SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Hotel Automation System (HAS)

Prepared By –
Ankesh Anand (11MA20052)

Table of Contents

1 Introduction

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description

of the requirements for the Hotel Automation System (HAS). This SRS will allow for a complete understanding of what is to be expected of the HAS to be constructed. The clear understanding of the HAS and its' functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the HAS can be designed, constructed, and finally tested.

This SRS will be used by the software engineers constructing the HAS and the hotel end users. The software engineers will use the SRS to fully understand the expectations of this HAS to construct the appropriate software. The hotel end users will be able to use this SRS as a "test" to see if the software engineers will be constructing the system to their expectations. If it is not to their expectations the end users can specify how it is not to their liking and the software engineers will change the SRS to fit the end users' needs.

1.2 Scope

The software product to be produced is a Hotel Automation System which will automate the major hotel operations. The first subsystem is a Reservation and Booking System to keep track of reservations and room availability. The second subsystem is the Catering System that charges the corresponding guest. The

third subsystem is a General Management Services and Automated Tasks System which generates reports to audit all hotel operations and allows modification of subsystem information. These three subsystems' functionality will be described in detail in section 2-Overall Description.

There are two end users for the HAS. The end users are the hotel staff (receptionist) and hotel managers(catering service manager and manager). Both user types can access the Reservation and Booking System and the Catering System.

The Hotel Automation System's objectives is to provide a system to manage a hotel that contains certain number of rooms. Without automation the management of the hotel has become an unwieldy task. The end users' day-to-day jobs of managing a hotel will be simplified by a considerable amount through the automated system. The system will be able to handle many services to take care of all customers in a quick manner. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

1.3 Definitions, Acronyms, and Abbreviations.

SRS – Software Requirements Specification
HAS – Hotel Management System

End users – The people who will be actually using the system
Guest – Customer
IEEE – Institute of Electrical and Electronic Engineers

1.4 Refrences

Refrence for the document is "IEEE Guide for SRS" and "IEEE SRS template Version 1" .

1.5 Document Convention

The format of this SRS is simple. Bold face and indentation is used on general topics and or specific points of interest. The remainder of the document will be written using the font, Arial and font size 12.

1.6 Intended Audience and Document Overview

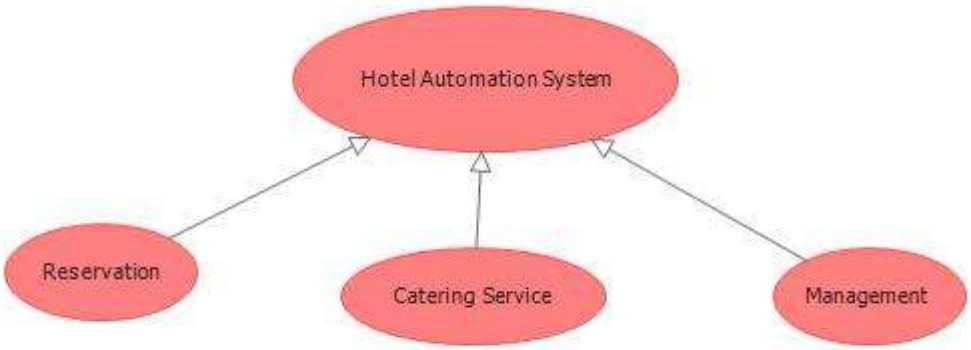
The SRS is intender for Hotel staff and Professors. The SRS is organized into two main sections. The first is The Overall Description and the second is the Specific Requirements. The Overall Description will describe the requirements of the HAS from a general high level perspective. The Specific Requirements section will describe in detail the requirements of the system.

2 The Overall Description

Describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead it provides a background for those requirements and makes them easier to understand.

2.1 Product Perspective

The HAS is an independent stand–alone system. It is totally self contained.It provides an interface to automate the hotel administration as it becomes very tedious task to manage room’s database and bills and customize rates of room on the basis of average occupancy monthly. So this software provides an ease to automate the hotel system.



2.2 System Environment

The software interacts exclusively with the Receptionists,Catering Service Manager and Manager of the hotel. The manager has acess to all subsytems of the system. The software uses QSql for maintaining the database. The software runs on high-end PC and on any operating system that supports and compile C++ with qt framework such as linux,windows.

2.3 Product Functions

Reservation and Booking System

- Allows for typing in guest information
- Has a room tariff provided by Manager to System.
- When reservation is completed, a unique token is assigned to the customer.
- When a guest checks in, the room number will be changed to occupied in the database.
- Checks whether checked in guest exist in database or not and if not then database will be updated.
- When a guest checks out total bill and the amount owed is displayed and frequency of guest is updated
- If guest is frequent then assign id to the customer.

- Records that room is available.
- Provide discount to guest on behalf of customer id (if any)
- Records payment

Catering Service System

- Tracks all meals purchased
- Charges the corresponding customer token number

General Management Services and Automated Tasks System

- Reports generated to audit hotel occupancy and room revenue of corresponding month
- Allows percentile change in room tariffs depending on the occupancy
- Allows to change the frequency to be a frequent user.

2.4 User Classes and Characteristics

2.4.1 User Classes

- Receptionist
- Hotel Catering Service Manager
- Hotel Manager

2.4.2 User Characteristics

Educational level of HAS computer software – Low
Experience of HAS software – None
Technical Expertise – Little

2.5 Design and Implementation Constraints

The operating environment must have C++ and qt runtime as all coding is done in C++ with qt framework **.** The Hotel Management System shall be a

stand-alone system running in an environment having C++ complier. The system shall be developed using C++ and QSql as database.

2.6 Assumptions and Dependencies

- Payment is supposed to be done through cash
- No interaction is required between the guest and the software.
- All details on reservation is supposed to be provided by the receptionist
- Discount on the basis of customer id is provided by users' choice (receptionist or manager) individually
- Number of customer tokens are assigned on the basis of number of reservation not the number of occupants or rooms.
- Manager decides that what frequency per month is fair enough to recognize a user as frequent user and assign customer id.

3 Specific Requirements

This section contains all the software requirements at a level of detail, that when combined with the class diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

3.1 External Interfaces Requirements

The Hotel Management System will use the standard input/output devices for a personal computer. This includes the following:

- Keyboard
- Mouse
- Monitor
- Printer

3.1.1 User Interfaces

The User Interface Screens are described in table 1.

Table 1: Hotel Management User Interface Screens

Screen Name	Description
Reservation	Make reservation, cancel reservation, modify reservation, accept payment,allot room and customer token
Check-in	Check-in guest (with or without a reservation), Identify Reservation, accept advance payment, request for Customer Id (If Available), generate customer id (if applicable).
Checkout	Checkout guest, generate bill including Catering services, display bill, accept payment for room and catering, provide discount (if customer id available)
Hotel Catering Service	Create order, update catering bill

Screen Name	Description
Manager	View average occupancy, change rates,change frequency (required for being a frequent user)

3.1.2 Hardware Interfaces

The system shall run on any system that supports C++ and qt runtime.

3.1.3 Software Interfaces

All databases for the HAS will be configured using QSql. These databases include hotel rooms and customers information. These can be modified by the end users. The room database will include the room numbers and if they are vacant or occupied. The customers information database will contain all the information of the customer such as first name, lastname,customer token, number of occupants, assigned room, type of room (AC or Non-AC or Single or Double beded), room tariff, expected check in date and time, actual check in date and time, expected check out date and time, amount owed by guest, and customer id.

3.1.4 Communication Interfaces

The system shall be a standalone product that does not require any communication interfaces.

3.2 Functional Requirements

Functional requirements define the fundamental actions that system must perform. The functional requirements for the system are divided into three main categories or sub-sytems, Reservation/Booking, Food, and Management.

3.2.1 Reservation/Booking

1.1. The system shall record reservations.

1.2. The system shall record the customer's first name.

1.3. The system shall record the customer's last name.

1.4. The system shall record the number of occupants.

1.5. The system shall record expected check-in and check-out date and time.

1.6. The system shall record the type of room required (Single or double, Ac or Non-Ac)

1.7. The system shall check for availability.

1.8. If reservation is not available, system should generate an apology message.

1.9. The system shall record the room number reserved.

1.10. The system shall display the room rate.

1.11. The system shall generate a unique customer token for each reservation.

1.12. The system shall automatically cancel reservation if user don't check in on the expected check-in date.

1.13. The system shall check-in customers

1.14. The system shall record the actual check-in date and time. 1.15. The system shall record the room occupied.

1.16. The system shall count the hotel occupany in current month. 1.17. The system shall record amount advance paid.

1.18. The system shall check whether corresponding guest is recorded in database.

1.19. If guest is recorded in database then system shall update frequency, else add customer entry in database.

1.20. The system shall assign a default generated Customer Id to frequent customers.

1.21. The system shall checkout customers.

1.21.1. The system shall display the bill of the customer.

1.21.2. The system shall display the amount owed by the customer.

1.21.3. To retrieve customer information the customer token shall be used

1.21.4. The system shall record that the room is empty.

1.21.5. The system shall record the payment.

1.21.6. The system shall allow to provide discount on behalf of customer id if customer is frquent.

3.2.2 Catering Service

1.22. The system shall track all meals purchased in the hotel.

1.23. The system shall record date and time of purchase.

1.24. The system shall bill the corresponding customer token.

3.2.3 Management

1.25. The system shall display the average hotel occupancy for a specified month.

1.26. The system shall allow to change the room tariff by a percentage.

1.27. The system shall allow to change the threshold frequency to recognize a guest as a frequent user.

1.28. The system shall update the average occupancy of current month by the end of the every month.

3.3 Behavioural Requirements

Use case view

U1. ***Reservation***

1. Resceptionist provides the customer information as name of customer,number of occupancy,type of rooms required,number of rooms required ,advance paid amount and expected time of stay .
2. Check the provided data is valid.
3. Check for availability of rooms.

4. If room is available,provide guest a unique customer token and allot room numbers and update corresponding rooms reserved.
5. If room is not available display an apology message.

U2***. Check Availabilty***

1. Check whether requirement is valid or not.
2. Check whether availability fulfills the requirement.
3. If not then show the availability of rooms as how many rooms are available of the required type.

U3***. Check in***

1. Identify the reservation.
2. Provide corresponding reserved rooms.
3. Update corresponding rooms' status as occupied.
4. Check if corresponding customer exist in database or not.
5. If it exists then update the frequency of visits of the customer.
6. If it does not exist then add the customer into the database. .
7. If frequency is greater than threshold then issue id to customer.
8. Occupancy of this particular month will be updated.
9. Create new bill.

U4. ***Identify Reservation***

1. Identify the reservation on the basis of customer token.
2. If reservation is not identified then allow to make a new reservation.

U5. ***Issue ID***

1. If the visiting frequency is greater than threshold frequency, issue Customer Id to the Guest.

U6. ***Check out***

1. Generate Bill of the corresponding guest.

2. Check the customer id and if it exists provide discount if applicable.
3. Receive the payment.
4. Update status corresponding rooms as available.

U7. ***Generate Bill***

1. Generate the whole bill including the catering bill.
2. Show the total amount and amount owed by the guest as advance paid amount has to be reduced.

U8. ***Provide Discount***

1. Provide special discount on their bills by the hotel staffs choice.

U9. ***Catering Order***

1. Catering manager should provide quantity of meal, type of meal , date and time of meal and customer token.
2. Update the catering bill of corresponding customer token.

U10. ***Allot room and Token***

1. Available rooms and unique token will be allotted to the guest.
2. Status of the allotted tooms will be changed to reserved.

U11. ***Revise Tariff***

1. Manager could view the average occupancy on the basis of the months.
2. Manager could update the tariff of the rooms.

U12. ***Change Frequency***

1. Manager could change the threshold frequency required to consider a particular user as a frequent one.

3.4 Non-Functional Requirements

Non - Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

3.4.1 Performance Requirements

Performance requirements define acceptable response times for system functionality.

- The load time for user interface screens shall take no longer than two seconds.
- Queries shall return results within 8-10 seconds.
- System shall recover at most in 10 minutes by restarting.

3.4.2 Logical Database Requirements

The logical database requirements include the retention of the following data elements. This list is not a complete list and is designed as a starting point for development.

Booking/Reservation System

- Guest first name
- Guest last name
- Guest address
- Customer token
- Number of occupants
- Alloted room
- Current room tariff
- Check-in date
- Check-in time
- Current Occupancy

- Guest's frequency of visting
- Customer id of checking in Guest
- Expected check-out date
- Expected check-out time
- Actual check-out date
- Actual check-out time
- Advance Paid
- Total Bill

Catering Services

- Meal
- Meal type
- Meal item
- Meal date and time
- Meal Bill (to Customer Token)

Management

- Average Occupancy (for each month)
- Room tariff
- Frequency (to be frequent user)

3.4.3 Security Requirements

Hotel Staff will have access to the Reservation/Booking.Hotel Catering Service Manager will have access to catering subsystem. Managers will have access to the Management subsystem as well as the Reservation/Booking and Catering

subsystem. Access to the Management subsystems will be protected by a Manager log-in screen that requires a password.

3.4.4 Standards Compliance

The graphical user interface shall have a consistent look and feel.

3.4.5 Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery.

3.4.6 Availability

The system shall be available during hotel operating hours.

3.4.7 Maintainability

The Hotel Management System is being developed in C++. C++ is an object oriented programming language and shall be easy to maintain.

3.4.8 Portability

The Hotel Management System shall run in any environment that conatains C++ complier.

4 Supporting Information

A system context diagram as well as use cases and use case descriptions have been developed in separate documents.