Hospital Management System Requirements Specification

Version 1.0

April 21, 2022

Table of Contents

VERSION 1.0

APRIL 21, 2022

1. Executive Summary

1.1 Project Overview

This is a Project for the Software Engineering course at Epoka University, created by a group of 6 students of CEN. The project is a web site/application designed to help Hospitals, here in Albania, manage their operation in a more effective and convenient way.

1.2 Purpose and Scope of this Specification

The Main objective of the Web application is to allow Hospitals here in Albania a system to interact better and more conveniently with patients as well as having an easier time managing different aspects of running an effective Hospital. One of the main aspects of this solution is a way for patients to have an easier time looking for doctors and booking and appointment with specialists in regards to their needs. As for the hospital side, it will be a website that will allow an easier time to the hospital staff to manage their interaction with patients as well as the documentation needed for them (i.e. “Librez shendetsore”, analysis results, x-rays, MRI, ect).

In scope:

- This project is responsible for creating a platform where patients can contact and book appointments, as well as being able to see which doctors are available in the Hospital at which times.
- On the Hospital side, the web application will be able to create appointments for the Doctors. Will be able to archive and manage the necessary documents to run the hospital.
- There will be web security for the website, as well as 2FA for the login giving the log in process an extra layer of security.
- Maintenance of the web service will be provided by us.
- A mobile app will be developed to offer the booking services more easily.

Out of scope:

- The Web Application (and hence our service) will not be responsible for any monetary transaction between the Hospital and its patients.
- Changes will be made to accommodate for legislative requirements and changes along the way.

2. Product/Service Description

A website that serves at the same time the possible/current patients of the hospital, as well as the hospital staff/management to manage their appointments, their schedule as well as their documents.

2.1 Product Context

For now a good way to demonstrate the context of our project would be the diagrams made available on the projects repository on GitHub :
https://github.com/EloiSherifi/SoftwareEngineeringProject/tree/main/Diagrams

2.2 User Characteristics

User :

- Register
- Log-in
- Request appointment
- View own profile
- View available Doctors
- Use filters to better find the appropriate Doctor
- Delete their own profile
- Review their booked times
- Look up when a certain doctor is free.
- Look at their appointment history
- Look and print their X-Rays/MRI/Receipts

Doctor/Medical staff:

- Log-in
- Check his assigned patients / his history of patients
- Check their timetable
- Check their requested appointments and accept/deny them
- Upload Receipts into their patients profile
- Look at medical documents and the medical history of their patients

Non-medical staff:

- Log-in
- Look at their schedule
- In the case of Reception, request an appointment for an in-place patient.
- Fill up necessary documents for their respective work

Admin:

- Log-in
- Create/Delete/Change internal accounts (Medical and non-medical users)
- Archiving former staff members account and their respective documents

Guest:

- View the front page
- Look up the Frequently asked questions page (health library)
- Get the Hospital Contacts.

2.3 Assumptions

Our assumptions are that the staff members will adapt the use of this software and integrate it into their daily work. It's assumed that the technical background of the users of this software is of a degree that renders the use of the website intuitive. It's assumed that the scheduled appointments will be respected and the doctors as well as the patients will be on time.

2.4 Constraints

- Deadline : End of course/semester (spring semester of 2022)
- Technologies used : HTML, CSS, JavaScript, Bootstrap, PHP, MySql.
- The timetable will be updated by the internal staff of the Hospital and will be respected.

2.5 Dependencies

- This product will require daily/weekly maintenance by the system admin in order to add/remove staff members
- The timetable for the staff (working hours) needs to be updated with eventual changes that may happen during the working shifts.
- Regular updates of available staff will also be required to maintain functionality for the patients.

3. Requirements

3.1 Functional Requirements

Req#	Requirement	Priority	Date Rvwd	SME Reviewed / Approved
FR#01	The website should have an appointment booking system where users make requests and staff accept or deny them.	1	20/04/2022	
FR#02	The website should offer a way for doctors as well as patients to upload relevant medical files (in .pdf format) that are obtained from 3rd party sources.	1	20/04/2022	
FR#03	The website should offer to everyone a view of the available doctors in the hospital with their specific qualifications.	3	20/04/2022	
FR#04	The website should provide users who have selected a specific doctor out of the staff list with their timetable, showing their available time to take appointments.	1	20/04/2022	
FR#05	The website should allow doctors to give patients drug receipts and the patients should be able to view these receipts.	2	20/04/2022	
FR#06	The website should offer patient users to view their appointment history as well as their former receipts	2	20/04/2022	
FR#07	The website should be able to send to patients their specific documentation via email (as well as having it available in their profile page)	4	20/04/2022	
FR#08	The website should have a system for the receptionist(non-medical staff member) and the pathologist (family doctor) to arrange meetings between patients and other doctors.	1	20/04/2022	
FR#08	The website will have customizable font size to offer better accessibility to visually impaired users	4	22/04/2022	

3.2 Non-Functional Requirements

3.2.1 Product Requirements

3.2.1.1 **User Interface Requirements**

UI requirements :

- Legible fonts which will help users with visual impairments to still be easily readable as well as keeping a professional look for the website.
- The different aspects of the webpage will be flexible to be able to have the best user experience at different screen sizes (like desktop, smaller laptops, resized windows as well as mobile devices.
- The navigation bar will be present in every page of the website and have tabs specific to different users.
- The login/signup form will be as a sidebar opened on the side of the web page without loading in a new page.

User specific UI:

- Patients:
    - They will have a different taskbar offering the view of their profile (where they will find their documents as well as the ability to add and remove documents they have uploaded.

    - “Doctors” where they will be taken to the doctor’s list page where they can view and select the doctor they need.
    - “Appointments” where the patients will be able to check their booked appointments as well as a history of their past appointments.
    - “History” where the patients will have a view of their receipts.
    - A profile icon which will, when clicked, take the user to their profile where they will have the option to change their email and their password.
- Doctors:
    - Their navigation bar will have “Appointments”, “Schedule”, “Patients” and “Profile”.
    - “Appointments” will contain the timetable of the doctor’s appointments as well as the ability to see new requests that can then be accepted or denied.
    - “Schedule” will contain the working schedule of the Doctor for this week and for the upcoming weeks.
    - “Patients” will contain a history of patients that have taken appointments before with the doctor as well as the ability to view their profile as well as inserting documents/receipts.
    - “Profile”, just like with the patient's profile, doctors will be able to change their email and password.
- Staff:
    - They will have different navigation bar with respect to their position.
    - Receptionist will have a “create appointment request” tab to create an appointment for a patient that has shown up to the reception desk.
    - “View Doctors” tab to view the available doctors at this hour (doctors that are currently working this shift) to better aid the patients seeking help at the reception desk.
    - “Schedule” to, just like the doctors, view their shift for the current week and the upcoming weeks.
- Administrator :

- Will have a more simplistic taskbar which will allow him to access the database and add/remove doctors from

3.2.1.2 **Usability**

Include any specific usability requirements, for example,

Learnability

- The user documentation and help should be complete
- The help should be context sensitive and explain how to achieve common tasks
- The system should be easy to learn

(See http://www.usabilitynet.org/)

3.2.1.3 **Efficiency**

3.2.1.3.1 Performance Requirements

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

- Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.
- Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form. For example, "95% of the transactions shall be processed in less than 1 second" rather than “an operator shall not have to wait for the transaction to complete”.

3.2.1.3.2 Space Requirements

3.2.1.4 **Dependability**

Availability

Include specific and measurable requirements for:

- Hours of operation
- Level of availability required
- Coverage for geographic areas
- Impact of downtime on users and business operations
- Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures
- reliability (e.g., acceptable mean time between failures (MTBF), or the maximum permitted number of failures per hour).

Reliability

Monitoring
Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.
Maintenance

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

Integrity

3.2.1.5 Security

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

- encryption
- activity logging, historical data sets
- restrictions on intermodule communications
- data integrity checks

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

3.2.2 Organizational Requirements

Requirements which are a consequence of organizational policies and procedures e.g. process standards used, implementation requirements, etc

3.2.2.1 Environmental Requirements

3.2.2.2 Operational Requirements

3.2.2.3 Development Requirements

3.2.3 External Requirements

- Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

3.2.3.1 Regulatory Requirements

3.2.3.2 Ethical Requirements

3.2.3.3 Legislative Requirements

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values

3.2.3.3.1 Accounting Requirements

3.2.3.3.2 Security Requirements

3.3 Domain Requirements

Everything related to the domain that might be needed in the project shall be mentioned here. Sometimes the domain Requirements might be thought of as part of either functional or non-functional requirements.

Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

4. User Scenarios/Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

- Describes a significant business need
- Identifies, documents, and ranks the problem that is driving the scenario
- Describes the business and technical environment that will resolve the problem
- States the desired objectives
- Shows the “Actors” and where they fit in the business model
- Is specific, and measurable, and uses clear metrics for success

APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

Appendix A. Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

Appendix B. References

List all the documents and other materials referenced in this document.

Appendix C. Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

Business Requirement	Area	Deliverables	Status
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process =	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01	Accepted

Business Requirement	Area	Deliverables	Status
"Assigning a Bargaining Unit to an Appointment" (Priority 1)		Derive Bargaining Unit-Process Flow Diagram	
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview
BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_0	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
9					
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		
BR_LR_11		X											

Appendix D. Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode
Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class
Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects
Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature
A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

By Stimulus
Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response
Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy
When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments
Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.