**Software Requirements**

**Specification**

**For**

**Hospital Management System for**

**Apollo Hospitals**

**Version 1.0**

**Prepared by Sanjay P**

**National Institute of Technology, Karnataka**

**January 2, 2018**

|  |  |
| --- | --- |
| **Table of Contents**  **Table of Contents**  **Revision History** | **ii iii** |
| **1. Introduction** | **1** |
| 1.1 Purpose | 1 |
| 1.2 Document Conventions | 1 |
| 1.3 Intended Audience and Reading Suggestions | 1 |
| 1.4 Product Scope | 1 |
| 1.5 References | 1 |
| **2. Overall Description** | **2** |
| 2.1 Product Perspective | 2 |
| 2.2 Product Functions | 2 |
| 2.3 User Classes and Characteristics | 2 |
| 2.4 Operating Environment | 2 |
| 2.5 Design and Implementation Constraints | 2 |
| 2.6 User Documentation | 2 |
| 2.7 Assumptions and Dependencies | 3 |
| **3. External Interface Requirements** | **3** |
| 3.1 User Interfaces | 3 |
| 3.2 Hardware Interfaces | 3 |
| 3.3 Software Interfaces | 3 |
| 3.4 Communications Interfaces | 3 |
| **4. System Features** | **4** |
| 4.1 System Feature 1 | 4 |
| 4.2 System Feature 2 (and so on) | 8 |
| **5. Other Nonfunctional Requirements** | **8** |
| 5.1 Performance Requirements | 8 |
| 5.2 Safety Requirements | 8 |
| 5.3 Security Requirements | 8 |
| 5.4 Software Quality Attributes | 8 |
| 5.5 Business Rules | 8 |
| **6. Other Requirements** | **9** |
| **Appendix A: Glossary** | **9** |
| **Appendix B: Analysis Models** | **9** |
| **Appendix C: To Be Determined List** | **9** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Sanjay P | 02-01-2018 | Creation of the SRS for *Apollo Hospitals.* | 1.0 |

**1. Introduction**

**1.1 Purpose**

This Software Requirement Specification (SRS) outlines the steps taken in the deployment of an Hospital Management System (HMS); describes the requirements involved in the development of HMS for Apollo Hospitals. The main purpose of this software would be to make it convenient for the customers (end users) to check availability of Doctors, check the status of Patients and check the treatment details.

**1.2 Document Conventions**

Main topics are bolded in heading (whole numbered) followed by subtopics (decimal numbered) and bullets. All acronyms have been introduced with their full names, followed by the acronym in the parenthesis. In the glossary of Appendix A, all the acronyms used within this document are outlined in the alphabetical order.

**1.3 Intended Audience and Reading Suggestions**

This document is written in response to outlining first phase of the HMS and is intended for use by all the developers, designers, programmers, testers and documentation writers involved in the development of the HMS. This document is best read from beginning to the end to fully grasp the implementation and development details. However, the table of contents can be used to hone in on specific areas of interest to specific users. This document however, is not intended for the use by marketing staff.

**1.4 Product Scope**

This software (HMS) being developed will provide the following functionality:

With the correct login and password, the user will be directed to the Main menu else it will exit.

* The HMS can be used to check the details of the Doctor with the input of the Doctor identity.
* The HMS can be used to check the details of the Patient with the input of the Patient identity.
* The treatment details can be checked with the input of Doctor Id, Patient Id and Illness.

The HMS will be accessible by users with the access to authenticated computers with the working application installed.

**1.5 References**

[1] The complete e-commerce book, Book by Janice Reynolds. Retrieved from https://www.amazon.in/Complete-Commerce-Book-Successful-Web.../1578203120

[2] Hospital Management system SRS. Retrieved 2017, December 29 from

https://www.slideshare.net/kataria55/srs-for-hospital-management-system

[3] SRM University. (2012 - 2013). CS0411 - Software Engineering Lab - Laboratory Manual. Retrieved from [http://www.srmuniv.ac.in/sites/default/files/files/SOFTWARE%20Engineering%20LAB-CS0411.p](http://www.srmuniv.ac.in/sites/default/files/files/SOFTWARE Engineering LAB-CS0411.p)

df

[4] Wiegers K. E. (1999). IEEE Requirements Specification Template. Retrieved from https://web.cs.dal.ca/~hawkey/3130/srs\_template-ieee.doc

**2. Overall Description**

**2.1 Product Perspective**

The HMS is an independent application and is a self-contained project; eliminating Site Adaptation

Requirements, i.e., no modifications are required to adapt to a particular installation. The HMS will work with the following hardware interfaces:

● *Hard disk*: The database connectivity requires a hardware configuration with a fast database system running on high rpm hard disk permitting complete data redundancy and backup systems to support the primary goal of reliability.

● The system must interface with the standard output device, keyboard and mouse to interact with this software.

The HMS will work with the following software interfaces (both system and user interfaces (UI)):

● *Back End*: MySQL Ver 14.14 Distrib 5.7.20, for Linux (x86\_64) using EditLine wrapper.

● *Front End*: NetBeans 8.2.

The HMS will stage the following operations:

● The user mode enables the end users to do the end user operations like checking the details of Doctors and Patients and treatments.

The HMS must be user-friendly and interactivity must be ensured; functionality provided by the system like displaying error messages should adapt itself to the different users of the software.

**2.2 Product Functions**

The HMS allows users to perform a series of actions to aid in the *Apollo Hospitals Management* process. These functions include the following:

*● Checking Doctors details:* The user must enter the Doctor’s Id to retrieve information about the particular Doctor or one must enter the doctor’s details to update it to the Database.

● *Checking Patients details*: The user must enter the Patient’s Id to retrieve information about the particular Patient or one must enter the patient’s details to update it to the Database.

● *Checking Treatment details*: The user must enter the Doctor id, Patient id and the illness to retrieve details about the treatment.

**2.3 User Classes and Characteristics**

Passengers with the access to the installed application are the target users of the HMS and can utilize the benefits of the HMS application. The characteristics of these users can be described as follows:

● The product is absolutely user friendly, so the intended users can be naive users.

● Need not have specific knowledge as to what the internal operation of the system is. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user.

● May or may not be technically proficient; any person who knows to use the mouse and the keyboard can successfully use this product.

● May or may not have prior experience with health related applications.

● May or may not be educated and trained in the domain.

**2.4 Operating Environment**

This HMS application has chosen the following elements of the environment for the *Apollo Hospitals* reservation application:

● Linux or Windows based Operating System (OS) will be used to house the application.

● The language chosen for the deployment of the application would be Java (javac 1.8.0\_151).

● The UI will be handled by NetBeans, a Java Integrated Development Environment (IDE).

● The database which holds Doctor, Patient and Treatment details is deployed in MySQL.

**2.5 Design and Implementation Constraints**

The following are the constraints laid before utilizing the HMS application in full scale:

● It is the responsibility of the potential user (Doctor or Patient) to ensure that he retrieves information from authenticated computers or servers hosting the application.

● *Reliability requirements*: Data redundancy and use of special characters must be avoided.

● *Safety and Security considerations*: The user must always exit the application normally.

*● Programming language requirements*: The developers and programmers must have experience with Java and MySQL.

● At the time of reservation, each user is provided a Id (Doctor id or Patient id) that must be used for further operation like getting the details of the particular person. Hence the user is required to remember or store this number carefully.

● The necessity of providing options to customer to choose their Doctor for an appointment or getting details printed can be delayed until future versions of the software are developed.

**2.6 User Documentation**

The *Bon Voyage* reservation application will use very intuitive design, leaving the following considerations:

● A Portable Document Format (PDF) User manual will be provided by the documentation writers to the *Apollo Hospitals Management* with an overview of application and its functionalities.

● It is the responsibility of the *Apollo Hospitals Management* to provide their users with instructions on the use of the application from a users’ viewpoint.

● The application will hold a *help* option, which can be accessed on the main page of the application. When the user clicks on this option, a help center article provided by *Apollo Hospitals Management* will be displayed.

**2.7 Assumptions and Dependencies**

The following assumptions have been made while deploying the HMS:

● The application is used only by confirmed Doctors and potential patients of *Apollo Hospitals Management System.*

● Users of the application are knowledgeable about using applications on a computer system.

● It is the responsibility of *Apollo Hospitals Management* to keep the application updated with the most current content.

**3. External Interface Requirements**

**3.1 User Interfaces**

The HMS application has Five main interfaces with the help page excluded, that the users will view. Initial configurations of these interfaces are provided below. The interfaces must be easy to understand and the user interface typically includes the following:

● *Screen Formats*: The introductory screen (Login page) will be the first to be displayed which will allow the users to Login into the application with the Username and a Password (help and exit options included).

● *Window Formats*: When the user chooses one of the available options, the corresponding page associated with the choice will be displayed in a new window, which ensures multiple windows (single instance of a window) to be visible on the screen.

● *Data Formats*: The data and information presented by and to the user will be alphanumeric.

● *End Messages*: Appropriate error messages will be displayed, guiding the user to get out of such scenarios.

The visual mockups (prototypes) of the four main interfaces of the HMS application including the Login page, Main Menu page, Doctor details page, Patient details and Treatment details page will be provided in the upcoming versions of this SRS.

**3.2 Hardware Interfaces**

The system must basically support certain input and output devices, whose descriptions are described as in Table 1.

**Table 1.** List of Hardware Interfaces

Name of the Interface Description or Purpose

Keyboard and Mouse To accept data from the user like name, address etc. Processor Number 1 or 2 per node

Processor Model i5 6200U Processor Speed 2.5 GHz

Processor Cores 8, 6 or 4

Hard disk Fast and reliable database connectivity

Memory 8GB DDR3 @ 2.8GHz

**3.3 Software Interfaces**

The following software interfaces are used in the deployment of the HMS:

*MySQL Database Management System (DBMS)*:

**Table 2.** List of Software Interfaces with respect to MySQL DBMS Name of the Interface Description or Purpose

Type System Dynamic, Static Architecture Relational Model Software License Proprietary

Operating System Windows, Mac OS X, Linux, Unix

Max DB Size Unlimited

Max Table Size 4 GB Max Row Size 8 KB Max Column Name Size 30

Max Columns per Row 1000

The front end of the application is developed using Netbeans 8.2 IDE with Java as the development language and Windows 10 is used as the base operating system.

**3.4 Communications Interfaces**

Not applicable as the application being developed is a stand-alone application that can be accessed only via authorized computers. In the future, we may maintain a server that hosts the application, where in which, every client system would require a secure Local Area Network (LAN) connection via Secure Sockets Layer (SSL) for security, to communicate with the server.

**4. System Features**

This section outlines the low level details (implementation) of every system function of the HMS application system. It sets the priorities of each function so that the developers and the investors would have a lucid picture of what will be accomplished first and what functions will be available to review first.

**4.1 Search for Doctor**

*4.1.1 Description and Priority*

Upon successful Login into the *Apollo* Hospital Management Application, user can select *Check Doctor’s details from* the application’s Main menu page. This allows users to search for available Doctors, get all their details and book appointments with them. This function is one of the main aspects to Hospital management, only not so crucial, so its feature is of medium priority.

*4.1.2 Stimulus/Response Sequences*

After a user selects *Check Doctor’s details* from the *Apollo* Hospital Management Main menu page, the user can enter the details of the newly registering Doctor and stores it into the database. If no doctors or patients available with the provided input, an appropriate message will be displayed to the user.

*4.1.3 Functional Requirements*

*SD-1:* A search form will be provided to the user to check details of doctors.

*SD-2:* A submit button submits the entered data of a newly registering doctor.

*SD-3:* Data is validated for completeness, accurate dates, and valid data options.

*SD-4:* If the data is complete and valid, the hospital management application will display the requested details retrieved from the database.

*SD-5:* If no information is available in the database for the requested input, then an appropriate message will be displayed to the user.

**4.2 Search for Patients**

*4.2.1 Description and Priority*

Upon successful Login into the *Apollo* Hospital Management Application, user can select *Check Patient’s details from* the application’s Main menu page. This allows users to search for admitted or registered patients. This function is one of the main aspects to Hospital management, only not so crucial, so its feature is of medium priority.

*4.2.2 Stimulus/Response Sequences*

After a user selects *Check Doctor’s details* from the *Apollo* Hospital Management Main menu page, the user can enter the details of the newly registering Doctor and stores it into the database. If no doctors or patients available with the provided input, an appropriate message will be displayed to the user.

*4.2.3 Functional Requirements*

*SP-1:* A search form will be provided to the user to check details of Patients.

*SP-2:* A submit button submits the entered data of a newly registering Patient.

*SP-3:* Data is validated for completeness, accurate dates, and valid data options.

*SP-4:* If the input data is complete and valid, the hospital management application will display the requested details retrieved from the database.

*SP-5:* If no information is available in the database for the requested input, then an appropriate message will be displayed to the user.

**4.3 Search for Treatment Details**

*4.1.1 Description and Priority*

Upon successful Login into the *Apollo* Hospital Management Application, user can select *Check Treatment details from* the application’s Main menu page. This allows users to search for treatment details with the input of Doctor id and Patient id. This function is one of the main aspects to Hospital management, only not so crucial, so its feature is of medium priority.

*4.1.2 Stimulus/Response Sequences*

After a user selects *Check Treatment details* from the *Apollo* Hospital Management Main menu page, the user can check the details of the patient upon the entry of Doctor id and Patient id. If no doctors or patients available with the provided input, an appropriate message will be displayed to the user.

*4.1.3 Functional Requirements*

*ST-1:* A search form will be provided to the user to check details of Treatment.

*ST-2:* A Add button updates the newly entered Treatment details.

*ST-3:* Data is validated for completeness, accurate dates, and valid data options.

*ST-4:* If the data is complete and valid, the hospital management application will display the requested details retrieved from the database.

*ST-5:* If no information is available in the database for the requested input, then an appropriate message will be displayed to the user.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The following are the performance requirements of the *Apollo Hospital* management application:

**Table 3.** Performance Requirements

Requirement Description

System Availability Unscheduled downtime and maintenance must be addressed and resolved in as little time as possible.

System Responsiveness The system must be responsive and interactive.

System Response The system must take lesser time to respond to any request and return data from the Database.

**5.2 Safety Requirements**

The following are the safety requirements of the *Apollo Hospital Management* application:

**Table 4.** Safety Requirements

Requirement Description

System Accessibility The application responds to the NetBeans settings for font styles and sizes which include default settings and special accommodations.

Soft Color Scheme The application uses color scheme that does not use high contrast colors for reduced eye strain.

Lost or Low Battery The application closes when the computer’s battery is depreciated, which requires user to restart the application on the battery recovery.

**5.3 Security Requirements**

The following are the security requirements of the *Apollo Hospital* reservation application:

**Table 5.** Security Requirements

Requirement Description

Cache The application will not store any personal data on the device

Memory Protection The application performs garbage collection on closing it.

**5.4 Software Quality Attributes**

The following are the software quality attributes for the *Apollo Hospital* reservation application:

**Table 6.** Software Quality Attributes

Requirement Description

Font Family The application uses NetBeans default fonts.

Font Size The application uses default 12-point to 24-point font sizes. Usability The application’s usability should be high; ease of use must be

ensured.

Help Icon Help icon on the welcome page can be used to access the instruction manual for the application.

Updates Each update of the application includes versioning for troubleshooting and traceability purposes.

Reusability The application is deployed using Object Oriented Programming

(OOP), which allows the design features to be reused.

**5.5 Business Rules**

The following business rules can be implied in the deployment of *Apollo Hospital* ARS:

**Table 7.** Business Rules

Individual Role Access or Circumstance

Database Administrators Access to the database used in the application. Java Programmers Access to the back-end of the application.

End Users Access to the UI or front-end of the application.

**Appendix A: Glossary**

Terms and acronyms which may not be familiar to all readers are provided in this section. Definitions provided are limited to their scope within this document. All terms are listed in alphabetical order for easy reference.

**Table 8.** Glossary of Terms used

Term Definition

HMS- Hospital Management System An application that allows users to search/enter for available doctors, make an appointment, view/enter patient and treatment details..

DBMS- Database Management System Management of a database which is an organized collection of data.

IDE- Integrated Development Environment It is a software application that provides comprehensive facilities to computer programmers for software development.

LAN- Local Area Network A local area network is a computer network that interconnects computers within a limited area such as a residence, school, laboratory, university campus or office building.

OOP- Object Oriented Programming Object-oriented programming is a programming paradigm based on the concept of *objects*.

OS- Operating System The software that supports the basic functions of the device on which it is installed.

PDF- Portable Document Format A file format that provides an electronic image of text and graphics that can be transmitted.

RTM- Requirements Traceability Matrix A table used to track the requirements of the

Apollo Hospital Management System.

SRS- Software Requirements Specification A description of software to be developed.

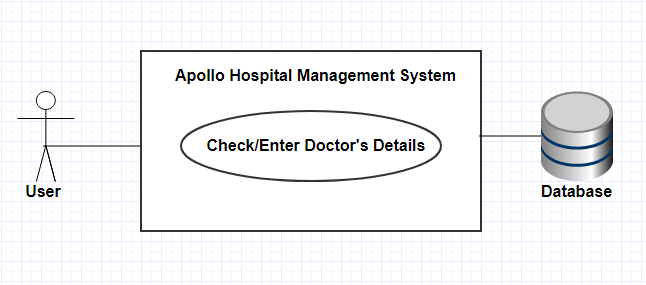
SSL- Secure Sockets Layer It is a standard security protocol for establishing encrypted links in an online communication.

UI- User Interface The space where interactions between humans and machines occur.

UTN- Unique Ticket Number Unique number presented while reserving tickets.

**Appendix B: Analysis Models**

**UC-1: Check Doctor’s details**



**Fig. 6.** Use-case diagram to check doctor’s details

The above use case diagram has been elucidated in Table 9.

**Table 9.** UC-1 to check doctor’s details

Term Explanation

Actors User or Doctor, *Apollo Hospital* Database.

Description The user searches for available doctors or enters the details of a doctor.

Trigger The user selecting “*Doctor’s details”* optionin the main menu page.

Flow The user gets directed to the Main menu page upon successful login..

The user enters Doctor Identity. The most relevant *Apollo Hospitals’* Doctor is presented to the user. If no doctors are available then an appropriate error message is displayed to the user.

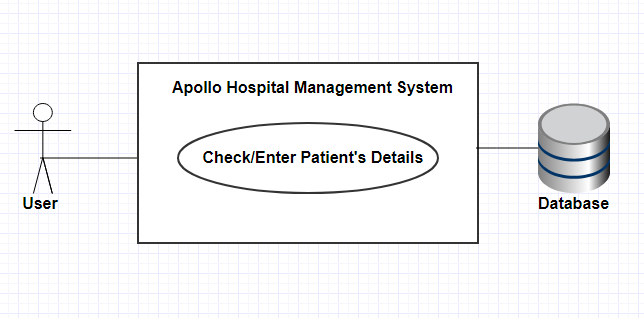
Business Rules User must enter the Doctor Id (Identity of the Doctor)

Exceptions None. Priority Medium.

Frequency of Use Each time the *Doctor’s Details* option is selected to search for available *Apollo Hospital* Doctors.

Assumptions *Apollo Hospital* maintains a current and most accurate database on the authenticated system(s).

**UC-2: Check Patients details**



**Fig. 7.** Use-case diagram to check patient’s details

The above use case diagram has been elucidated in Table 10.

**Table 10.** UC-2 to create check patient’s details

Term Explanation

Actors User or Patient, *Apollo Hospital* Database.

Description The user searches for admitted patients of the hospital or enters the details of a newly registering patient.

Trigger The user selecting “*Patient’s details”* optionin the main menu page.

Flow The user gets directed to the Main menu page upon successful login and

selects an appropriate option.

The user enters Patient Identity. The most relevant *Apollo Hospitals’* admitted patient is presented to the user. If no patients are available then an appropriate error message is displayed to the user.

Business Rules User must enter the Patient Id (Identity of the Patient).

Exceptions The parallelism when servers are deployed is a serious concern (synchronization must be ensured).

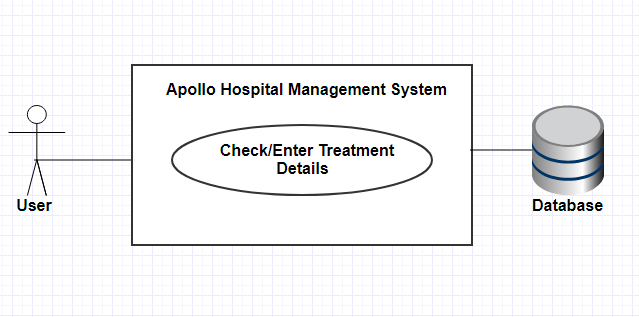
Priority Medium.

Frequency of Use Each time the *Patient’s Details* option is selected to search for admitted

*Apollo Hospital* patients.

Assumptions *Apollo Hospital* takes care of the banking and money transactions; also *Apollo Hospital* provides developers with the admission form and data that is to be requested from the end user.

**UC-3: Check treatment details**



**Fig. 8.** Use-case diagram to check treatment details

The above use case diagram has been elucidated in Table 11.

**Table 11.** UC-3 to check treatment details

Term Explanation

Actors User or Doctor or Patient, *Apollo Hospital* Database.

Description A passenger cancels their reservation for a flight that is scheduled for at least a day in the future.

Trigger The user selecting “*Treatment details”* optionin the main menu page.

Flow The user gets directed to the Main menu page upon successful login and

selects an appropriate option. The user enters Patient Identity. The most relevant *Apollo Hospitals’* admitted patient is presented to the user. If no patients are available then an appropriate error message is displayed to the user.

Business Rules User must enter both Patient Id and Doctor’s Id (Identity numbers of the Patient and Doctor).

Exceptions None. Priority Low

Frequency of Use Each time the *Patient’s Details* option is selected to search for admitted

*Apollo Hospital* patients.

Assumptions The patient has been admitted into the Hospital.

**Appendix C: Traceability Matrix**

The Requirements Traceability Matrix (RTM) tracks system features throughout the software validation process. It ensures that each of the requirements defined for the HMS application are included for full functionality.

**Table 7.** RTM

|  |  |  |  |
| --- | --- | --- | --- |
| System Features | Requirement Specification | Use Case | Specific Requirement |
| SF-1 | Checking Doctor Details | UC-1, | SD-1, SD-2, SD-3, |
|  |  |  | SF-D, SD-5 |
| SF-2 | Checking Patient Details | UC-2 | SP-1, SP-2, SP-3,  SP-4, SP-5 |
| SF-3 | Checking Treatment Details | UC-3 | ST-1, ST-2, ST-3,  SF-T, ST-5 |