
SOFTWARE REQUIREMENTS SPECIFICATION

for

EMR SYSTEM

Version 1.2

Prepared by : 1. Aser Osama (202101266)
2. Aya Shereif (202100642)
3. Mustafa Muhhamed (202202211)
4. Ahmed Khaled (202201755)

Submitted to : Hanan Moussa
Professor

December 25, 2024

1 Version History

Ver.	Date	Author	Change Description
1.0	Dec 20, 2024	Aser Osama	Initial SRS Created
1.1	Dec 24, 2024	Aya Shereif	Added new Functional Reqs
1.2	Dec 25, 2024	Aya Shereif	Added new Non-Functional Reqs

Contents

1	Version History	2
2	Introduction	5
2.1	Purpose	5
2.2	Intended Audience and Reading Suggestions	5
2.3	Project Scope	5
2.4	System Features Overview	6
3	Use Cases	7
3.1	Use Case 1: Add Patient Health Data	7
3.2	Use Case 2: Request Tests/Imaging	8
3.3	Use Case 3: Generate Patient Reports	8
3.4	Use Case 4: View Patient Records	9
3.5	Use Case 5: Prescribe Medication	10
3.6	Use Case 6: Give Patient Access	11
3.7	Use Case 7: Update Patient Records with Limitation	11
3.8	Use Case 8: View Patient Records	12
3.9	Use Case 9: Register New Patient	13
3.10	Use Case 10: View Medication Requests	14
3.11	Use Case 11: Verify and Dispense Medication	14
3.12	Use Case 12: View Test Requests from Doctors	15
3.13	Use Case 13: Show Patient Data Related to Tests	16
3.14	Use Case 14: Perform and Upload Test Results	17
3.15	Use Case 15: View Imaging Requests from Doctors	18
3.16	Use Case 16: Upload Imaging Results	19
3.17	Use Case 17: Upload Radiology Report	19
3.18	Use Case 18: Share Patient Records	20
3.19	Use Case 19: Manage User Roles and Permissions	21
3.20	Use Case 20: Show Patient Reports	22
3.21	Use Case 21: Patient Health Data Logging	23
3.22	Use Case 22: Generate Financial Reports	24
3.23	Use Case 23: Notify Departments of Outstanding Payments	24
3.24	Use Case 24: Automate Medical Insurance Claims Processing	25
3.25	Use Case 25: Order Inventory for Pharmacy	26
3.26	Use Case 26: Review Inventory Costs	27

4	User Journeys	28
5	Priority Criteria	32
5.1	Criteria of prioritization	32
5.2	Priority of our Functional Requirements	33
6	Requirements Traceability Matrix (RTM)	39
7	NFRs	45
7.1	Performance	45
7.2	Scalability	45
7.3	Availability	45
7.4	Security	45
7.5	Usability	46
7.6	Reliability	46
7.7	Legislative Requirements	46

2 Introduction

2.1 Purpose

Maintaining patient, staff, and administrative data in a government hospital can be challenging when relying on manual processes. Data may frequently require updates, additions, or deletions, making the process inefficient. To address these issues, we propose the "Hospital Management System (HMS)" as a comprehensive solution. HMS is a centralized web-based platform designed to streamline patient records, diagnostics, and administrative workflows, ensuring efficiency, accuracy, and compliance with healthcare regulations.

2.2 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- **Developers:** To understand system functionalities and technical requirements.
- **Project Managers:** To oversee the project scope and progress.
- **End Users:** Hospital staff including doctors, nurses, and administrative personnel.
- **Testers:** To validate the functionalities and ensure compliance with requirements.

Readers are advised to refer to the functional and non-functional sections for specific requirements, the use case diagrams for system workflows, and the requirements traceability matrix for testing and validation.

2.3 Project Scope

The "Hospital Management System (HMS)" aims to modernize patient and hospital management processes by digitizing record-keeping and automating essential tasks. Key features of the system include:

- Registration of new patients and the assignment of unique identifiers.
- Logging and updating patient health metrics, test results, and imaging data.
- Managing user roles and permissions for doctors, nurses, and administrative staff.

- Automating financial tasks such as billing, reporting, and insurance claims processing.
- Enabling seamless data sharing between departments and external institutions while maintaining strict security protocols.

2.4 System Features Overview

The HMS system provides the following high-level features:

1. **Patient Management:** Registration, record maintenance, and access to health data.
2. **Doctor Tools:** Assigning courses, viewing patient records, prescribing medication, and uploading test results.
3. **Administrative Management:** Managing user roles, permissions, and overall hospital operations.
4. **Data Security and Compliance:** Adherence to GDPR, HIPAA, and local healthcare laws for secure data handling.

3 Use Cases

3.1 Use Case 1: Add Patient Health Data

Actor: Doctor

Goal: To add or update detailed information about the medical history of the patient.

Preconditions:

- The doctor is logged into the system.
- The patient is already registered in the system.

Main Flow:

1. The doctor searches for the patient by ID or name.
2. The system displays the patient's record.
3. The doctor selects the "Add Data" option.
4. The system prompts the doctor to select the type of illness from a predefined list (e.g., Diabetes, Hypertension, Asthma).
5. The doctor selects the illness type.
6. The system dynamically generates questions about the selected illness (e.g., for Diabetes: Blood Sugar Level, Medication Taken Today, Diet Details).
7. The doctor is presented with these questions.
8. The doctor answers the displayed questions and optionally adds additional notes.
9. The doctor confirms or edits the details.
10. The doctor submits the information.
11. The system updates the patient's record and logs the changes for future reference.
12. **The system adds a bill profile for the user**

Alternative Flows:

- **Patient Not Found:**

- The doctor may ask the receptionist to register the patient.

Illness Type Not Available:

- If the required illness type is not listed, the doctor selects “Other” and enters illness details manually.

Postconditions:

- The patient’s data is updated and logged for future reference.

3.2 Use Case 2: Request Tests/Imaging

Actor: Doctor

Goal: To request diagnostic tests or imaging for a patient.

Preconditions:

- The doctor is logged into the system.
- The patient’s record exists in the system.

Main Flow:

1. The doctor opens the patient’s record.
2. The doctor selects the “Request Test/Imaging” option.
3. The doctor chooses the required tests or imaging from a predefined list or chooses a custom one if it doesn’t exist.
4. The doctor provides any additional notes for the test or imaging department.
5. The doctor submits the request.

Alternative Flows:**- Urgent Request:**

- For urgent cases, the doctor marks the request as high priority.
- The system sends an immediate notification to the relevant department.

Postconditions:

- The test or imaging request is sent to the respective department.
- The department is notified, and the request is queued for processing.

3.3 Use Case 3: Generate Patient Reports

Actor: Doctor

Goal: Create a detailed report summarizing a patient's condition, diagnosis, and treatment plan.

Preconditions:

- The doctor is logged into the system.
- The patient's data is up-to-date in the system.

Main Flow:

1. The doctor searches for the patient by ID.
2. The doctor accesses the patient's record.
3. The doctor selects the "Generate Report" option.
4. The system displays a report template.
5. The doctor inputs observations, diagnoses, and treatment plans.
6. The doctor reviews and saves the report.

Postconditions:

- The report is saved in the patient's record and is accessible for future reference.

3.4 Use Case 4: View Patient Records

Actor: Doctor

Goal: To view a patient's medical history and current information.

Preconditions:

- The doctor is logged into the system.
- The patient's record exists in the system.

Main Flow:

1. The doctor searches for the patient by ID.
2. The doctor accesses the patient's record.
3. The doctor reviews medical history, current condition, test results, and prescribed medications.
4. The doctor can optionally filter the data to focus on specific time periods or details.

Alternative Flows:

- **Patient Record Not Found:**

- If the record is missing, the system suggests alternative search criteria (e.g., name spelling variations).
- If not found the Dr ask the receptionist to add the patient.

Postconditions:

- The doctor has accessed the necessary patient information.

3.5 Use Case 5: Prescribe Medication

Actor: Doctor

Goal: To prescribe medication for a patient.

Preconditions:

- The doctor is logged into the system.
- The patient's record exists in the system.

Main Flow:

1. The doctor searches for the patient by ID.
2. The doctor accesses the patient's record.
3. The doctor selects the "Prescribe Medication" option.
4. The doctor prescribes the appropriate treatment.
5. The doctor selects the option "Verify Medication".
6. The doctor specifies dosage, frequency, and duration.
7. The doctor submits the prescription.

Alternative Flows:

- Drug Interaction Warning:

- If the system detects a potential interaction with existing medications, it alerts the doctor.
- The doctor reviews and adjusts the prescription if needed.

Patient Allergies Detected:

- If the patient's record indicates an allergy to the prescribed drug, the system alerts the doctor, who selects an alternative.

Postconditions:

- The prescription is saved in the patient's record.

- The pharmacy is notified to dispense the medication.

3.6 Use Case 6: Give Patient Access

Actor: Doctor

Goal: To grant a patient access to specific parts of their medical record.

Preconditions:

- The doctor is logged into the system.
- The patient is registered and has an account in the system.

Main Flow:

1. The doctor accesses the patient's record.
2. The doctor selects the "Manage Patient Access" option.
3. The doctor specifies which parts of the record that the patient can add data to.
4. The doctor saves the access settings.

Alternative Flows:

- **Access Permissions Error:**

- If the doctor does not have permission to modify patient access, the system denies the request and notifies the admin.

Access Settings Conflict:

- If access settings conflict with hospital policies, the system suggests permitted options to the doctor.

Postconditions:

- The patient is granted access to update their own records.

3.7 Use Case 7: Update Patient Records with Limitation

Actor: Nurse

Goal: To update certain allowed criteria for a patient (BP, HR, Blood sugar, etc.).

Preconditions:

- The nurse is logged into the system.
- The patient is already registered in the system.

Main Flow:

1. The nurse searches for the patient by ID or name.
2. The system displays the patient's record.
3. The nurse selects the "Add Data" option.
4. The system opens the list of default things to log for the nurse to pick from (e.g., BP, HR, blood sugar).
5. The nurse confirms or edits the details.
6. The nurse submits the information.
7. The system updates the patient's record and logs the changes for future reference.

Alternative Flows:**- Patient Not Found:**

- The nurse may ask the receptionist to register the patient.

Postconditions:

- The patient's data is updated and logged for future reference.

3.8 Use Case 8: View Patient Records

Actor: Nurse

Goal: To view a patient's medical history and current information.

Preconditions:

- The nurse is logged into the system.
- The patient's record exists in the system.

Main Flow:

1. The nurse searches for the patient by ID.
2. The nurse accesses the patient's record.
3. The nurse reviews some medical history, current condition, and prescribed medications.
4. The nurse can optionally filter the data to focus on specific time periods or details.

Alternative Flows:**- Patient Record Not Found:**

- If the record is missing, the nurse asks the receptionist to register the patient.

Postconditions:

- The nurse has accessed the necessary patient information.

3.9 Use Case 9: Register New Patient

Actor: Receptionist

Goal: To create a new patient profile in the system.

Preconditions:

- The receptionist is logged into the system.
- The patient is not already registered in the system.

Main Flow:

1. The receptionist selects the “Register Patient” option.
2. The system displays the registration form.
3. The receptionist enters the patient’s details, including but not limited to:
 - Full Name
 - National ID (It Will be the way to access his records)
 - Date of Birth
 - Gender
 - Contact Information
 - Address
4. The receptionist reviews and submits the form.
5. The system validates the information and creates a new patient record.

Alternative Flows:

- Duplicate Patient Found:

- The system alerts the receptionist to review potential duplicates.

Postconditions:

- A new patient record is created and stored in the system.

3.10 Use Case 10: View Medication Requests

Actor: Pharmacy Staff

Goal: To view and process medication requests submitted by doctors.

Preconditions:

- The pharmacy staff is logged into the system.
- Medication requests have been submitted by doctors.

Main Flow:

1. The pharmacy staff selects the “View Medication Requests” option from the system menu.
2. The system displays a list of pending medication requests, sorted by priority or request time.
3. The pharmacy staff selects a specific medication request to view details.
4. The system displays the request details, including but not limited to:
 - Patient Information (Name, ID, Contact Details)
 - Prescribed Medications (Name, Dosage, Quantity)
 - Doctor’s Notes or Special Instructions
 - Request Priority (Normal/Urgent)
5. The pharmacy staff reviews the request and marks it as “In Progress.”

Alternative Flows:

- **No Pending Requests:**

- If there are no pending medication requests, the system displays a message indicating this.

Postconditions:

- The selected request is marked as “In Progress” for further processing.
- The system logs the activity for tracking purposes.

3.11 Use Case 11: Verify and Dispense Medication

Actor: Pharmacy Staff

Goal: To verify and dispense medications based on a doctor’s request.

Preconditions:

- The pharmacy staff is logged into the system.
- The medication request has been reviewed and marked as “In Progress.”

Main Flow:

1. The pharmacy staff selects a medication request marked as “In Progress.”
2. The pharmacy staff verifies the availability of the prescribed medications in stock.
3. If all medications are available, the pharmacy staff prepares the medications for dispensing .
4. The pharmacy staff updates the system with the dispensing details, including:
 - Quantity Dispensed
 - Dispensing Date and Time
 - Alternative used if any
5. The pharmacy staff generates a dispensing slip for the patient.
6. **The pharmacy staff adds a bill to be confirmed by the finance office.**
7. The pharmacy staff marks the request as “Completed” in the system.

Alternative Flows:

- Medication Not Available:

- If any medication is out of stock:
 - The pharmacy staff updates the request with an “Out of Stock” status.
 - The system notifies the doctor of the unavailability.
 - the Doctor send another request with an equivalent medicine and notes it on the system

Request Cancelled:

- If the request is cancelled by the doctor, the system removes it from the “In Progress” list and notifies the pharmacy staff.

Postconditions:

- The medications are dispensed, and the request is marked as “Completed.”
- The system updates the patient’s record with the dispensing details.

3.12 Use Case 12: View Test Requests from Doctors

Actor: Lab Technician

Goal: The Lab Technician views test requests submitted by doctors for patients.

Preconditions:

- The doctor has submitted a test request in the system.
- The Lab Technician has logged into the system and has permission to view test requests.

Main Flow:

1. The Lab Technician logs into the system.
2. The Lab Technician navigates to the “Test Requests” section.
3. The Lab Technician selects the test request to view detailed information.
4. The Lab Technician can search for requests using the patient’s name, test type, or date.
5. The Lab Technician confirms the test details so that preparation can proceed.

Alternative Flows:

- Test Request Not Found:

- If no test requests are found, the system displays an error message.
- The Lab Technician can attempt a new search using different criteria (e.g., patient ID or test type).
- If no request is found, the Lab Technician contacts the doctor for clarification or further instructions.

Postconditions:

- The Lab Technician successfully views the requested test details and is ready to proceed with the test preparation.

3.13 Use Case 13: Show Patient Data Related to Tests

Actor: Lab Technician

Goal: The Lab Technician retrieves the necessary patient data to perform tests.

Preconditions:

- The patient is registered in the system.
- The Lab Technician is authorized to access patient data.

Main Flow:

1. The Lab Technician accesses the patient data system.
2. The Lab Technician enters the patient's ID or search criteria.
3. The system retrieves and displays the patient's medical records.
4. The Lab Technician reviews the patient's data to ensure accuracy and prepare for the test.

Alternative Flows:

- Patient Data Not Available:

- If the system cannot retrieve patient data, the Lab Technician is notified of the issue.
- The Lab Technician may contact the medical records department to retrieve the data.
- If data is still unavailable, the Lab Technician may consult with the doctor to proceed with alternative test plans.

Postconditions:

- The Lab Technician has the necessary patient information to perform the test accurately.

3.14 Use Case 14: Perform and Upload Test Results

Actor: Lab Technician

Goal: The Lab Technician performs a test and uploads the results to the system for review by the doctor.

Preconditions:

- A test has been requested by the doctor.
- The necessary test equipment is available and functional.

Main Flow:

1. The Lab Technician enters the patient's ID or search criteria.
2. The Lab Technician records the test results into the system manually or uploads them via connected equipment.
3. The Lab Technician confirms the upload and submits the results.
4. The results are made available to the doctor for review.

Alternative Flows:

- Test Not Available:

- The Lab Technician writes a note for the doctor, and the doctor asks the patient to make the test in another place.
- The doctor will upload the results after the patient comes again.

Postconditions:

- The test results are uploaded and available for the doctor to access.

3.15 Use Case 15: View Imaging Requests from Doctors

Actor: Radiology Technician

Goal: The Radiology Technician views imaging requests submitted by doctors for X-ray procedures.

Preconditions:

- The doctor has submitted the imaging request in the system.
- The Radiology Technician has logged into the system and is authorized to view the request.

Main Flow:

1. The Radiology Technician logs into the system.
2. The Radiology Technician accesses the “Imaging Requests” section.
3. The Radiology Technician views the list of requests or searches by patient name, test type, or imaging request ID.
4. The Radiology Technician selects an imaging request to view detailed information.
5. The Radiology Technician verifies the imaging request details before proceeding.
6. **The Radiology Technician adds a bill to be confirmed by the finance office.**

Alternative Flows:

- Imaging Request Incomplete or Invalid:

- If the imaging request is incomplete or unclear (e.g., missing patient data or test specifications), the Radiology Technician contacts the doctor for clarification.
- If no response is received in time, the Radiology Technician escalates the issue to a supervisor or manager for next steps.

Postconditions:

- The Radiology Technician has all the necessary information to proceed with performing the imaging.

3.16 Use Case 16: Upload Imaging Results

Actor: Radiology Technician

Goal: The Radiology Technician performs the X-ray imaging as per the doctor's request.

Preconditions:

- The imaging request has been verified.
- The X-ray machine is functional and available.

Main Flow:

1. The Radiology Technician enters the patient's ID or search criteria.
2. The Radiology Technician checks the quality of the images for accuracy.
3. The Radiology Technician saves the X-ray images for upload.
4. The Radiology Technician uploads the image to the system.

Alternative Flows:

- Imaging Not Available:

- The Radiology Technician writes a note for the doctor, and the doctor asks the patient to make the imaging in another place.
- The doctor will upload the results after the patient comes again.

Postconditions:

- The X-ray images are uploaded into the system.
- The referring doctor is notified of the upload.
- The radiology order status is updated as completed.

3.17 Use Case 17: Upload Radiology Report

Actor: Radiology Technician

Goal: Upload completed radiology images and reports for review by the referring doctor.

Preconditions:

- The X-ray or imaging has been completed.
- The patient record is accessible.

Main Flow:

1. The system displays pending radiology orders.
2. The Radiology Technician selects the appropriate patient order from the list.
3. The Radiology Technician enters report findings and observations.
4. The system saves the report to the patient's record with its images.
5. The system notifies the referring doctor.
6. The system marks the radiology order as completed.

Alternative Flows:

- Upload Error:

- If the upload fails, the system displays an error message.
- The Radiology Technician retries the upload or adjusts the file format/size as suggested by the system.

Postconditions:

- The radiology report is securely stored in the patient record.
- The referring doctor is notified of the upload.
- The radiology order status is updated as completed.
- The report is available for review in the system.

3.18 Use Case 18: Share Patient Records

Actor: Administrator

Goal: Securely share patient records with another hospital to ensure continuity of care.

Preconditions:

- Admin has sharing permissions.
- Patient records are complete and exist in the system.
- The receiving hospital is registered in the system.

Main Flow:

1. Admin navigates to the patient record sharing section.

2. Admin searches for and selects the patient record(s) to share.
3. Admin selects the receiving hospital from the registered hospitals list.
4. The system validates sharing permissions and data completeness.
5. The system encrypts the selected records.
6. The system notifies the receiving hospital.
7. The system logs the sharing transaction.

Alternative Flows:

- Patient Record Not Found:

- The system displays “Record Not Found.”
- The admin modifies search criteria.

Postconditions:

- Patient records are securely shared.
- The receiving hospital is notified.

3.19 Use Case 19: Manage User Roles and Permissions

Actor: Administrator

Goal: Manage and configure user roles and permissions to restrict data access appropriately.

Preconditions:

- Admin is logged into the system.
- Admin has user management permissions.

Main Flow:

1. Admin navigates to the user management section.
2. The system displays a list of current users.
3. Admin selects a user or creates a new user.
4. Admin assigns or modifies the user’s role (e.g., Doctor, Nurse, Staff).
5. Admin saves the role configuration.
6. The system updates the user’s access rights.

Alternative Flows:**- New User Creation:**

- The admin enters user details.
- The system validates the email format.
- The system generates a temporary password.

Postconditions:

- User roles are updated.
- System access is aligned with new permissions.

3.20 Use Case 20: Show Patient Reports

Actor: Administrator

Goal: Generate and view patient reports for hospital performance analysis and decision-making.

Preconditions:

- Admin has reporting permissions.
- Patient data exists in the system.

Main Flow:

1. Admin navigates to the reporting section.
2. Admin selects the type of report (e.g., Medical History, Treatment Plans).
3. Admin specifies report parameters and the time period.
4. The system validates data availability.
5. The system generates the report with visualizations.
6. The system displays a report preview.
7. Admin exports or shares the report.

Alternative Flows:**- Incomplete Data:**

- The system shows a missing data warning.
- Admin generates a partial report.

Postconditions:

- The report is generated.
- The report is available for sharing or export.

3.21 Use Case 21: Patient Health Data Logging

Actor: Patient

Goal: Record and monitor health metrics like blood pressure and sugar levels.

Preconditions:

- The patient has an active account.
- The patient is logged into the system.
- The patient has a condition requiring monitoring.

Main Flow:

1. Patient accesses their health portal.
2. The system displays condition-specific data entry forms.
3. The patient enters measurements with a timestamp.
4. The system validates data ranges.
5. The system saves the data to the patient record.
6. The system confirms the data entry.

Alternative Flows:

- **Connection Issues:**

- The system saves data locally.
- The system syncs data when the connection is restored.

Postconditions:

- Health data is recorded.
- The healthcare team is notified if necessary.

New Requirements:

3.22 Use Case 22: Generate Financial Reports

Actor: Finance Manager

Goal: Generate comprehensive financial reports combining data from the MIS and financial system for performance tracking and decision-making.

Preconditions:

- The MIS and financial system are integrated.
- Financial data and patient billing records are synchronized.
- The finance manager has reporting permissions.

Main Flow:

1. The finance manager navigates to the reporting dashboard.
2. The finance manager selects a report type (e.g., Revenue Breakdown, Cost Analysis).
3. The system retrieves data from both the MIS and financial system.
4. The system merges and formats the data for consistency.
5. The system generates a detailed report with visualizations.
6. The system displays the report preview.
7. The finance manager exports or shares the report.

Alternative Flows:

- **Data Retrieval Delay:**

- The system notifies the user of a delay.
- The finance manager chooses to generate a partial report or retry later.

Postconditions:

- Financial reports are generated and available for analysis or sharing.
- Data from both systems is reflected in the report.
- The report generation is logged on the financial system.

3.23 Use Case 23: Notify Departments of Outstanding Payments

Actor: System

Goal: Automatically notify relevant hospital departments of outstanding payments to ensure timely action.

Preconditions:

- The MIS is integrated with the financial system.
- Outstanding payment data exists in the financial system.
- Departments are registered in the system with valid contact information.

Main Flow:

1. The system queries the financial system for outstanding payments once a week.
2. The system identifies relevant departments based on payment categories (e.g., Pharmacy, Radiology).
3. The system generates notifications detailing the outstanding payments.
4. The system sends notifications to the respective departments via email or dashboard alerts.
5. The system logs the notification activity for audit purposes.

Alternative Flows:

- **Notification Failure:**

- The system retries sending notifications.
- If the issue persists, the system alerts the system administrator.

3.24 Use Case 24: Automate Medical Insurance Claims Processing

Actor: Insurance Administrator

Goal: Automate the processing of medical insurance claims by integrating data from the MIS and the financial system.

Preconditions:

- Insurance claim records are available in the MIS.
- The financial system tracks insurance payouts and pending claims.
- The insurance administrator has access permissions.

Main Flow:

1. The insurance administrator accesses the claims automation module.
2. The system retrieves claim data from the MIS.
3. The system validates claim details against financial records.
4. The system prepares and submits claims to the insurance company via API.

5. The system logs claim submissions and monitors their status.
6. The insurance administrator reviews claim progress and updates as needed.
7. The insurance company confirms or denies the claim

Alternative Flows:

- Claim Validation Error:

- The system flags invalid claims and notifies the administrator.
- The administrator corrects errors and resubmits the claims.

Postconditions:

- Insurance claims are processed efficiently.
- Claim status is updated in real-time.

3.25 Use Case 25: Order Inventory for Pharmacy

Actor: Pharmacist

Goal: Place inventory orders for the pharmacy to ensure stock availability and avoid shortages.

Preconditions:

- The pharmacy inventory is tracked in the MIS.
- The pharmacist has ordering permissions.
- Approved suppliers are registered in the system.

Main Flow:

1. The pharmacist navigates to the pharmacy inventory section.
2. The system displays current stock levels and reorder thresholds.
3. The pharmacist selects items to reorder and specifies quantities.
4. The system suggests approved suppliers for each item.
5. The pharmacist confirms the order and submits it.
6. The system sends the order to the selected suppliers and logs the transaction.

Alternative Flows:

- Supplier Not Available:

- The system notifies the pharmacist.
- The pharmacist manually selects an alternative supplier or defers the order.

Postconditions:

- Inventory orders are placed with approved suppliers.

- The pharmacy inventory is updated with expected stock arrivals.

3.26 Use Case 26: Review Inventory Costs

Actor: Inventory Manager

Goal: Review and confirm inventory costs recorded in the MIS and financial system.

Preconditions:

- Inventory records exist in the MIS.
- Cost data is available in the financial system.

Main Flow:

1. The Inventory manager accesses the inventory cost section.
2. The system displays inventory records alongside corresponding costs.
3. The Inventory manager verifies the records and confirms accuracy.
4. If needed, the manager updates the data in the system.

Alternative Flows:

- **Cost Discrepancy Found:**

- The system flags the issues.
- The Inventory manager resolves the issue and saves updates.

Postconditions:

- Inventory costs are reviewed and confirmed.

4 User Journeys

Doctor's Journey

- Log In
- Search for a Patient
- View Patient Records
- Add Patient Data
- Generate Patient Reports
- Request Tests/Imaging
- Prescribe Medication
- Give Patient Access
- Log Out

Nurse's Journey

- Log In
- Search for a Patient
- Update Patient Records
- View Patient Records with Limitations
- Log Out

Lab Technician's Journey

- Log In
- View Test Requests from Doctors
- Show Patient Data Related to Tests
- Upload Test Results
- Notify the Doctor
- Log Out

Radiology Technician's Journey

- Log In
- View Imaging Requests from Doctors
- Access Patient Data Related to Imaging
- Upload Imaging Results
- Upload Imaging Report
- Notify the Doctor
- Log Out

Receptionist Journey

- Register New Patient
- Patient Arrival
- Provide Details
- Data Entry
- Verify Details
- Confirm Registration

Admin's Journey

- Log In
- Share Patient Records
- Manage User Roles
- Generate Patient Reports
- Log Out

Patient's Journey

- Log In
- Log Health Data
- Log Out

Finance Manager's Journey

- Log In
- Navigate to Reporting Dashboard
- Select Report Type
- Generate Financial Reports
- Export or Share Report
- Log Out

System's Journey

- Query Outstanding Payments
- Identify Relevant Departments
- Generate Notifications
- Send Notifications
- Log Notification Activity

Insurance Administrator's Journey

- Log In
- Access Claims Automation Module
- Validate Claim Details
- Submit Claims to Insurance Company
- Monitor Claim Status
- Log Out

Pharmacist's Journey

- Log In
- Navigate to Pharmacy Inventory Section
- Select Items to Reorder
- Confirm and Submit Order
- Log Out

Inventory Manager's Journey

- Log In
- Access Inventory Cost Section
- Review Inventory Costs
- Confirm Accuracy or Resolve Discrepancies
- Save Updates
- Log Out

5 Priority Criteria

5.1 Criteria of prioritization

Value – This approach focuses on the business benefit of any given requirement; the requirements that will return the greatest business or economic value are given the highest priority. This focus on value helps to ensure “quick wins” for the organization.

Cost – With an eye toward funding, this approach may be implemented in a number of ways—implementing the least expensive requirements first or first implementing requirements with the greatest ROI (return on investment) (profit/investment).

Risk – This approach prioritizes the riskiest requirements first, with the logic that should they fail, the project can be abandoned with a minimum of investment. This approach often makes sense when a controversial or untested initiative is planned.

Regulatory Compliance – With this approach, the requirements that are needed to meet legal and/or regulatory requirements are given the highest priority.

Relationship to Other Requirements / Dependency – Requirements often intermingle in complex relationships of interdependence. With this approach, requirements that support other high-priority requirements are also given high priority.

5.2 Priority of our Functional Requirements

Funct. Reqs	Value	Cost	Risk	Comp.	Dep.	Just.	Tot. Pri.
Add Patient Health Data	5	3	2	5	5	This is the base of the entire application. It must be implemented first to enable other features. Since it handles private and sensitive data, strict compliance with data protection laws is essential. The technical risk is moderate-low since data handling practices are well known.	20
Request Test- s/Imaging	2	3	3	3	4	This feature supports diagnosis and treatment. It needs to be operational soon after basic patient data is established as other functionality rely on it. However, it is not completely valuable to our customer as they have systems in place for this which make it less of a priority to complete. Compliance with medical standards is not too high as it is a custom system, while technical complexity is moderate.	15
Generate Patient Reports	3	2	2	3	3	This feature summarizes patient data for easy access. It is less critical operationally but supports audits and legal documentation. Low risk and dependency make it easier to implement.	13

Funct. Reqs	Value	Cost	Risk	Comp.	Dep.	Just.	Tot. Pri.
View Patient Records	5	3	4	5	4	Provides essential access to patient data for healthcare providers such as doctors or insurance workers. Compliance with data access policies is mandatory. Higher technical complexity due to secure data retrieval and high risk of mistakes.	21
Prescribe Medication	4	3	4	5	4	Enables accurate treatment plans and medication safety and an easier way to track the patients medicine than relying on them remembering it, making it an essential feature. Compliance with drug prescription regulations is essential. Moderate technical risk due to potential system integration challenges.	20
Give Patient Access	5	2	2	3	2	Increases transparency and engagement by allowing patients to access their records. It is beneficial but less critical according to the stakeholders.	14
Update Patient Records with Limitation	5	3	3	5	3	Allows for updates while minimizing risks through access control for nurses or technicians. Ensures the system stays up-to-date without compromising security. Needs to comply with data privacy laws.	19

Funct. Reqs	Value	Cost	Risk	Comp.	Dep.	Just.	Tot. Pri.
Register New Patient	5	3	3	5	5	Establishes the foundation for creating patient records. Compliance and moderate technical dependencies make this a high priority as well as it being a fundamental part of our program.	21
View Medication Requests	3	3	3	4	4	Enables clinicians and pharmacists to review and validate medication plans, improving patient care, however not as important to the customer. Requires adherence to privacy regulations and builds on patient/prescription data.	17
Verify and Dispense Medication	4	3	4	4	2	Ensures correct medication distribution and adherence to regulatory standards, reducing risk of errors. Moderate complexity with strong dependency on core medication request features however with nothing depending on it itself.	17
View Test Requests from Doctors	3	3	3	4	4	Requires secure handling of test orders and depends on patient/test data integration and following of data privacy regulations.	17

Funct. Reqs	Value	Cost	Risk	Comp.	Dep.	Just.	Tot. Pri.
Show Patient Data Related to Tests	3	2	3	5	3	Links patient details with test results, aiding diagnosis accuracy. Low cost and moderate risk, leveraging existing patient and test data infrastructures however high compliance to data privacy laws since what technicians see is very limited compared to doctors so should be handled with care.	16
Perform and Upload Test Results	5	3	4	5	3	Ensures availability of accurate test results and reduces risk of test result loss due to paper so very important to customer. High compliance and security needs, with moderate complexity and strong dependency on patient data but nothing relies on it itself.	20
View Imaging Requests from Doctors	4	3	4	4	3	Ensures timely communication of imaging requests. Requires secure integration and depends on foundational patient/imaging workflows but with few dependencies of its own.	18
Upload Imaging Results	4	3	4	4	2	Provides prompt access to imaging outcomes, enhancing patient care decisions. Strict compliance, moderate complexity, dependent on imaging requests and patient records.	17

Funct. Reqs	Value	Cost	Risk	Comp.	Dep.	Just.	Tot. Pri.
Upload Radiology Report	5	3	4	4	2	Completes the patient's diagnostic picture with radiological findings. High compliance standards, moderate complexity, and depends on core imaging infrastructures. No dependencies but fundamental feature to our program	18
Share Patient Records	5	3	5	5	3	Sharing records easily between hospitals is one of the main features required and requested for implementing this system. It has no dependencies and is an independent feature but it is very difficult to implement as it needs to be very safe and secure and ensure no data leakage.	21
Manage User Roles and Permissions	4	3	4	4	4	Protects sensitive data through controlled access. Mandatory for regulatory compliance and data security, relies on established user/patient data systems.	19
Show Patient Reports	3	2	2	3	3	Offers summarized patient information for new doctors or for patients who would like to learn more about their conditions. Low cost and risk, depends on underlying patient data and reporting structures.	13

Funct. Reqs	Value	Cost	Risk	Comp.	Dep.	Just.	Tot. Pri.
Patient Health Data Logging	4	3	3	5	4	Enables continuous monitoring for audits and compliance. Moderate complexity and risk, depends on patient data management frameworks. Data needs to be well secure and encrypted.	19
Generate Financial Reports	4	3	3	4	4	Provides detailed financial insights combining MIS and financial system data. Moderate complexity; depends on integration quality and data synchronization.	18
Notify Departments of Outstanding Payments	3	2	2	3	3	Ensures timely notification of unpaid amounts to relevant departments. Low complexity and risk; automated and periodic operation.	13
Automate Medical Insurance Claims Processing	5	4	4	4	5	Streamlines claims submissions and tracking with insurers. High impact and complexity; depends on accurate data validation and insurer API compatibility.	22
Order Inventory for Pharmacy	3	3	3	4	3	Supports inventory management to avoid shortages. Moderate complexity; depends on supplier integration and inventory tracking accuracy.	16

6 Requirements Traceability Matrix (RTM)

Req ID	Req Descr	Source	Mapped Test Case ID	Status	Priority	Dep.
FR-1	The system must allow doctors to add or update detailed patient data and specify illness types, with related dynamic questions.	Use Case 1 [7], Doctor's Journey [8]	TC-1	Pending	High	None
FR-2	The system must enable doctors to request diagnostic tests or imaging and include notes for relevant departments.	Use Case 2 [7], Doctor's Journey [8]	TC-2	Pending	High	FR-1
FR-3	The system must generate detailed patient reports, including observations, diagnoses, and treatment plans.	Use Case 3 [7], Doctor's Journey [8]	TC-3	Pending	High	FR-1, FR-2

Req ID	Req Descr	Source	Mapped Test Case ID	Status	Priority	Dep.
FR-4	Doctors should view comprehensive patient records, including medical history, test results, and prescriptions.	Use Case 4 [7], Doctor's Journey [8]	TC-4	Pending	Medium	FR-1
FR-5	The system must enable doctors to prescribe medication, including dosage, frequency, and duration, with warnings for allergies or interactions.	Use Case 5 [7], Doctor's Journey [8]	TC-5	Pending	High	FR-4
FR-6	Nurses should update specific patient data like BP, HR, or blood sugar levels.	Use Case 7 [7], Nurse's Journey [8]	TC-6	Pending	Medium	FR-1
FR-7	Lab technicians must view and process test requests and upload results into the system.	Use Cases 12, 14 [7], Lab Tech's Journey [8]	TC-7	Pending	Medium	FR-2

Req ID	Req Descr	Source	Mapped Test Case ID	Status	Priority	Dep.
FR-8	Radiology technicians should view imaging requests, upload imaging results, and submit detailed reports.	Use Cases 15-17 [7], Radiology Journey [8]	TC-8	Pending	Medium	FR-2
FR-9	Patients must log health metrics like blood pressure or sugar levels, with system validation for accuracy.	Use Case 21 [7], Patient's Journey [8]	TC-9	Pending	Low	FR-1
FR-10	Admins must manage user roles and permissions to restrict data access based on roles.	Use Case 19 [7], Admin's Journey [8]	TC-10	Pending	High	None
FR-11	Admins must securely share patient records with external hospitals and maintain logs of these transactions.	Use Case 18 [7], Admin's Journey [8]	TC-11	Pending	Medium	FR-10

Requirement ID	Requirement Description	Source	Mapped Test Case ID	Status	Priority	Dependencies
FR-1	The system must allow doctors to add or update detailed patient data and specify illness types, with related dynamic questions.	Use Case 1 [7], Doctor's Journey [8]	TC-1	Pending	High	None
FR-2	The system must enable doctors to request diagnostic tests or imaging and include notes for relevant departments.	Use Case 2 [7], Doctor's Journey [8]	TC-2	Pending	High	FR-1
FR-3	The system must generate detailed patient reports, including observations, diagnoses, and treatment plans.	Use Case 3 [7], Doctor's Journey [8]	TC-3	Pending	High	FR-1, FR-2
FR-4	Doctors should view comprehensive patient records, including medical history, test results, and prescriptions.	Use Case 4 [7], Doctor's Journey [8]	TC-4	Pending	Medium	FR-1

Requirement ID	Requirement Description	Source	Mapped Test Case ID	Status	Priority	Dependencies
FR-5	The system must enable doctors to prescribe medication, including dosage, frequency, and duration, with warnings for allergies or interactions.	Use Case 5 [7], Doctor's Journey [8]	TC-5	Pending	High	FR-4
FR-6	Nurses should update specific patient data like BP, HR, or blood sugar levels.	Use Case 7 [7], Nurse's Journey [8]	TC-6	Pending	Medium	FR-1
FR-7	Lab technicians must view and process test requests and upload results into the system.	Use Cases 12, 14 [7], Lab Tech's Journey [8]	TC-7	Pending	Medium	FR-2
FR-8	Radiology technicians should view imaging requests, upload imaging results, and submit detailed reports.	Use Cases 15-17 [7], Radiology Journey [8]	TC-8	Pending	Medium	FR-2

Requirement ID	Requirement Description	Source	Mapped Test Case ID	Status	Priority	Dependencies
FR-9	Patients must log health metrics like blood pressure or sugar levels, with system validation for accuracy.	Use Case 21 [7], Patient's Journey [8]	TC-9	Pending	Low	FR-1
FR-10	Admins must manage user roles and permissions to restrict data access based on roles.	Use Case 19 [7], Admin's Journey [8]	TC-10	Pending	High	None
FR-11	Admins must securely share patient records with external hospitals and maintain logs of these transactions.	Use Case 18 [7], Admin's Journey [8]	TC-11	Pending	Medium	FR-10

7 NFRs

7.1 Performance

Taking into consideration the system will support a government hospital, the system must be performant however the expectations are limited due to the lower funding in Egypt. This can be expressed as requiring the following from our system:

- Support up to 1500 concurrent users without noticeable delays.
- Critical operations (e.g., updating records) must respond within 100-300 milliseconds.
- Generate reports within 1-3 seconds.
- Handle up to **500 transactions per second** using workers and threads.
- Stress tested to ensure it can function without noticeable slowdown at 2x-3x the normal load for up to 15 minutes.
- **Enable real-time data synchronization (e.g., payments, bills, and insurance reimbursements) within 2 seconds of updates.**

7.2 Scalability

In case some clinics want to join the system and work under the hospital (most hospitals in Cairo have an external clinics section) we need our system to be able to quickly scale.

- Allow scaling to **support up to 7000** users as the system expands.
- Easy to integrate with: Enable data sharing between hospitals in standard formats (JSON, CSV, TSV).
- Easily accessible and secure API for the data transfer.
- **Support standardized data formats (e.g., HL7, FHIR, or APIs) for compatibility in the financial system integration.**

7.3 Availability

- **Ensure 99.8% uptime, with minimal maintenance downtime**, basic redundancy should be used such as **4 nodes at least**.
- Keep backup servers for critical functions, can be slower than main one.
- **Keep constant backups at multiple locations**

7.4 Security

- Encrypt all data using stored and transmitted.
- Use role-based access for restricting access for certain people in the hospital.

- Comply with Egyptian healthcare privacy laws.
- Should be tested against and resilient to OWASP top 10 vulnerabilities.

7.5 Usability

We expect our patient and staff to be familiar with smartphones to a certain extent, however we have no expectations from them in terms of computers as they are not as common in smaller cities.

- Hospital user interface should be intuitive, requiring no more than 10 hours of training.
- Learnability: Patients should be able to log health metrics (e.g., blood pressure) or view their data via a mobile app with no prior training, only a 10 minute guide.
- Prioritize accessibility for less tech-savvy users.
- Constantly take feedback via in app survey to ensure user satisfaction

7.6 Reliability

- Perform daily backups and store them offsite.
- Implement a disaster recovery plan to prevent data loss and make sure the backup server is constantly up to date.

7.7 Legislative Requirements

- System should not lose any financial data or have any rounding errors
- Ensure all data exchanges adhere to security and compliance regulations, such as HIPAA or GDPR.