



HEALTH CARE SYSTEM

**Software requirement specification (SRS)
Report of HCS Project**

Course: Advanced Programming(CSEg3203)

**Submitted to:
Yared Daba Tesfaye**

Department of Computer Science and Engineering

Section 2

Name

Id.No

1. Samuel Tesfachew----- UGR/31190/15
2. Olani Shambel ----- UGR/31097/15
3. Kenenisa Mekonen ----- UGR/30771/15
4. Yeabsira Zerihun ----- UGR/30598/15
5. Israel Bekele ----- UGR/31494/15
6. Abdulgeni Abdulaziz ----- UGR/30027/15

Date: Dec, 2024

Abstract

The **Health Care System (HCS)** is a comprehensive software solution designed to streamline and automate the administrative, clinical, and financial operations of a hospital. It aims to enhance efficiency, accuracy, and patient satisfaction by integrating core functionalities such as patient registration, appointment scheduling, medical records management, billing, and inventory control.

This software requirement specification (SRS) outlines the functional and non-functional requirements of the HCS, providing a detailed framework for system development and implementation. The system ensures secure data handling, user-friendly interfaces, and interoperability with existing hospital infrastructure. By centralizing hospital operations, the HCS supports better resource utilization, improved decision-making, and adherence to regulatory standards.

Table of Contents

Table of Contents	i
1. Introduction	6
1.1 Purpose	6
1.2 Scope	7
1.3 overview.....	7
2. General Description	8
2.1 product perspective	8
2.2 product feature	8
2.3 Design and implementation constraints	8
2.4 assumption and dependencies	8
3. Function requirement	9
3.1 Description	9
3.2 Technique issues	9
4. Interference requirement	11
4.1 user interference	11
4.2 hardware interreference	11
4.3 software interference	12
4.4 communication interference	12
5. software requirement analysis	13
5.1 define problem	13
5.2 define module and functionality	13
6. software design	16
7. Non-function requirement	21
7.1 performance	15
7.2 security	15

21	7.3 reliability	
	7.4 availability	16
	7.5 safety	16
	7.6 software quality	16
	7.7 reusability	16
	7.8 maintenance	17
	8. conclusion	17
	9. reference	17

1. INTRODUCTION

1.1 PURPOSE

- **Define System Scope and Objectives:**

- Clearly identify the problems the HCS aims to solve, such as inefficiencies in patient management, billing, or inventory tracking.
- Outline the system's expected outcomes, like improved workflow efficiency and better patient care.

- **Establish Functional Requirements:**

- Specify the core functionalities, such as patient registration, appointment scheduling, medical records management, billing, and reporting.

- **Outline Non-Functional Requirements:**

- Detail performance metrics, scalability, security protocols, and system usability standards.

- **Serve as a Communication Tool:**

- Bridge the gap between stakeholders (e.g., health care management, IT teams, and developers) by ensuring everyone aligns on the system's requirements.

- **Provide a Blueprint for Development:**

- Act as a reference for software designers and developers to implement the system as intended.
- Include system constraints, hardware/software dependencies, and design considerations.

- **Facilitate Testing and Validation:**

- Serve as a baseline for verifying that the final product meets the specified requirements during testing phases.

- **Minimize Development Risks:**

- Identify potential challenges or ambiguities early in the development cycle, reducing the risk of costly modifications later.

1.2 SCOPE

The **Health Care System (HMS)** is designed to automate and integrate various administrative, clinical, and financial functions of a health care. It provides a centralized platform to manage patient records, appointments, billing, staff scheduling, inventory, and reporting. The system enhances efficiency, reduces paperwork, improves data accuracy, and ensures seamless communication between departments. It caters to the needs of hospital administrators, medical staff, and patients, aiming to improve operational workflows, patient care quality, and compliance with healthcare standards.

1.3 OVERVIEW

This **Software Requirements Specification (SRS)** defines the requirements for the development of the HMS. It outlines:

- **Functional Requirements:** Core features like patient registration, appointment scheduling, electronic medical records (EMR), billing, and inventory management.
- **Non-Functional Requirements:** Performance, reliability, scalability, security, and user-friendliness.
- **System Interfaces:** Integration with existing hospital infrastructure such as laboratory systems, pharmacy, and insurance portals. The document serves as a guideline for developers and stakeholders, ensuring alignment on project goals and facilitating the delivery of a robust and reliable system.

2. GENERAL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The **Health Care System (HCS)** is a centralized and automated solution that replaces traditional, manual hospital operations. It integrates various functions like patient management, billing, and staff scheduling into a unified platform. The system is scalable and designed to work independently or integrate with external systems such as diagnostic labs, insurance portals, and pharmacies.

2.2 PRODUCT FEATURES

- **Patient Management:** Registration, medical records, and history tracking.
- **Appointment Scheduling:** Streamlined scheduling for doctors and specialists.
- **Billing and Invoicing:** Automated billing with insurance claim processing.
- **Inventory Management:** Real-time tracking of medical supplies and resources.
- **Reports and Analytics:** Generating insights on hospital performance and patient data.

2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

- Must comply with healthcare regulations (e.g., HIPAA, GDPR).
- Require robust security measures for data protection.
- Should support multi-platform accessibility (web, desktop, mobile).
- Hardware and software limitations of hospital IT infrastructure may restrict features.

2.4 ASSUMPTIONS AND DEPENDENCIES

- Hospital staff will have basic computer literacy.
- The system will require internet connectivity for external integrations.
- Depends on compatibility with existing hospital systems and third-party services (e.g., diagnostic labs, insurance companies).

3. FUNCTIONAL REQUIREMENTS

Functional requirements define the specific behaviors, operations, and tasks the **Health Care System (HCS)** must perform to meet user and organizational needs.

3.1 DESCRIPTIONS

The HMS must include the following core functional features:

- 1. Patient Management:**
 - Ability to register and maintain patient details, including medical history, demographics, and visit records.
- 2. Appointment Scheduling:**
 - Allow patients to book appointments with doctors or specialists through online or offline systems.

3.2 TECHNICAL ISSUES

The following technical challenges may arise during the implementation of the HMS:

- 1. Data Security and Compliance:**
 - Ensuring patient data privacy and compliance with healthcare regulations (e.g., HIPAA, GDPR).
 - Implementing encryption, secure login systems, and regular audits to prevent data breaches.
- 2. System Integration:**
 - Establishing seamless interoperability with external systems like diagnostic labs, pharmacies, and insurance portals.
- 3. Performance and Scalability:**
 - Designing the system to handle high volumes of data and concurrent users, especially for larger hospitals.
 - Ensuring fast response times during peak operations.
- 4. Customization Requirements:**
 - Balancing the need for standard features with the customization demands of specific hospitals.
- 5. Network Dependency:**
 - Ensuring system functionality in environments with unstable or limited internet connectivity.

6. User Adoption and Training:

- Addressing potential resistance to change and providing adequate training for hospital staff to use the system effectively.

4. INTERFACE REQUIREMENTS

This section defines the interactions and interfaces necessary for the Hospital Management System (HMS) to function effectively. It includes requirements for user interaction, hardware compatibility, software integration, and communication protocols.

4.1 USER INTERFACE

- The HCS must have an intuitive, user-friendly interface to cater to various users, including hospital administrators, doctors, nurses, and patients.
 - **Features:**
 - Clear navigation menus for accessing modules (e.g., patient records, billing, inventory).

- Role-based dashboards (e.g., separate views for doctors, staff, and patients).
- Support for multiple languages based on user demographics.
- Accessibility features such as large fonts and screen reader compatibility.
- **Devices Supported:**
 - Desktop systems for administrators.
 - Tablets and mobile devices for doctors and nurses.
 - Web portals or apps for patient access to appointments and medical history.

4.2 HARDWARE INTERFACE

- The HCS must integrate seamlessly with the hospital's hardware infrastructure.
 - **Requirements:**
 - Compatibility with desktops, laptops, and tablets used by staff.
 - Support for hardware peripherals such as:
 - Barcode scanners for patient wristbands or inventory tracking.
 - Printers for generating bills, prescriptions, and reports.
 - Biometric devices for secure staff and patient authentication.
 - Servers with sufficient capacity to handle data storage and processing.

4.3 SOFTWARE INTERFACE

- The HCS must integrate with existing software systems and third-party applications.
 - **Requirements:**
 - Interoperability with electronic health record (EHR) systems.
 - Integration with accounting software for billing and financial reports.
 - API support for communication with:
 - Diagnostic lab systems for test results.
 - Pharmacy systems for prescription management.
 - Insurance portals for claim processing.

- Compatibility with standard operating systems (Windows) and databases (MySQL).

4.4 COMMUNICATION INTERFACE

- The HMS must support secure and efficient communication between systems and users.
 - **Requirements:**
 - **Internal Communication:**
 - Secure messaging for staff coordination.
 - Notifications for task assignments or alerts (e.g., low inventory, upcoming appointments).
 - **External Communication:**
 - Sending appointment reminders to patients via email or SMS.
 - Real-time data exchange with external systems like labs and insurance providers.
 - **Protocols and Standards:**
 - Support for healthcare standards like HL7 and FHIR for data exchange.

5. SOFTWARE REQUIREMENT ANALYSIS

This section provides an in-depth analysis of the software requirements for the **Health Care System (HMS)** by defining the problem it addresses and detailing the modules and their functionality.

5.1 DEFINE PROBLEM

The HMS is designed to address the following key challenges faced by hospitals:

1. **Inefficient Manual Processes:**
 - Paper-based systems for managing patient records, appointments, and billing are time-consuming, error-prone, and difficult to maintain.
2. **Data Accessibility Issues:**
 - Limited access to medical records, appointment schedules, and inventory levels leads to delays in decision-making and patient care.
3. **Lack of Integration:**

- Disconnected systems for billing, inventory, and diagnostics make it challenging to provide a seamless operational flow.
- 4. Security Concerns:**
 - Manual systems and outdated software often lack robust security measures to protect sensitive patient and hospital data.
- 5. Compliance with Standards:**
 - Heath Cares face difficulty in meeting regulatory requirements (e.g., HIPAA, GDPR) due to the absence of centralized and auditable systems.

5.2 DEFINE MODULES AND FUNCTIONALITY

The HCS is divided into several modules, each addressing specific operational needs:

1. Patient Management Module

- **Functionality:**
 - Registration of patient details, medical history, and contact information.
 - Management of inpatient and outpatient records.

2. Appointment Scheduling Module

- **Functionality:**
 - Schedule appointments with doctors and specialists.
 - Notify patients via SMS or email about appointment confirmations or changes.

3. Staff Management Module

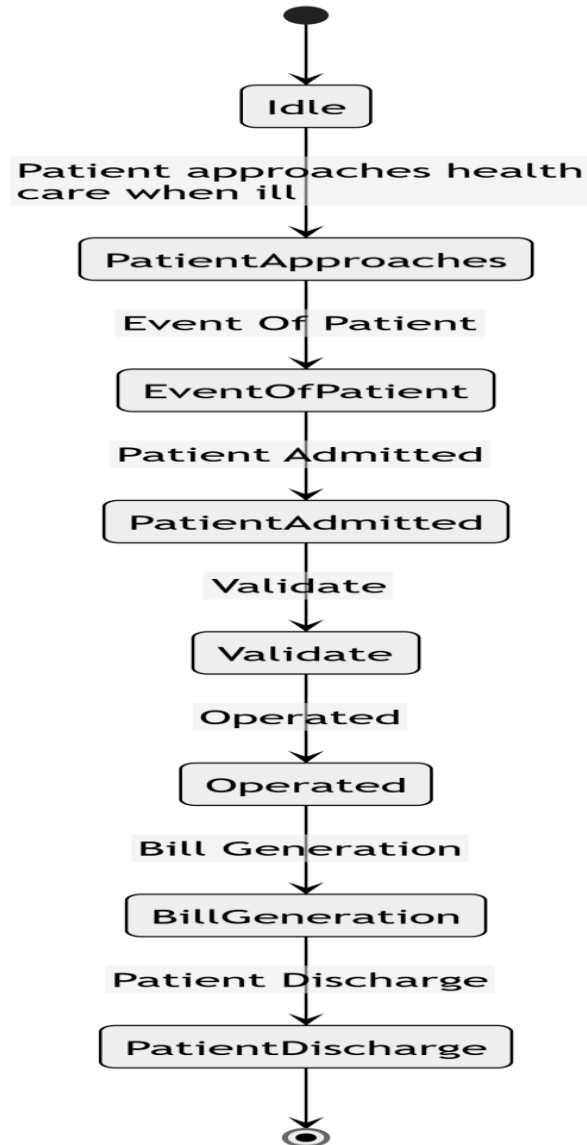
- **Functionality:**
 - Maintain staff records, schedules, and attendance.
 - Facilitate payroll generation based on working hours.

4. Security and Access Control Module

- **Functionality:**
 - Implement role-based access control to ensure data confidentiality.
 - Monitor user activity and maintain audit trails for compliance.

6. SOFTWARE DESIGN

STATE DIAGRAM



7. NON-FUNCTIONAL REQUIREMENTS

7.1 PERFORMANCE

- Support up to 500 concurrent users with response times under 2 seconds for critical operations.
- Handle large datasets efficiently and scale with a 30% annual increase in user load.

7.2 SECURITY

- Ensure data encryption with TLS 1.3 or higher and implement multi-factor authentication.
- Comply with HIPAA or equivalent standards, enforce role-based access control, and maintain audit logs.

7.3 RELIABILITY

- Achieve 99.9% uptime with automatic error detection and data recovery within 30 minutes of failure.

7.4 AVAILABILITY

- Operate 24/7 with failover mechanisms and minimal planned maintenance downtime.

7.5 SAFETY

- Safeguard data integrity, prevent accidental data loss, and display alerts for critical issues to avoid patient harm.

7.6 SOFTWARE QUALITY

- Ensure 95% code coverage in testing, adhere to software quality standards, and maintain high user satisfaction.

7.7 REUSABILITY

- Design modular components and APIs to enable integration and reuse in other systems.

7.8 MAINTENANCE

- Support remote diagnostics, provide updated documentation, and deploy critical bug fixes within 48 hours.

CONCLUSION

In conclusion, this document provides a comprehensive Software Requirements Specification (SRS) for the Health Care System. It outlines the purpose, scope, and

overview of the project, emphasizing its critical role in streamlining healthcare operations.

The general description elaborates on the product's perspective, features, design constraints, and dependencies, setting a strong foundation for the system's development. Functional requirements are detailed to ensure clarity in system behavior, while interference requirements address user, hardware, software, and communication interactions.

The software requirement analysis identifies key problems, defines modules, and specifies functionalities to achieve a cohesive solution. The software design section provides state, class, use case, sequential, and database diagrams for a structured approach to implementation.

Non-functional requirements, including performance, security, reliability, availability, and maintenance, ensure that the system is robust, secure, and maintainable. This document serves as a blueprint for building a high-quality system tailored to meet the needs of a modern healthcare environment.

REFERENCES

1. Sommerville, Ian. *Software Engineering*. 10th Edition, Pearson, 2015.
2. Pressman, Roger S. *Software Engineering: A Practitioner's Approach*. 8th Edition, McGraw-Hill, 2014.
3. ISO/IEC 25010:2011, *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*.
4. HIPAA Compliance Guidelines, U.S. Department of Health & Human Services. Retrieved from <https://www.hhs.gov/hipaa>