access the system and specific data.
- The software should have granular access controls to ensure that users can only access the data and functionality they need to perform their job functions.

5.3. Security Requirements

- NFR1(R): The software should have robust monitoring systems in place to identify and prevent security breaches, system failures, and other potential safety hazards.
- NPR2(R): The software should be designed to ensure the physical security of hardware and data storage
- NFR3(R): The software should have role-based access controls to ensure that only authorized personnel can access sensitive information or perform specific tasks.

5.4. Software Quality Attributes

5.4.1. Availability

Available to only customers, and employees, and Management at HCK Hotels.

5.4.2.Correctness

Correctness will be limited by the users willingness to use the application, as well as their technological literacy.

5.4.3.Maintainability

As a fairly small program, the software will be easy to maintain. Many applications have bug report functionality, and this one will likely be the same.

5.4.4.Reusability

The software could be repurposed for other hotel chains, or even for single hotels.

5.4.5.Portability

The software is somewhat portable, but would require xampp to be installed, and possibly NetBeans.

5.5. Process Requirements

5.5.1. Development Process Used

We will be using Agile methodologies to develop the software. For example, the software will start out as a barebones initial version, that will be developed incrementally. The development method will also have feature planning and divided responsibilities.

5.5.2. Time Constraints

The prototype needs to be finished by Mar 20, 2023 . The final project must be completed by Apr 9, 2023 . These dates are the ***latest*** possible dates, and The HCK Hotel shareholders will expect early completion of the prototype and project.

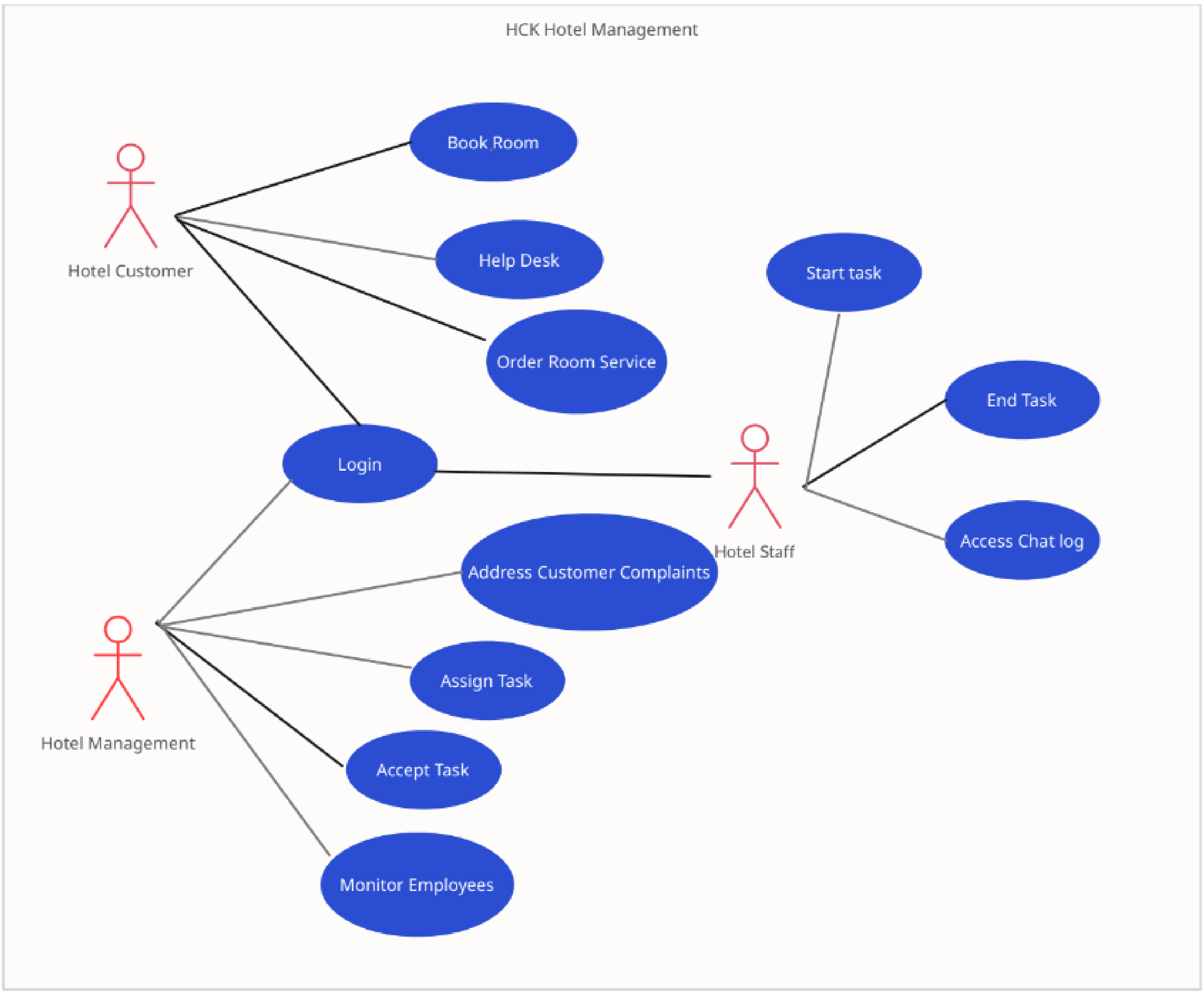
5.5.3. Cost and Delivery Date

We expect the program to require 100 hours of work at the most to complete. This will be divided among the three developers.

5.6. Other Requirements

TBD

5.7. Use-Case Model Diagram



Hotel Customer: Hilda Ramirez
Hotel Management: Christopher Jacobs
Hotel Staff: Kachif Adefalou

5.8. Use-Case Model Descriptions

5.8.1. Actor: Hotel Customer (Hilda Ramirez)

- **Book room**: Hotel guests can make reservations, modify reservations or cancel reservations.
- **Help desk:** Customers can send messages to hotel staff for any complaint, services or request they may have.
- **Order room service:** Customers can request services such as laundry, housekeeping, wake up call etc..

5.8.2. Actor: Hotel Management (Christopher Jacobs)

- **Address customer complaints**: The management will receive the complaint from the customer and resolve it by taking appropriate actions.
- **Assign task**: The management will create a task, assign it to an employee, and set the due date.
- **Monitor employees:** The management will view employee task progress, evaluate their performance, and provide feedback.

5.8.3. Actor: Hotel Staff (Kachif Adefalou)

- **Start task**: Hotel's staff will start working on tasks assigned by management.
- **End task**: Hotel's staff will update assigned tasks as completed once worked on.
- **Access chat log:** Hotel's staff will be able to access chat logs as a way to communicate with management for updates on tasks.

5.9. Use-Case Model Scenarios

5.9.1. Actor: Customer (Hilda Ramirez)

- **Use-Case Name**: Book room
    - **Initial Assumption**: The user has logged in to their account and can navigate to the room booking tab.
    - **Normal**: The user will select a room and book the reservation.
    - **What Can Go Wrong**: The user may have mistakenly entered the wrong information or booked the wrong room.
    - **Other Activities**: The user can open help desk tickets to rectify the issue.
    - **System State on Completion**: The room is booked successfully. It is updated, and the customer can see the room on their dashboard, along with a message in their inbox containing the confirmation number for the user after the transaction is completed.
- **Use-Case Name**: Help desk
    - **Initial Assumption**: User has logged into their account and can navigate to the help desk tab.
    - **Normal**: The user will write and submit a help ticket.
    - **What Can Go Wrong**: The user may make a typo writing the ticket
    - **Other Activities**: The user can view and edit tickets they have submitted
    - **System State on Completion**: The ticket is submitted successfully. It is stored, and sent to the database where a manager can respond to the help ticket.
- **Use-Case Name**: Order room service
    - **Initial Assumption**: User has logged into their account and can navigate to the Special Services tab
    - **Normal**: The user will select room service and order the service.
    - **What Can Go Wrong**: The user might select the wrong service package or may order by accident/
    - **Other Activities**: The user can view previous orders and send requests to cancel orders.
    - **System State on Completion**: The order is placed successfully. It is updated and shows the service. Management can see this on their Service Fulfillment tab.

5.9.2. Actor: Management (Christopher Jacobs)

- **Use-Case Name**: Address customer complaints
  - **Initial Assumption**: The manager has logged into their account and can view their dashboard and navigate to the customer complaints tab.
  - **Normal**: The manager will open a complaint ticket and deal with the complaint.
  - **What Can Go Wrong**: The manager could resolve a complaint prematurely by accident.
  - **Other Activities**: The manager can unresolve a complaint from the resolved complaints screen.
  - **System State on Completion**: The complaint is resolved successfully. It is logged.
- **Use-Case Name**: Assign task
  - **Initial Assumption**: The manager has logged into their account and can view their dashboard and navigate to the Tasks tab.
  - **Normal**: The manager will update a particular task and assign it
    to one of the registered employees.
  - **What Can Go Wrong**: The manager can assign a task to the wrong employee.
  - **Other Activities**: A task can be reassigned using a re-assign drop-down box to select the employee.
  - **System State on Completion**: The task is assigned successfully. It is updated and shows the assignee. The assignee can also see this task on their dashboard in the tasks tab.
- **Use-Case Name**: Monitor employees
  - **Initial Assumption**: The manager has logged into their account and can view their dashboard and navigate to the Employees tab.
  - **Normal**: The manager will read employee task history, which employees are free, and other relevant live information. They might also chat with the employees.
  - **What Can Go Wrong**: Some employees may have done more tasks while others will have done less tasks.
  - **Other Activities**: A manager can sort the employee list by tasks finished today, or other relevant information.
  - **System State on Completion**: The manager is finished chatting, or has absorbed the information from the Employees tab. Chats are logged, and any chats that weren't closed are added to the Chat tab to be continued later.

5.9.3. Actor: Staff (Kachif Adefalou)

- **Use-Case Name**: Start task
    - **Initial Assumption**: The staff member has logged in and notice a task has been assigned
    - **Normal**: Staff member will accept the task as a way to acknowledge to management that it's been received
    - **What Can Go Wrong**: network connectivity could prevent real time view of assigned task
    - **Other Activities**: Try switching to the hotel's wifi for better internet connection.
    - **System State on Completion**: Task was completed and management was informed of the status of the task
- **Use-Case Name**: End task
    - **Initial Assumption**: Task assigned by management has been completed
    - **Normal**: Staff member would update the status of the assigned task to completed
    - **What Can Go Wrong**: network connectivity could prevent real time view of change of status made by staff member.
    - **Other Activities**: Try switching to the hotel's wifi for better internet connection.
    - **System State on Completion**: Task status has been updated to reflect changes made.

6. Design Documents

6.1. Software Architecture

6.2. High-Level Database Schema

6.3. Software Design

6.3.1. State Machine Diagram: Actor Name (Responsible Team Member)

6.3.2. State Machine Diagram: Actor Name (Responsible Team Member)

6.3.3. State Machine Diagram: Actor Name (Responsible Team Member)

6.4. UML Class Diagram

7. Scenario

7.1. Brief Written Scenario with Screenshots