Software Requirements Specification

for

Hospital Management System

Version 1.0 approved

Prepared by :- JEWEL SIMON(13)

SURAJ KAMAT(16)

VIKRAM JAIN(11)

St .John college of Engineering and Management

3 -FEB-2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Matthew Slater	*12/15/14*	*Revised Sections, Changed ToC, Updated Information*	*1.0*
Matthew Slater	*1/27/15*	*Revised Sections and Errors based on Review Feedback*	*1.1*

1. Introduction

1.1 Purpose

The purpose of the Hospital Management System (HMS) is to create a Free and Open Source Software (FOSS) Hospital Management System.
This project is being developed mainly for our client, Appalachian Health Network (AHN), but will be extensible enough to be adapted and customizable for deployment and integration into any hospital network. The SHMS contains five distinct sub-systems:

- Radiology Information System (RIS)
- Picture archiving and Communication System (PACS)
- Image Acquisition Modality System (IAM)
- Admit-Discharge-Transfer/Patient Registration System (ADT/PRS)
- Hospital Management Systems (HIS)

1.2 Document Conventions

This document will use IEEE format. For clarity, acronyms and technical jargon, deemed uncommon by the author, will be annotated and included in the glossary. The format for headings is as followed:
Major headings are in **bold 18pt font,** and concurrent headings in **bold 14 pt font**. Sections are in the format of x.y, where x and y are real, positive integers.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification document is intended for software engineers, system testers and software designers in developing, testing, and producing the SHMS and for the project. It is suggested to read the sections sequentially, and to reference the appendices as one progresses, in order to clarify jargon terms and definitions.

1.4 Product Scope

This SRS details the development of the Hospital Management System project and the five subsystems. This project is open source and shall be available to modification at no restriction from Inc. Inc. is not responsible or liable for any changes made to this project outside of its initial release.
The scope of the RIS subsystem is to create a generic FOSS RIS program, which can be customized for deployment and integration into any hospital's use; however, once developed, the RIS will be customized to the needs of our client, Allegheny Health Network.

1.5 References

https://en.wikipedia.org/wiki/Hospital_information_system

2. Overall Description

2.1 Product Perspective

The Hospital Management System is an open source system comprising of five different subsystems. The five subsystems are as follow:
The FOSS RIS project is a separate program, which is a component of a larger FOSS Hospital Management System (HMS), similar to how Microsoft Word is a separate program inside Microsoft Office suite. In the FOSS HMS system, the FOSS RIS program performs all HIS operations. The RIS module uses the shared, global variables, enums, framework, and used to create the other FOSS HIS program components, just like with Microsoft Office. All data exclusive to the RIS module will be programmed in the RIS module.
Hospital Information System will replace all traditional and outdated means of tracking patient information and other data useful to the hospital. A Hospital Information System shall replace forms of databases using manual or outdated hardcopy databases. Accessing data can be better monitored, organized, and time conscientious. The IAM program shall be a new management system which shall make individual systems obsolete. It shall allow one program to control all the different image acquisition devices and shall interact with the other components of the hospital management system being designed.
The driving principle of this PACS is to automate and provide the infrastructure to digitally control the storage and transportation of images taken with compatible devices within a general hospital.
The ADT/PRS subsystem stores patient data, which other subsystems can access as required. This is accomplished by granting the other systems access to this subsystem's patient database.

2.2 Product Functions

- The RIS subsystem shall include patient list management, radiology department workflow management, request and document x-ray scanning, result entry, and reporting and printout/faxing and emailing of clinical reports.
- The PACS subsystem shall perform image importing/capturing, image encryption, local image storage, remote image retrieval, image compression, image display, and image processing.
- The HIS subsystem shall contain a secure database. The database GUI shall be user friendly for all staff members and properly enter/obtain/modify patient information. The DB shall utilize the token authentication for secure access and will be relative in size and flexibility of the data demand.

- The IAM subsystem shall have a simple user interface which allows the user to log in then access any imaging device connected to the imaging intranet, select what type of device and then which specific device within the hospital they will use. The images shall be controlled from one console and share these images with the hospital patient database.
- The ADT/PRS subsystem shall allow an administrator to enter patient information, such as name, age, etc. That information is then stored, and shared with other users as appropriate. It shall also alert the medical staff when a patient that requires different treatment is admitted, such as some with an infectious disease.

2.3 User Classes and Characteristics

The entire FOSS SHMS suite program has a set of users, each with different security privileges. These user types are head doctor/nurse, and doctor/nurse. The head doctor/nurse can control most of the system, can transfer data in/out of hospital networks and to other doctors/nurses who need the patient's medical information from the patient database, and possesses read/write permissions on sending/receiving data to/from the database. The head can also control the PACS subsystem and the ADT/PRS subsystem. The doctor can receive data in/out of a health network from the database with permission from a head doctor/nurse, but the data is read-only. If changes or updates need to be made on the data, the doctor/nurse must put in a request with the head doctor/nurse to make the changes to the database. This is the same for the ADT/PRS subsystem, where only the administrator can enter/edit data. All access and data transfers/receiving is logged, in order to maintain a level of transparency, in order to prevent abuse of the system, and in order to hunt down any unauthorized users or hackers. This logging is done by the software logging each function call along with its parameters being passed, as well as the current user logged on who performed them.

2.4 Operating Environment

The FOSS SHMS program runs on Windows 7, for 32-bit/x86 and 64-bit/x64 PC architectures. The software for the RIS subsystem will be written in C#, using Microsoft Visual Studio 2010. The program will be GUI-based (like with most modern Windows software).
The HIS subsystem will run off a Cloud-Based Platform. The Cloud-based server will utilize Oracle or SQL database running on the cloud. The operating system shall be a MS-Windows or UNIX. Integration to the server shall be done via a HTTPS, SFTP, or VPN to create, update, fetch, or delete data.

2.5 Design and Implementation Constraints

Items and issues that may limit the options available to the software developers are legal and ethical constraints with regard to SHMS development and medical practices, and possible social and legal opposition by HIS corporations who loathe FOSS software. Moreover, parallel threads will need to take place in the larger HIS operation, which will require research in how to program and operate with several, parallel-running threads in the same application. Constraints of the user-permissions system specified in §2.4 must be programmed, for the database system. This project shall implement a series of subsystems that shall contain sensitive medical and personal records. Due to this, security features and login fail safes shall be of the highest concern when developing this project. Such security features include high-security of data transfers, and encrypted network communications, as well as programming logging of function calls as well as parameters passed.
It is anticipated that all related governing directives both social and governmental regulations will be adhered to; thus in accordance with The Health Insurance Portability and Accountability Act of 1996 (HIPAA), access to images will be strictly enforced by the Authentication Module. Encryption will be employed to keep health information secure, but may impose a processing overhead that can potentially hinder timing requirements
Due to the large nature of the project, keeping track of the source code between the developer sub-teams will be difficult. We plan to implement a subversion/source control system, most likely Github, where we will pull/push code commits to/from the Github server. The source code, as well as the current folder/file structure, will be able to be uploaded and fetched from our Github account. Once completed, the software will be continuously updated by the developers, and

major upgrades to the system can be downloaded from our website, Softright.com, as service packs. Smaller bug fixes can be downloaded as hotfixes, also available for download from the website. Updates can be discovered by manually browsing our website, or by pulling down the help tab, which has a “Check for Updates…” feature.

2.6 User Documentation

The application will come with an “About” tab, which will allow users to access the offline and online HTML .hlp help manual. This manual will be updated with each new service pack. Other user documentation includes one user manual for lowest level users, one technical document describing the functionality of the sub-section in detail for use of technicians, one copy of documentation and link to current source for future contributors.

2.7 Assumptions and Dependencies

The developers, assume that we will have to “pave our own way” concerning programming the majority of the application, due to the mostly closed-source and secretive nature of major SHMS software. For what we cannot find from open documentation and research, it is assumed that we will have to deduce how HIS standards and protocols work from observing external behaviors found in existing HIS software, and we will have to replicate the results using our own code and other FOSS applications and libraries. It is assumed that social and legal opposition by money-hungry HIS corporations who loathe FOSS software could occur. The project will have to depend on FOSS SQL database libraries, 7zip .7z compression libraries, OpenTLS libraries, TCP/IP libraries, and other FOSS libraries, in order to keep this software free of proprietary libraries, in order to keep the software in a FOSS status. This project is developed under the working assumption that as an open source project it shall be noted that the project shall change overtime. Regular changes to this SRS shall occur for each change enacted by Inc.
It is assumed that the PACS will be used in a Hospital Environment by untechnical users. It is assumed that the infrastructure for capturing digital images in either .JPG, .GIF, .DICOM, etc will exist. It is assumed that the System will be networked, and capable of routing to an internet gateway.

3. External Interface Requirements

3.1 User Interfaces

The user interface of the software will use standard Windows API and GUIs using C#. All five subsystems will utilize the Windows-style GUI. Due to the varying age group of users (from younger interns to middle-aged doctors), the GUI needs to adapt to the age group’s GUI and computing preference. The application has two (2) GUI styles:
• Classic
o For older users
o Similar to Microsoft Office XP
o Uses the traditional GUI style from Microsoft XP and older, has
▪ Tabs
▪ Buttons
▪ Dialog boxes

- List boxes
- Etc

3.2 Hardware Interfaces

Database interfacing will use standard TCP/IP protocols, but using FOSS libraries, for computers connected to the internet via LAN Ethernet cables. Due to security concerns and the ease of hackers cracking WiFi hotspots, WiFi internet support with the software is prohibited by the software. The software detects whether a LAN or WiFi connection is used, and will terminate the program if WiFi is detected to be in use for the internet connection.
The PACS subsystem shall utilize USB support for importing photos directly from the devices into the PACS. Meanwhile, the HIS subsystem shall enter patient information with a secure connection through an encrypted computer or tablet being used by the medical staff of the hospital. An encrypted computer that has a secure VPN is the preferred method of communicating to the database.

3.3 Software Interfaces

The FOSS RIS module will be able to interface with the bigger, FOSS HMS module, and will inherit the global, shared, structs, enums, functions, and variables, as well as use its own data pertinent and exclusive to itself. It will interface with standard Windows API and GUI. Data that will be shared between computers and instance of the software being run will be pushed and pulled from the patient database as needed/requested. This database can be configured and deployed for usage in case-specific usages for each hospital network that uses the software in its own LAN or intranet. All offline and online access will be monitored, for transparency purposes, and in order to reduce abuse and unauthorized access of the system.
For the PACS subsystem, software interfaces will include standard TCP/IP interfaces with network devices, and software drivers for USB devices. The database will be accessed through standard PL/SQL scripts, which will be native to the system and included as packages.

3.4 Communications Interfaces

Communication interfaces will use TCP/IP for data transmission and SMTP/HTTP for generating emails of reports from the software. FTP can also be used in pushing generated document reports to a hospitals FTP server. All communication interfaces should have high baud data Tx and Rx rates ranging from Mbps to Gbps. FOSS TLS or higher encryption standards are a must-have, high priority requirement. All offline and online access will be monitored, for transparency purposes, and in order to reduce abuse and unauthorized access of the system.

4. System Features

4.1 Radiology Information System

4.1.1 Patient Registration and scheduling

This is the feature which registers patients and schedules appointment for X-Rays. It will allow the user to pull up the list of existing patients and edit data in an Excel-like spreadsheet database, schedule and view appointments from a calendar, and push/pull that data to/from the database, with head doctor permission. An email will be sent to the patient, reminding him of his appointment.

4.1.2 Functional Requirements

R-PRS-1: User shall pull up list of existing patients via dialog box
R-PRS-2: User shall be able to add/remove/edit (with permission) patient information from Excel-like spreadsheet database
R-PRS-3: User shall be able to schedule patients for X-Ray appointments
R-PRS-4: Doctor shall be able to transfer/receive data to/from the database, after making request to Head Doctor
R-PRS-5: Software shall send email to patient reminding them of their appointment
R-PRS-6: All offline and online actions shall be monitored by the software's logging feature, in order to maintain transparency and to minimize risk/abuse of the system.

4.2 Radiology Department workflow management

- 4.2.1 Description and Priority

This feature allows the radiology department to schedule, list, and cross off work tasks in a to-do list.

- 4.2.2 Functional Requirements

R-RWM-1: Doctor shall be able to receiver/transfer work tasks to/from the workflow database
R-RWM-2: Doctor shall be able to send request of adding info to the Head Doctor
R-RWM-3: Head Doctor shall be able to dis/approve adding/crossing off work from the workflow database
R-RWM-4: Doctor shall be able to request crossing off tasks from Head Doctor
R-RWM-5: All offline and online actions shall be monitored by the software's logging feature, in order to maintain transparency and to minimize risk/abuse of the system.

4.3 Request and document X-Ray scanning

- 4.3.1 Description and Priority

This feature allows the radiology department to request x-ray scanning, which will then be sent to the Image Acquisition Module (IAM), and to keep and manipulate the image files of the x-rays.

- 4.3.2 Functional Requirements

R-XRAY-1: Doctor shall be able to send request of patient's x-ray to the Head Doctor
R-XRAY-2: Head Doctor shall be able to dis/approve the x-ray
R-XRAY-3: Head Doctor will send his dis/approval of x-ray to radiology technicians, and to the IAM

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software should have **high** performance and **low** failure rates. The hardware and software should be able to transmit/receive data from databases with high baud rates, ranging from Mbps to Gbps. Machines should have **all** recent Windows updates installed, and have their security not compromised by viruses. Machines must have firewalls installed and active virus scanning software in usage. Machines should solely be used for operation of the software, in order to maximize performance and security. Furthermore, X-Ray scanning hardware should operate within small safety tolerances.

5.2 Safety Requirements

In regards to the RIS subsystem and for the safety of the patients, all patients must wear lead vests for the body part being X-Rayed, and the X-Ray equipment must be regularly maintained, inspected, and used responsibly. The software shall have built in safeguards which shall terminate any imaging process should it produce an unsafe level of radiation. Should a user attempt to send a dosage of radiation which exceeds recommended levels the imaging session shall be terminated and any previously captured images stored.
The system shall not perform diagnoses. The User is responsible for performing diagnoses. The system shall only facilitate the display of information to aid the User in making quick and timely diagnoses; thus, the information must be displayed visibly and with as much clarity as the hardware infrastructure was capable of providing
Furthermore, as previously mentioned, all computers must be used solely for operation of the software as dedicated workstations, and must have all recent Windows updates installed, an active firewall instance, and have solid anti-virus software, in order to protect the patient's private and confidential medical information.
All offline and online access will be monitored, for transparency purposes, and in order to reduce abuse and unauthorized access of the system. Most actions will require a yes/no confirmation before it will actually be performed.

5.3 Security Requirements

All data receiving and transmissions should be done using FOSS TLS or higher encryption, in order to keep the patient's private medical and social security information out of the wrong hands. The FOSS TLS software must be inspected, in order to verify if recent security exploits/hacks of the system are patched (for example, HEARTBLEED for SSL encryption). In addition, all computers **must** have firewalls, and be operating on a **LAN** internet connection, **not** a WiFi connection. Moreover, all computers must have all recent Windows updates installed, and **must** have **solid** anti-virus software. Also, the user-permissions system mentioned in §2.3 will be implemented. Before any user can access the system, they shall be required to input a company username, an ID number, and a password. Each password shall be required to be between 8-12 characters in length and shall be required to contain at least one capital letter, one number, and one special character. Passwords will need to be changed every half-year, with a unique password.

5.4 Software Quality Attributes

Flexibility, reusability, robustness, and maintainability of the HMS system should be maximized, in order for clients to be able to deploy custom settings of the FOSS HMS to their individual hospital network needs.

Appendix A: Glossary

1. **7Zip**
a. Highly acclaimed and functional, multi-platform, FOSS compression file format standard
b. Utilizes GNU LGPL License
c. See http://www.7-zip.org/ for more information

2. **Baud rate**
a. Rate of transfer of data over the internet/network
b. Measured in bit per second (bps)

3. **Bit**
a. Binary Digit
b. One zero(0) or one (1).

4. **Classic style**
a. Refers to the type of GUI style in the RIS software that resembles Microsoft Office XP,
b. With the static menu bars and dialog boxes

5. **Database**
a. Big memory address block which contains large set of data
b. With subsets and fields that can search for by filter, read, and written

6. **FOSS**
a. Free and Open Source Software
b. Software that is freeware, and which has its source code available, for others to modify under the GNU Software License
c. See https://www.gnu.org/copyleft/gpl.html for GPL information.

7. **FTP**
a. File Transfer Protocol
b. Protocol to send/receive files to/from an FTP server

8. **GUI**
a. Graphical User Interface
b. The interface which the user uses on graphics displaying hardware
c. Refers to the layout of the dialog boxes, menu elements, etc

9. **HIS**
a. Hospital Information System
b. Element of health informatics
c. Focuses mainly on the administration needs of hospitals
d. In many implementations, a HIS is a comprehensive, integrated information system designed to manage all the aspects of a hospital's operation
i. Medical
ii. Administrative
iii. Financial
iv. Legal issues
v. Corresponding processing of services.