**Software Requirements Specification**

**For**

**Hospital Management System**

**Version 1.0 approval pending**

**Prepared by Sanjeev U Rao**

**15IT240**

**Information Technology**

**National Institute of Technology Karnataka**

**8/01/2017**

**Table of Contents**

[Revision History 2](#_Toc503280932)

[1. Introduction 3](#_Toc503280933)

[**1.1 Purpose** 3](#_Toc503280934)

[**1.2 Document Conventions** 3](#_Toc503280935)

[**1.3 Intended Audience and Reading Suggestions** 3](#_Toc503280936)

[**1.4 Product Scope** 3](#_Toc503280937)

[**1.5 References** 4](#_Toc503280938)

[2.Overall Description 5](#_Toc503280939)

[**2.1 Product Perspective** 5](#_Toc503280940)

[**2.2 Product Functions** 5](#_Toc503280941)

[**2.3 User Classes and Characteristics** 6](#_Toc503280942)

[**2.4 Operating Environment** 6](#_Toc503280943)

[**2.5 Design and Implementation Constraints** 6](#_Toc503280944)

[**2.6 User Documentation** 7](#_Toc503280945)

[**2.7 Assumptions and Dependencies** 7](#_Toc503280946)

[3. External Interface Requirements 8](#_Toc503280947)

[**3.1 User Interfaces** 8](#_Toc503280948)

[**3.2 Hardware Interfaces** 8](#_Toc503280949)

[**3.3 Software Interfaces** 8](#_Toc503280950)

[**3.4 Communication Interfaces** 8](#_Toc503280951)

[4. System Features 9](#_Toc503280952)

[**4.1 User Class 1 - The Administrator** 9](#_Toc503280953)

[**4.2 User Class 2 - Doctor** 9](#_Toc503280954)

[**4.3 User Class 3 - The Receptionist** 11](#_Toc503280955)

[5. Other Nonfunctional Requirements 13](#_Toc503280956)

[**5.1 Performance Requirements** 13](#_Toc503280957)

[**5.2 Safety Requirements** 13](#_Toc503280958)

[**5.3 Security Requirements** 13](#_Toc503280959)

[**5.4 Software Quality Attributes** 13](#_Toc503280960)

[**5.5 Business Rules** 13](#_Toc503280961)

[6. Other Requirements 14](#_Toc503280962)

[Appendix A: Glossary 14](#_Toc503280963)

[Appendix B: Analysis Models 14](#_Toc503280964)

[Appendix C: To Be Determined List 14](#_Toc503280965)

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Sanjeev U Rao | 20/01/2018 | Initial Version | 1.0 |
| Sanjeev U Rao | 22/02/2018 | Functional and Non Functional Requirements | 1.1 |

# **1. Introduction**

## **1.1 Purpose**

The Software Requirement Specification document describes requirements and functionality of the system. This document follows the IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998). The intended addressees for this SRS are project examiners and the supervisor. Requirement Analysis is the first technical step in the software process . A general statement of software scope is refined into a concrete pattern that becomes the basis for all software engineering activities that follows.

## **1.2 Document Conventions**

The document has used the font type 'Times New Roman'. The fixed font size that has been used to type this document is 12pt with 1.5 line spacing. It has used the bold property to set the headings of the document. All pages except the cover page are numbered, the numbers appear on the lower right hand corner of the page. Every image and data table are numbered and referred to the in the main text. Standard IEEE template is the template used to organize the appearance of the document and its flow.

## **1.3 Intended Audience and Reading Suggestions**

The intended audience of this document would be the client and specific employees like Manager and Receptionist, consultants and System Operators of the **Pearson Specter International Health Care**, and project team, supervisor with the objective to refer and analyze the information. The SRS document can be used in any case regarding the requirements of the project and the solutions that have been taken. The document would final provide a clear idea about the system that is building.

## **1.4 Product Scope**

Currently **Pearson Specter International Health Care** is using a manual system to handle the hospital process. When patients arrive they make an appointment at the reception to consult a Doctor. These are being recorded in a file. As the current system is file based , management of the hospital has to put much effort on securing the files. They can be easily damaged by fire, insects and natural disasters. Also they could be misplaced by losing data and information.

## **1.5 References**

IEEE Guide for Software Requirements Specifications

<http://ieeexplore.ieee.org/document/278253/>

# **2.Overall Description**

## **2.1 Product Perspective**

Pearson Specter International Health Care follows manual procedures to keep track of its day to day activities. When scenarios such as patient information handling, employee handling, stock handling, financial analysis and report generation is taken into consideration there exists many issues with regard to efficiency, security, accuracy and reliability. Due to improperly managed details medical center faces quite a lot of difficulties in accessing past data as well as managing present data. The manual file systems which are being used at present require storage facilities which is also another overhead. The fully functional automated hospital management system which will be developed through this project will eliminate the disadvantages caused by the manual system by improving the reliability, efficiency and performance. The usage of a database to store patient, employee etc. will accommodate easy access, retrieval, search and manipulation of data. The access limitations provided through access privilege levels will enhance the security of the system.

## **2.2 Product Functions**

OPD and Consultation Management

* Recording patient details
* Recording Appointments taken
* Set Appointment status

Patient Management

* Register Patient
* Login/Logout
* Take Appointment

## **2.3 User Classes and Characteristics**

**Admin**

Admin has the full access to the system which means he/she is able to manage any activity with regard to the system. Admin is the highest privileged user who can access the system.

**Doctor**

Every doctor has a unique account. Doctor can look at all the appointments requested by the patients.

**Patient**

Patient can create an account and take an appointment with a doctor. Patient can view all available doctors and their respective specialties.

**Receptionist**

Every receptionist has a unique account. The receptionist can manage doctor appointments. The receptionist sets appointment status.

**Appointment**

Users can schedule appointments. Every appointment has fields like patient, doctor, date, status, status message. The status and status message are set by the receptionist.

## **2.4 Operating Environment**

**Software requirements**

● Django framework

● mysqli server

**Hardware Requirements**

● Core i5 processor

● 4GB Ram

● 20GB of hard disk space in terminal machines

## **2.5 Design and Implementation Constraints**

● System is only accessible within the hospital premises.

● Database is password protected.

● Should use less RAM and processing power.

● Each user should have individual ID and password.

● Only admin can access the whole system.

## 

## 

## 

## **2.6 User Documentation**

As a part of the system itself a user documentation is provided to the customers which gives an overview of the system. It will include the full description about the product and complete orderly followed steps to install the software. The users will get the opportunity to use the system without having any trouble.

## **2.7 Assumptions and Dependencies**

● Each user must have a valid user id and password

● Server must be running for the system to function

● Users must log in to the system to access any record.

● Only the Administrator can delete records.

# 

# 

# **3. External Interface Requirements**

## **3.1 User Interfaces**

TBD

## **3.2 Hardware Interfaces**

Since the application does not have a designated hardware, it does not have any direct

hardware interfaces. The hardware connection to the database servers is managed by the underlying operating system on the application.

## **3.3 Software Interfaces**

**Developing End**

* **JDK 1.8 -** Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet.
* **Netbeans 8.1** - IDE for Java developing.
* **MySQL server** - Database connectivity and management

**Client End**

* **OS** – Windows / Ubuntu
* **JRE 1.8**  - JAVA Runtime Environment for run Java Application and System
* **MySQL server** - Database connectivity

## **3.4 Communication Interfaces**

TBD

# 

# **4. System Features**

This section includes the requirements that specify all the fundamental actions of the software system.

## **4.1 User Class 1 - The Administrator**

**4.1.1 Login**

**4.1.1.1 Description and Priority**

The administrator has access to every database record. This is a high priority.

**4.1.1.2 Stimulus/Response Sequences**

On the homepage, login button is pressed. Below the login form, there will be a link to the registration page.

**4.1.1.3 Functional Requirements**

REQ-1: The registration page must accept name, username, password and phone number as the input.

REQ-2: The password must be verified with a confirm password field.

REQ-3: The username must be unique for each user.

REQ-4: No field must be blank.

**4.1.2 Manage Database**

**4.1.2.1 Description and Priority**

The administrator has access to every database record. This is a high priority.

**4.1.2.2 Stimulus/Response Sequences**

Admin has access to the database once logged in.

**4.1.2.3 Functional Requirements**

REQ-1: Manage patient details.

REQ-2: Manage doctor details.

REQ-3: Manage scheduled appointments.

REQ-4: Manage all employee details.

## **4.2 User Class 2 - Doctor**

**4.2.1 Registration**

**4.2.1.1 Description and Priority**

Every doctor must have a unique account through which he/she can manage appointments and visiting hours. This has a high priority.

**4.2.1.2 Stimulus/Response Sequences**

On the homepage, login button is pressed.

**4.2.1.3 Functional Requirements**

REQ-1: The registration page must accept name, username, password and phone number as the input.

REQ-2: The password must be verified with a confirm password field.

REQ-3: The username must be unique for each user.

REQ-4: No field must be blank.

**4.2.2 Login**

**4.2.2.1 Description and Priority**

Given that a doctor has an account, then the doctor should be able to log in to the application. This has a high priority, because without this snippet, the entire application would be useless.

**4.2.2.2 Stimulus/Response Sequences**

On the homepage, login button is pressed. The login form is then shown.

**4.2.2.3 Functional Requirements**

REQ-1: The login form must have radio buttons to select between doctor, employee and administrator.

REQ-2: The login form must query for username and password.

REQ-3: In case of an error, the error is shown explicitly.

**4.2.3 Manage working hours**

**4.2.3.1 Description and Priority**

Once the doctor has logged in, he must be able to manage his visiting hours so that the hospital can schedule appointments accordingly.

**4.2.3.2 Functional Requirements**

REQ-1: Doctor must be able to see scheduled appointments.

## 

## 

## **4.3 User Class 3 - The Receptionist**

**4.3.1 Registration**

**4.3.1.1 Description and Priority**

The receptionist must have a unique account. This has a high priority.

**4.3.1.2 Stimulus/Response Sequences**

On the homepage, login button is pressed. Below the login form, there will be a link to the registration page.

**4.3.1.3 Functional Requirements**

REQ-1: The registration page must accept name, username, password and phone number as the input.

REQ-2: The password must be verified with a confirm password field.

REQ-3: The username must be unique for each user.

REQ-4: No field must be blank.

**4.3.2 Login**

**4.3.2.1 Description and Priority**

The receptionist must be able to login using unique username and password. This has a high priority.

**4.3.2.2 Stimulus/Response Sequences**

On the homepage, login button is pressed. The login form is then shown.

**4.3.2.3 Functional Requirements**

REQ-1: The login form must have radio buttons to select between doctor, employee and administrator.

REQ-2: The login form must query for username and password.

REQ-3: In case of an error, the error is shown explicitly.

**4.3.3 Schedule appointments**

**4.3.3.1 Description and Priority**

The receptionist must schedule appointments to the patients according to the doctors’ availability. This has a high priority.

**4.3.3.2 Functional Requirements**

REQ-1: Receptionist must schedule appointments according to the doctors’ visiting hours.

**4.3.4 Manage Patient Details**

**4.3.4.1 Description and Priority**

The receptionist must record the patient details . This has a high priority.

**4.3.3.2 Functional Requirements**

REQ-1: Receptionist must schedule appointments according to the doctors’ visiting hours.

REQ-2 Search for a patient record.

# 

# 

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

* Fast response time.
* Efficiency and reliability in database access.

## **5.2 Safety Requirements**

* Prevent access by unauthorized users.
* Handle catastrophic failure to the software or the database.

## **5.3 Security Requirements**

## All the administrative and data entry operators have unique logins so system can understand who is login in to system right now no intruders allowed except system administrative nobody cannot change record and valuable data.

## **5.4 Software Quality Attributes**

* **AVAILABILITY**: The system shall be available all the time.
* **CORRECTNESS**: A bug free software which fulfils the correct need/requirements of the client.
* **MAINTAINABILITY**: The ability to maintain and modify information and update fix problems of the system.
* **USABILITY**: software can be used again and again without distortion.
* **ACCESSIBILITY**: Administrator and many other users can access the system but the access level is controlled for each user according to their work scope.
* **STABILITY**: The system outcome/output won’t change time to time. Same output will be given always for a given input.

## **5.5 Business Rules**

* Warranty period of maintaining the software would be one year.
* If any error occur due to a user’s improper use. Warranty will not be allocated to it.
* Won’t take the responsibility of failures due to hardware malfunctioning.

# **6. Other Requirements**

# **Appendix A: Glossary**

# **Appendix B: Analysis Models**

# **Appendix C: To Be Determined List**

* “Forgot password” section.
* In-App pharmacy.
* Security requirements
* Software quality attributes
* Business rules