

# **Electronic Healthcare Record System**

**June 2nd, 2025**

## **Table of Contents**

1.	Executive Summary	3
1.1	Project Overview	3
2.	Product/Service Description	3
2.1	Product Context	3
2.2	User Characteristics	3
2.3	Assumptions	3
2.4	Constraints and Dependencies	3
3.	Requirements	3
3.1	Functional Requirements	4
3.2	Non-Functional Requirements	5
3.2.1	Product Requirements	5
3.2.1.1	Usability Requirements	5
3.2.1.2	Performance Requirements	5
3.2.1.3	Availability	6
3.2.1.4	Security	6
3.2.2	Organizational Requirements	6
3.2.3	External Requirements	6
4.	User Scenarios/Use Cases	6
5.	Diagrams	7
6.	Design Patterns	7

# 1. Executive Summary

## 1.1 Project Overview

**Description:** This project involves the development of an **Electronic Healthcare System** designed to streamline patient care and medical record management. The system will support a wide range of users, including healthcare providers, patients, and emergency responders. It aims to improve efficiency, enhance patient care, and ensure compliance with healthcare regulations.

**Intended Audience:** The system is intended for use by:

- **Healthcare Providers:** Doctors, and laboratory staff.
- **Patients:** Individuals seeking medical care and managing their health records.
- **Emergency Responders:** Personnel needing quick access to patient information during emergencies.
- **IT Personnel:** Staff responsible for system maintenance and troubleshooting.

# 2. Product/Service Description

## 2.1 Product Context

**Description:** The **Electronic Healthcare System** is a standalone system designed to manage patient medical records and support healthcare delivery. While it is self-contained, it will interface with external systems such as pharmacies and laboratories to ensure seamless data exchange and coordination.

### How the Product Relates to Other Products

- **Standalone but Integrated:** The system manages core healthcare operations (e.g., appointment scheduling, medical record management, prescription management) internally but interfaces with external systems for extended functionality and provide end-to-end healthcare services.
- **Interdependence:** The system relies on external systems (e.g., pharmacies, laboratories) for specific functionalities, such as prescription management and lab test coordination.

### Interfaces with Related Systems

- **Internal Interfaces:**
  - **Hospital Databases:** The system integrates with internal hospital databases to manage patient records.
  - **IT Maintenance Tools:** The system supports tools for system monitoring, maintenance, and troubleshooting.

## **Electronic Healthcare System Requirements Specification**

- **Nutrition Services:** The system integrates with internal systems to support nutrition planning, allowing counselors and nutritionists to access patient records and provide personalized care.
- **External Interfaces:**
  - **Pharmacies:** Integration with pharmacies enables prescription management, including automated refill requests and drug interaction checks.
  - **Laboratory Systems:** The system interfaces with laboratories to manage test requests, process results, and notify relevant parties.
  - **Emergency Services:** Integration with emergency services provides real-time access to critical patient information during emergencies.
  - **Organ Donation Networks:** Integration with organ donation networks facilitates donor and recipient coordination.

## **Major Components and Interconnections**

Below is a high-level description of the system's major components and their interconnections:

1. **Core System:**
  - Manages **appointments, medical records**.
  - Acts as the central hub for all internal operations.
2. **Internal Systems:**
  - **Hospital Databases:** Store patient records, appointments.
  - **IT Maintenance Tools:** Support system monitoring, maintenance, and troubleshooting.
  - **Nutrition Services:** Provide tools for nutrition planning, integrated with patient records.
3. **External Systems:**
  - **Pharmacies:** Handle prescription management and drug dispensation.
  - **Laboratories:** Manage lab test requests and results.
  - **Emergency Services:** Access critical patient information during emergencies.
  - **Organ Donation Networks:** Coordinate organ donation processes.
4. **Mobile Applications:**
  - Provide patients and doctors with mobile access to the system for scheduling, record access, and notifications.
  - **Functionalities:**
    - **Patients:** Schedule appointments, access medical records, request prescription refills, and receive reminders.
    - **Healthcare Providers:** View patient records, update treatment plans, conduct virtual therapy sessions, and receive alerts for critical conditions.

## **2.2 User Characteristics**

### **User Profiles**

1. **Patients:**
  - **Role:** Individuals seeking medical care and managing their health records.

## ***Electronic Healthcare System Requirements Specification***

- **Experience:** Limited experience with healthcare systems; may have varying levels of comfort with technology.
- **Technical Expertise:** Low to moderate; need an intuitive and user-friendly interface.
- **Other Characteristics:** May require assistance with scheduling appointments, accessing medical records, and understanding medical terminology.

### **2. Doctors:**

- **Role:** Healthcare providers responsible for diagnosing and treating patients.
- **Experience:** Extensive experience in healthcare; familiar with medical systems and terminology.
- **Technical Expertise:** Moderate to high; comfortable using digital tools for managing patient records, prescribing medications, and viewing lab results.
- **Other Characteristics:** Need quick access to patient data and tools for efficient decision-making.

### **3. Pharmacy Staff:**

- **Role:** Personnel responsible for managing prescriptions and inventory.
- **Experience:** Moderate experience in pharmacy operations; familiar with prescription management.
- **Technical Expertise:** Moderate; comfortable using digital tools for managing prescriptions and inventory.
- **Other Characteristics:** Need tools for verifying prescriptions, checking drug interactions, and managing inventory.

### **4. Laboratory Staff:**

- **Role:** Personnel responsible for processing lab tests and managing test results.
- **Experience:** Moderate experience in laboratory operations; familiar with lab test processes.
- **Technical Expertise:** Moderate; comfortable using digital tools for managing test requests and results.
- **Other Characteristics:** Need tools for scheduling tests, processing results, and notifying relevant parties.

### **5. Emergency Services Personnel:**

- **Role:** First responders requiring quick access to critical patient information during emergencies.
- **Experience:** Moderate experience in emergency response; familiar with healthcare systems.
- **Technical Expertise:** Moderate; comfortable using digital tools for accessing patient data in real-time.
- **Other Characteristics:** Need prioritized access to patient records and tools for updating patient status during emergencies.

### **6. IT Support:**

**Role:** Team responsible for system maintenance and troubleshooting.

- **Experience:** High experience in IT operations; familiar with healthcare systems.
- **Technical Expertise:** High; comfortable using digital tools for system monitoring, maintenance, and troubleshooting.
- **Other Characteristics:** Need tools for monitoring system performance, applying updates, and resolving technical issues.

### **7. Nutritionists:**

## ***Electronic Healthcare System Requirements Specification***

- **Role:** Healthcare providers responsible for creating and monitoring personalized diet plans.
- **Experience:** Moderate experience in nutrition and diet planning; familiar with healthcare systems.
- **Technical Expertise:** Moderate; comfortable using digital tools for creating and updating diet plans.
- **Other Characteristics:** Need tools for tracking patient adherence to diet plans and monitoring progress.

### **8. Organ Donor Coordinators:**

- **Role:** Personnel responsible for managing organ donation processes and records.
- **Experience:** Moderate experience in organ donation coordination; familiar with healthcare systems.
- **Technical Expertise:** Moderate; comfortable using digital tools for managing donor and recipient records.
- **Other Characteristics:** Need tools for coordinating organ donation processes and tracking donor and recipient statuses.

## **2.3 Assumptions**

The following assumptions have been made during the development of the **Electronic Healthcare System**. If any of these assumptions change, the requirements may need to be revised accordingly.

### **1. Hardware and Software Availability:**

- The system will run on modern hardware and software platforms, including **Windows**, **macOS**, and **Linux**.
- Hospitals and clinics will have the necessary infrastructure (e.g., servers, network connectivity) to support the system.

### **2. User Expertise:**

- Users (e.g., patients, doctors) will have basic training to use the system effectively.
- IT support will have the technical expertise to maintain and troubleshoot the system.

### **3. Data Availability:**

- Patient records, lab results, and other critical data will be available in digital format and accessible through the system.
- External systems (e.g., pharmacies, laboratories) will provide real-time data exchange capabilities.

### **4. Regulatory Compliance:**

- The system will comply with healthcare regulations such as **GDPR** (General Data Protection Regulation).
- Hospitals and clinics will have the necessary processes in place to ensure compliance with these regulations.

### **5. Integration with External Systems:**

- External systems (e.g., pharmacies, insurance providers, laboratories) will provide APIs or other integration mechanisms for seamless data exchange.
- Emergency services and legal authorities will have secure access to the system as needed.

### **6. System Performance:**

## **Electronic Healthcare System Requirements Specification**

- The system will operate in environments with reliable internet connectivity and sufficient bandwidth to support real-time data access and updates.
- The system will handle peak usage without significant performance degradation.

### **7. Security:**

- Hospitals and clinics will implement necessary security measures (e.g., firewalls, encryption) to protect the system from unauthorized access.
- Users will follow security protocols (e.g., multi-factor authentication, password policies) to ensure data security.

### **8. Procurement and Supply Chain:**

- The procurement office will have access to reliable suppliers for medical equipment and supplies.
- Inventory levels will be tracked in real-time, and reorder alerts will be acted upon promptly.

### **9. User Adoption:**

- Patients, doctors, and other users will adopt the system and use it as their primary tool for managing healthcare operations.
- Training and support will be provided to ensure smooth adoption.

### **10. Mobile Applications:**

- Patients and doctors will have access to smartphones or tablets to use the mobile applications.
- The mobile applications will be compatible with **Android** and **iOS** platforms.

## **2.4 Constraints and Dependencies**

The following constraints and dependencies affect the requirements for the **Electronic Healthcare System**:

### **Constraints**

#### **1. Regulatory Compliance:**

- The system must comply with healthcare regulations such as **GDPR** (General Data Protection Regulation).
- This constraint affects the design of security features, data encryption, and audit trails.

#### **▪ Data Security and Privacy:**

- The system must ensure the security and privacy of patient data at all times.
- This constraint requires the implementation of end-to-end encryption, multi-factor authentication, and secure access controls.

#### **▪ System Performance:**

- The system must handle at least **100,000 concurrent users** without performance degradation.
- Response times for critical operations must not exceed **2 seconds**.
- This constraint affects the design of the system's architecture and database.

#### **▪ Integration with Legacy Systems:**

- The system must operate in parallel with existing legacy systems during the transition period.

## ***Electronic Healthcare System Requirements Specification***

- This constraint requires the system to support data migration and interoperability with older systems.
- **Budget and Resource Limitations:**
  - The development and implementation of the system are subject to budget and resource constraints.
  - This constraint may limit the scope of certain features or require prioritization of requirements.
- **User Training and Adoption:**
  - The system must be easy to use, with minimal training required for new users.
  - This constraint affects the design of the user interface and the development of training materials.
- **Mobile Application Compatibility:**
  - The system must support mobile applications for **Android** and **iOS** platforms.
  - This constraint affects the design and development of the mobile applications.

## **Dependencies**

- **Integration with External Systems:**
  - The system depends on integration with external systems such as **pharmacies**, **laboratories**, **emergency services**, and **organ donation networks**.
  - These integrations are necessary for functionalities like prescription management, lab test coordination, and emergency response.
- **Data Availability:**
  - The system depends on the availability of patient records, lab results, and other critical data in digital format.
  - If data is not available or accessible, the system's functionality may be impacted.
- **Third-Party APIs:**
  - The system depends on third-party APIs for integration with external systems (e.g., pharmacies, insurance providers, laboratories).
  - If these APIs are not available or do not function as expected, the system's functionality may be limited.
- **IT Infrastructure:**
  - The system depends on the availability of modern IT infrastructure (e.g., servers, network connectivity) in hospitals and clinics.
  - If the infrastructure is not available or insufficient, the system's performance may be affected.
- **Regulatory Approvals:**
  - The system depends on obtaining necessary regulatory approvals (e.g., HIPAA compliance certification).
  - If approvals are delayed or denied, the system's deployment may be impacted.
- **User Adoption:**
  - The system depends on the adoption and active use by patients, doctors, and other users.
  - If users do not adopt the system, its effectiveness may be limited.
- **Development of Specific Modules:**

- Certain modules (e.g., **Emergency Services Integration, Organ Donation Coordination**) must be completed before other modules can be built or deployed.
- This dependency affects the development timeline and prioritization of tasks.
- **Data Migration from Legacy Systems:**
  - The system depends on the successful migration of data from existing legacy systems.
  - If data migration is not completed on time or encounters issues, the system's deployment may be delayed.

## 3. Requirements

### 3.1 **User Requirements**

These requirements define the specific actions and functionalities that users need to perform within the system to fulfill their roles effectively:

#### **Patients**

- Patients should be able to schedule, modify, and cancel appointments through the system.
- Patients should have secure access to their medical records and test results.
- Patients should be able to request prescription refills through the system.
- Patients should receive automated reminders for upcoming appointments and medication.
- Patients should have access to telemedicine consultations through the platform.
- Patients should be able to provide feedback and rate their care experience.

#### **Doctors**

- Doctors should be able to securely log into the system.
- Doctors should have a dashboard that displays their appointments, messages, and tasks.
- Doctors should be able to view and update patient records easily.
- Doctors should be able to manage and reschedule appointments conveniently.
- Doctors should be able to prescribe medications electronically without complications.
- Doctors should be able to review and integrate lab results seamlessly.

#### **Laboratory Staff**

- Laboratory technicians should be able to securely log into the system.
- Laboratory technicians should have a dashboard to view and manage pending test requests.
- Laboratory technicians should be able to schedule patients for tests based on priority.
- Laboratory technicians should be able to conduct tests and record results in patient records.
- Laboratory technicians should be able to upload and attach scanned or machine-generated test reports.
- Laboratory technicians should be able to notify doctors when test results are available.
- Laboratory technicians should be able to notify patients when their results are ready.
- Laboratory technicians shall have all their actions logged for security and audit purposes.

#### **Pharmacy Staff**

## ***Electronic Healthcare System Requirements Specification***

- Pharmacy staff should be able to log in securely.
- Pharmacy staff should be able to process prescriptions efficiently and check for drug interactions.
- Pharmacy staff should be able to track and manage inventory in real-time.
- Pharmacy staff should be able to communicate with doctors for prescription clarifications.
- Pharmacy staff should be able to issue medications to patients in a timely manner.
- Pharmacy staff should be able to receive automatic alerts for low-stock or expired medications.
- Pharmacy staff should be able to generate reports on dispensed medications and inventory levels.
- Pharmacy staff should have access to patient medication history to avoid duplicate or conflicting prescriptions.
- Pharmacy staff should ensure compliance with pharmacy regulations and patient safety guidelines.
- Pharmacy staff should have access to an intuitive and efficient interface for quick access to necessary information.

### **Organ Donor Coordinator**

- Organ Donor Coordinator should be able to register new organ donors in the system.
- Organ Donor Coordinator should be able to match donors with recipients based on medical compatibility.
- Organ Donor Coordinator should be able to receive notifications when a matching donor-recipient pair is found.
- Organ Donor Coordinator should be able to track the real-time transport status of organs.
- Organ Donor Coordinator should be able to generate reports on organ donation statistics.

### **Emergency Service**

- Emergency staff should have instant access to patient emergency medical records, including allergies and ongoing treatments.
- Emergency responders should receive automatic navigation to the emergency location with traffic-aware routing.
- Emergency responders should be able to communicate directly with hospital staff while transporting a patient.
- Emergency staff should receive automatic alerts if a patient has infectious diseases or special care requirements.
- Emergency staff should be able to submit digital post-incident reports immediately after handling a case.

### **Nutritionist**

- Nutritionist should be able to log in securely.
- Nutritionist should be able to create and update personalized meal plans for patients.

### ***Electronic Healthcare System Requirements Specification***

- Nutritionist should be able to track patient progress over time (e.g., weight, BMI, dietary adherence).
- Nutritionist should be able to calculate nutritional needs based on patient data (e.g., age, gender, activity level).
- Nutritionist should have access to an intuitive interface for easy navigation.
- Nutritionist should be able to access patient data and generate reports quickly.
- Nutritionist should be able to share patient data and recommendations with other healthcare providers.
- Nutritionist should be able to receive referrals from physicians.
- Nutritionist should have access to educational resources for patients (e.g., articles, videos).
- Nutritionist should ensure that patient data is secure and accessible only to authorized personnel.

#### **IT Support**

- IT Support should be able to add new users (e.g., patients, doctors) to the system.
- IT Support should be able to update user information (e.g., roles, permissions).
- IT Support should be able to deactivate or delete users when necessary.
- IT Support should have the ability to manage user permissions and roles.
- IT Support should be able to monitor system performance in real time.
- IT Support should receive alerts for potential system failures or performance drops.
- IT Support should be able to apply patches and software updates remotely.

## **3.2 Functional Requirements**

#### **Laboratory Staff**

Req#	Requirement	Comments	Priority
FR_LAB_01	The system shall manage lab test requests and results, including scheduling and dissemination, and automatically notify doctors, patients, and laboratory staff when new results are posted.	Ensures efficient lab operations and reduces delays in medical decisions.	1
FR_LAB_02	The system shall support uploading, digital storage, and access to imaging files (X-rays, MRIs, CT scans).	Enables secure record management.	1

#### **Organ Donor Coordinator**

Req#	Requirement	Comments	Priority

## ***Electronic Healthcare System Requirements Specification***

FR_ORG_01	The system must allow coordinators to register organ donors by storing personal and medical details in the database. It should automatically generate donor-recipient matches using data from the database and notify coordinators for further review.	Storing donor information in a database helps keep records organized and easy to access. Using the database to find matches saves time by automating the process and improves the chances of finding compatible recipients quickly	1
FR_ORG_02	The system must allow coordinators to generate monthly and yearly reports using data from the database, covering donor registrations, successful transplants, waiting list status, and other key metrics. These reports should support medical research, performance analysis, and ensure compliance with healthcare regulations.	Generating regular monthly and yearly reports helps track how many donors have registered, how many transplants were successful, and the current waiting list status. Using the database ensures the information is accurate and up-to-date. These reports help hospitals make better decisions, improve performance, and meet legal requirements.	1

### **Doctors**

Req#	Requirement	Comments	Priority
FR_DOC_01	The system must allow doctors to create, edit, and update patient records, including medical history, diagnoses, treatments, and progress notes. These records should be securely stored and accessible in real-time to authorized healthcare providers.	Keeping patient records updated helps doctors make better decisions. When records are accurate and easy to access, doctors can provide faster and safer treatment.	1
FR_DOC_02	The system must enable doctors to electronically prescribe medications, check for potential drug interactions, and send prescriptions directly to connected pharmacies.	Electronic prescriptions help doctors provide accurate medication, prevent errors from handwriting or wrong doses, check for drug interactions, and speed up treatment by sending prescriptions directly to pharmacies.	1
FR_DOC_03	The system must allow doctors to view, manage, and update their schedules in	This requirement is crucial for the overall functionality of the system. Real-time updates ensure that any changes made to a doctor's schedule are	1

## ***Electronic Healthcare System Requirements Specification***

	real time, ensuring that patient appointments, surgeries, and other tasks are accurately recorded and synchronized to prevent scheduling conflicts.	immediately reflected across all relevant parts of the system, thus avoiding conflicts. The inclusion of patient appointments and surgeries is also vital, as these activities are time-sensitive and need to be handled with the utmost accuracy and synchronization.	
--	---	--	--

### **Emergency Service**

Req#	Requirement	Comments	Priority
FR_ES_01	The system shall provide emergency responders with real-time streaming of patient vitals from wearable health devices, integrate with GPS for automatic ETA updates, and pre-notify hospital staff about incoming cases.	Ensures timely response and hospital preparedness.	1
FR_ES_02	The system shall include a digital emergency checklist that guides responders based on patient conditions and enables automated report generation through voice commands and sensor data.	Reduces human error and administrative workload.	2
FR_ES_03	The system shall allow ambulances to reroute dynamically based on real-time traffic conditions and allow automatic hospital bed availability checks to ensure patients are taken to the nearest facility with available capacity.	Enhances efficiency in emergency response.	2

### **Nutritionist**

Req#	Requirement	Comment	Priority
FR_NUT_01	The system shall allow the nutritionist to manage (view, add, update, and delete) patient dietary records, assess intake, create personalized plans, and generate progress reports for data-driven decision-making.	Ensures accurate dietary records, continuous monitoring, and actionable insights.	1
FR_NUT_02	The system shall enable the nutritionist to share patient data with other healthcare professionals for coordinated care.	Supports collaboration for improved patient outcomes.	1

***Electronic Healthcare System Requirements Specification***

FR_NUT_03	The system shall allow the nutritionist to schedule, update, and conduct consultations, both in-person and virtual, for flexible patient care.	Provides accessibility and flexibility in care.	2
FR_NUT_04	The nutritionist shall be able to provide patients with educational materials, meal plans, send reminders, and track adherence to support patient engagement and compliance with dietary plans.	Encourages patient involvement and adherence to dietary plans.	3

**Pharmacy Staff**

Req#	Requirement	Comment	Priority
FR_PHA_01	The system shall manage prescriptions by allowing pharmacy staff to securely access, verify, process, and update patient prescriptions while checking for drug interactions, duplicate prescriptions, and ensuring secure communication with doctors.	Enhances patient safety, prevents medication errors, and ensures accurate prescription handling.	1
FR_PHA_02	The system shall track and manage pharmacy inventory in real-time, generating automatic reorder alerts for low stock and providing reports on dispensed medications, stock levels, and expiry dates.	Supports efficient inventory control, prevents shortages, and ensures regulatory compliance.	1
FR_PHA_03	The system shall let the pharmacy staff enroll patients in a loyalty program, track their purchase history, apply discounts based on accumulated loyalty points, and generate reports on customer purchasing trends and high-demand medications.	This feature supports customer retention by rewarding loyalty, improves inventory management through insights, and helps in optimizing pharmacy promotions and discounts.	3

## IT Support

Req#	Requirement	Comments	Priority
FR_IT_01	The system shall allow IT Support to add, update, deactivate, or delete user accounts and user information (e.g., roles, permissions).	Ensures new users can be registered and granted appropriate access and ensures user data remains accurate and up-to-date.	1
FR_IT_02	The system shall monitor system performance in real time and provide alerts for potential issues.	Ensures proactive system monitoring and stability.	1
FR_IT_03	The system shall allow IT Support to apply patches and software updates remotely.	Ensures timely maintenance and security compliance.	2

## Patients

Req#	Requirement	Comments	Priority
FR_PT_01	The system shall allow patients to schedule, modify, and cancel appointments.	Ensures flexibility and accessibility for patients.	2
FR_PT_02	The system shall provide patients with secure access to their medical records and test results.	Protects patient privacy while enabling self-care.	1
FR_PT_03	The system shall allow patients to request prescription refills.	Simplifies medication management for patients.	2
FR_PT_04	The system shall provide access to telemedicine consultations through the platform.	Enhances healthcare accessibility.	3
FR_PT_05	The system shall allow patients to provide feedback and rate their care experience.	Supports continuous improvement in healthcare services.	3
FR_PT_06	The system shall allow patients to quickly access emergency services, enabling one-click alerts to emergency responders.	Enhances rapid response capabilities during medical emergencies, ensuring timely medical intervention and improved patient safety.	1

### **3.3 Non-Functional Requirements**

### **3.3.1 Product Requirements**

#### **3.3.1.1 Usability Requirements**

- New users should require no more than 2 hours of training to use basic system functionalities effectively.
- The system should provide a comprehensive help section with at least 50 indexed help frames
- The system shall ensure that all user interfaces are intuitive and accessible to users with minimal training.
- The system shall support multilingual interfaces to accommodate diverse user bases.
- The system shall provide a mobile-friendly experience, with dedicated apps for Android and iOS.
- The system shall provide a high-contrast, simplified interface for emergency responders, optimized for use in high-stress environments.

#### **3.3.1.2 Performance Requirements**

- The system shall handle at least 100,000 concurrent users without performance degradation.
- Response times for all critical operations shall not exceed 2 seconds.
- The system shall support at least 50,000 simultaneous emergency requests to ensure scalability.

#### **3.3.1.3 Availability**

- The system shall provide data backup and disaster recovery solutions to ensure data integrity and availability.
- The system should average 10,000 hours between failures. on average, the system should operate flawlessly for about 10,000 hours before encountering a problem
- The system should have a downtime probability of less than 0.1%.
- The system should not have more than one failure per 14 months.
- The system must be available 99.9% of the time, as previously stated.
- The system shall perform daily backups of all critical data, with backups stored in a secure, offsite location. Data recovery shall be possible within 1 hour of an outage.

#### **3.2.1.4 Security**

- The system shall implement end-to-end encryption for all data transmissions.
- The system shall use multi-factor authentication for users.
- The system shall automatically log out users after 15 minutes of inactivity to prevent unauthorized access.

## ***Electronic Healthcare System Requirements Specification***

- The system shall ensure all data related to emergency health interventions is securely encrypted and stored for a minimum of 6 years to comply with medical privacy laws.

### **3.2.1.5 Speed**

- The system must process at least 1000 transactions per second during peak usage.
- The response time for user interactions should not exceed 1 second under normal conditions.
- Screen refreshes should occur in no more than 2 seconds.

### **3.2.1.6 Size**

- Minimum Starting Capacity of 1 TB (1000 GB). For initial operations and accommodate the storage of detailed medical images and other large data files.
- For hardware installations, the system should be efficient enough to run on servers without requiring more than 4 ROM chips.

### **3.2.1.7 Robustness**

- The system should be capable of restarting within 5 minutes after any failure.
- Less than 0.005% of system events should result in failures.
- The likelihood of data corruption during failures should be less than 0.001%

### **3.2.1.8 Portability**

- The system should be deployable on all major operating systems including Windows, macOS, and Linux.
- The system shall also be accessible via **mobile applications** on **iOS** and **Android** platforms, ensuring seamless functionality across both desktop and mobile devices.

### **3.2.1.9 Scalability**

- The system shall be scalable to accommodate an increasing number of users, data volume, and transaction intensity.
- The system shall be capable of integrating multiple hospitals and clinics without system downtime.

### **3.2.1.10 Maintainability**

- The system shall be easy to maintain and update, with capabilities for modular upgrades and patches.
- The system shall provide comprehensive logging and monitoring tools to facilitate troubleshooting and system analysis.

### **3.3.2 Organizational Requirements**

- The system shall integrate with the organization's existing Enterprise Resource Planning (ERP) system for procurement and financial tracking.
- The system shall include a training program for all users, with documentation available in both digital and printed formats.
- The system shall provide role-based training modules tailored to procurement staff, insurance companies, and laboratory technicians etc.
- The system shall generate monthly performance reports for review by the system's administrative team.
- The system shall enforce the organization's role-based access control (RBAC) policy, ensuring users only access data relevant to their roles.

### **3.3.3 External Requirements**

#### **3.3.3.1 Legislative Requirements:**

- The system shall comply with GDPR (General Data Protection Regulation) for data protection and privacy.

#### **3.3.3.2 Data Retention and Archiving:**

- The system shall comply with legal data retention policies, retaining patient records for a minimum of 6 years and financial records for 7 years.
- The system shall provide secure archiving solutions for long-term data storage, ensuring data integrity and accessibility.

#### **3.3.3.3 Industry Standards:**

- The system shall comply with ISO 27001 for information security management.

## **4. Use Cases**

### **4.1.1 Doctor Use Cases**

UC Name	UC_DOC_01 Manage Patient Records
Summary	The system allows doctors to create, edit, and update patient records, including medical history, diagnoses, treatments, and progress notes.

***Electronic Healthcare System Requirements Specification***

<b>Dependency</b>	<b>Manage Patient Records</b> depends on <b>UC_DOC_03: Manage Doctor's Timetable</b> for appointment-based updates.
<b>Actors</b>	Doctor
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-Doctor must be logged into the system.</li> <li>-If the patient exists, their record must be retrievable from the database.</li> <li>-If the patient is new, the system must allow new patient entry.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>Doctor logs into the system</b> using a secure username and password.</li> <li>2. <b>Doctor navigates to the 'Patient Records' module</b> from the system dashboard.</li> <li>3. <b>Doctor searches for the patient</b> by name, patient ID, or other identifier.</li> </ol> <p>If the patient exists:</p> <ol style="list-style-type: none"> <li>4. <b>The system retrieves the patient's record</b> and displays the following:</li> <ul style="list-style-type: none"> <li>• Personal details (name, date of birth, contact information)</li> <li>• Medical history</li> <li>• Diagnoses and treatments</li> <li>• Prescribed medications</li> <li>• Progress notes</li> </ul> <ol style="list-style-type: none"> <li>5. <b>Doctor reviews the record</b> and makes necessary updates.</li> <li>6. <b>Doctor modifies or adds new details</b>, such as: <ul style="list-style-type: none"> <li>• Updating medical history (e.g., new allergies or conditions).</li> <li>• Entering new diagnoses and treatments.</li> <li>• Adding new medications.</li> <li>• Writing progress notes.</li> </ul> </li> </ol> <ol style="list-style-type: none"> <li>7. <b>Doctor submits the changes.</b></li> </ol> </ol>

***Electronic Healthcare System Requirements Specification***

	<p>8. The system validates and securely stores the updated record.</p> <p>If the patient does not exist:</p> <p>9. The system informs the doctor that no matching record was found.</p> <p>10. Doctor selects 'Create New Patient'.</p> <p>11. Doctor enters the patient's personal and medical details.</p> <p>12. Doctor submits the new patient record.</p> <p>13. The system validates and securely stores the new record.</p>
<b>Alternative Sequence</b>	<ul style="list-style-type: none"> <li>- If a patient is not found, the system prompts the doctor to create a new record.</li> <li>- If data validation fails, the system requests corrections.</li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Secure access and encryption</li> <li>- Easy search and update functions</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The patient record is updated (if the patient existed).</li> <li>- A new patient record is created (if the patient did not exist).</li> <li>- The information is securely stored and available in real time.</li> </ul>

UC Name	UC_DOC_02 Electronic Prescription
<b>Summary</b>	The system enables doctors to create and manage electronic prescriptions, check for potential drug interactions, and send prescriptions securely to pharmacies.
<b>Dependency</b>	<b>Electronic Prescription</b> depends on <b>UC_DOC_01: Manage Patient Records</b> for medical details and

## ***Electronic Healthcare System Requirements Specification***

	interacts with the <b>Pharmacy Use Case</b> for prescription fulfillment.
<b>Actors</b>	Actors Primary Actor: Doctor Secondary Actors: Pharmacy, Patients
<b>Preconditions</b>	- Doctor must be logged in. - The patient must have an active medical record.
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. The doctor accesses the "Electronic Prescription" module.</li> <li>2. The system displays a list of patients or allows the doctor to search for a specific patient.</li> <li>3. The doctor selects a patient to open their medical profile.</li> <li>4. The system retrieves the patient's medical history, including existing medications and allergies.</li> <li>5. The doctor selects the option to create a new prescription.</li> <li>6. The doctor enters prescription details such as: <ul style="list-style-type: none"> <li>o Medication name</li> <li>o Dosage and frequency</li> <li>o Duration of use</li> </ul> </li> <li>7. The system automatically checks for potential drug interactions, allergies, and contraindications.</li> <li>8. If interactions are detected, the system alerts the doctor and provides suggested alternatives.</li> <li>9. The doctor reviews the prescription, makes adjustments if needed, and confirms it.</li> <li>10. The system securely sends the prescription to the selected pharmacy.</li> <li>11. The doctor receives confirmation that the prescription was successfully transmitted.</li> </ol>
<b>Alternative Sequence</b>	- If the doctor enters incomplete prescription details, the system prompts for missing information before proceeding.

## ***Electronic Healthcare System Requirements Specification***

	<ul style="list-style-type: none"> <li>-If the patient does not have a medical record, the doctor is prompted to create one before prescribing medication.</li>   <li>-If the doctor accidentally submits a duplicate prescription, the system warns them and asks for confirmation.</li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- High security and encryption</li> <li>- Real-time interaction checking</li> <li>- Integration with pharmacy systems</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The prescription is securely stored and transmitted to the pharmacy.</li> <li>• The pharmacy can proceed with fulfillment without delays.</li> <li>• The patient's medical record is updated with the new prescription.</li> </ul>

UC Name	UC_DOC_03 Manage Doctor's Timetable
<b>Summary</b>	This use case allows doctors to manage their schedules by viewing, updating, and organizing patient appointments, surgeries, and administrative tasks. The system ensures that patient bookings are automatically reflected in the timetable and prevents scheduling conflicts. Doctors can also manually add appointments if a patient is physically present or unable to book online.
<b>Dependency</b>	Manage Doctor's Timetable depends on the UC_PT_02 for scheduling updates
<b>Actors</b>	<ul style="list-style-type: none"> <li>• Primary Actor: Doctor</li> <li>• Secondary Actors: System, Patients</li> </ul>
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The doctor must be logged into the system with valid credentials.</li>   <li>-The system must have access to the doctor's existing schedule and patient appointments.</li>   <li>-The system must support patient-initiated bookings that automatically reflect in the doctor's schedule.</li>   <li>-The doctor must have the appropriate system permissions to modify the schedule.</li> </ul>

## **Electronic Healthcare System Requirements Specification**

<b>Main Sequence</b>	<ol style="list-style-type: none"><li>1. Doctor logs in using secure credentials.</li><li>2. Doctor navigates to the "Timetable" or "Schedule" section.</li><li>3. The system retrieves and displays the doctor's full schedule, including:<ul style="list-style-type: none"><li>• Confirmed Patient Appointments (automatically booked by patients).</li><li>• Pending Appointments (requests that require manual approval).</li><li>• Surgery Schedules.</li><li>• Administrative Meetings or Tasks.</li><li>• Breaks or Blocked Time Slots.</li></ul></li><li>4. Doctor views appointment details, including:<ul style="list-style-type: none"><li>• Patient's name, contact details, and medical history (if authorized).</li><li>• Reason for the visit.</li><li>• Appointment type (physical visit, teleconsultation, follow-up).</li><li>• Appointment status (confirmed, pending, rescheduled).</li></ul></li><li>5. Doctor updates the schedule as needed:<ul style="list-style-type: none"><li>• Manually Add Appointments (New Step!):<ul style="list-style-type: none"><li>○ If a patient is physically present or unable to book online, the doctor selects "Add Appointment".</li><li>• Doctor enters patient details (or selects an existing patient from the system).</li><li>• Doctor selects an available time slot.</li><li>• System validates availability and prevents double booking.</li><li>• Doctor confirms, and the system updates the timetable.</li><li>• Patient receives an appointment confirmation.</li><li>• Reschedule Appointments: Selects a new time slot → System checks for conflicts → Patient is notified.</li><li>• Block Time Slots: Doctor blocks time for personal use or administrative tasks → System prevents bookings in those slots.</li></ul></li><li>6. System automatically prevents double bookings and ensures time slot availability.</li></ul></li></ol>
----------------------	--

	<ol style="list-style-type: none"> <li>7. Doctor confirms and saves changes.</li> <li>8. System updates the timetable and sends notifications to affected patients (if changes were made).</li> </ol>
--	---

#### **4.1.2 Organ Donor Coordinator Use Cases**

UC Name	UC_ORG_01 Register Organ Donors
<b>Summary</b>	The system allows coordinators to register organ donors by storing personal and medical details. It automatically generates donor-recipient matches and notifies coordinators for further review.
<b>Dependency</b>	<b>Register Organ Donors</b> depends on the <b>Donor-Recipient Matching System</b> to generate matches
<b>Actors</b>	Organ Donor Coordinator
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The coordinator must be logged into the system with valid credentials.</li> <li>-The donor must provide required personal and medical details for registration.</li> <li>-The system must have access to the donor-recipient matching database to check for potential matches.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. Coordinator logs into the system using valid credentials.</li> <li>2. Coordinator navigates to the 'Register Donor' section from the dashboard.</li> <li>3. Coordinator selects 'Add New Donor'.</li> <li>4. Coordinator enters donor's personal details, including: <ul style="list-style-type: none"> <li>• Name</li> <li>• Date of birth</li> <li>• Contact information</li> <li>• Address</li> </ul> </li> <li>5. Coordinator enters donor's medical details, including: <ul style="list-style-type: none"> <li>• Blood type</li> <li>• Known allergies</li> <li>• Any medical conditions that may affect donation</li> </ul> </li> <li>6. The system validates the entered information to ensure completeness and correctness.</li> <li>7. The system stores the donor details in the database for future reference.</li> <li>8. The system attempts to generate a donor-recipient match based on:</li> </ol>

## ***Electronic Healthcare System Requirements Specification***

	<ul style="list-style-type: none"> <li>• Blood type compatibility</li> <li>• Urgency level of recipient</li> <li>• Geographic proximity</li> </ul> <p>9. The system notifies the coordinator with a match recommendation if a match is found.</p> <p>10. Coordinator reviews the match details and finalizes the registration.</p> <p>11. System confirms donor registration and logs the activity.</p>
<b>Alternative Sequence</b>	<ul style="list-style-type: none"> <li>- If validation fails, the system prompts for corrections.</li> <li>- If no match is found, the donor remains in the database for future matches.</li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Secure data storage</li> <li>- Fast retrieval of donor information</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- Donor details are saved in the system.</li> <li>- A match is generated (if available).</li> <li>- Notification sent to the coordinator.</li> </ul>

UC Name	UC_ORG_02 Generate Reports
<b>Summary</b>	The system allows coordinators to generate monthly and yearly reports covering donor registrations, transplants, waiting list status, and key metrics.
<b>Dependency</b>	<b>Generate Reports</b> depends on <b>UC_ORG_01: Register Organ Donors</b> for donor data
<b>Actors</b>	Organ Donor Coordinator
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- Coordinator must be logged into the system.</li> <li>- The database must contain donor and transplant data.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. Coordinator logs into the system using authorized credentials.</li> <li>2. Coordinator selects 'Generate Report' from the dashboard.</li> <li>3. Coordinator chooses the report type and time period (monthly or yearly).</li> <li>4. • The system retrieves relevant data from the database, including: <ul style="list-style-type: none"> <li>• Number of donor registrations</li> <li>• Number of successful transplants</li> <li>• Current waiting list statistics</li> </ul> </li> <li>5. The system processes the data and compiles it into a structured report.</li> </ol>

	<ol style="list-style-type: none"> <li>6. The system formats the report (e.g., PDF, Excel, or dashboard view).</li> <li>7. Coordinator reviews the generated report for accuracy.</li> <li>8. Coordinator exports or shares the report with relevant stakeholders.</li> </ol>
<b>Alternative Sequence</b>	<ul style="list-style-type: none"> <li>- If no relevant data is available, the system notifies the coordinator that report generation is not possible.</li> <li>- If an error occurs during report generation, the system logs the error and provides a retry option.</li> <li>- If the system takes too long to generate the report, the coordinator is given an option to receive it via email when ready</li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Fast data retrieval</li> <li>- Secure report storage</li> <li>- Compliance with healthcare regulations</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The report is generated and available for review.</li> <li>- The coordinator can download or share the report.</li> </ul>

#### **4.1.3 Nutricionist Use Cases**

UC Name	Dietary Record Management & Monitoring
<b>UC Code and Name</b>	UC_NUT_01: Dietary Record Management & Monitoring
<b>Summary</b>	The system shall allow the nutritionist to manage (view, add, update, and delete) patient dietary records, assess intake, create personalized plans, and generate progress reports for data-driven decision-making.
<b>Dependency</b>	This use case may depend on the "Patient Profile Management" use case for retrieving patient details.
<b>Actors</b>	<b>Primary Actor:</b> Nutritionist <b>Secondary Actors:</b> Patient (for providing dietary data)
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The nutritionist must be logged into the system.</li> <li>• The patient must have a registered profile in the system.</li> <li>• If the patient is new, the system must allow the entry of a new patient profile.</li> </ul>

## ***Electronic Healthcare System Requirements Specification***

<b>Description of the Main Sequence</b>	<ol style="list-style-type: none"> <li>1. The nutritionist logs into the system using a secure username and password.</li> <li>2. The nutritionist navigates to the “Dietary Record Management” module from the dashboard.</li> <li>3. The nutritionist searches for a patient using a unique identifier (e.g., name, patient ID, or contact details).</li> <li>10. <b>If the patient exists:</b> 4. The system retrieves and displays the patient's dietary records. 5. The nutritionist reviews and updates dietary records. 6. The nutritionist modifies or adds new details. 7. The nutritionist submits the changes. 8. The system validates and securely stores the updated record.</li> <li>11. <b>If the patient does not exist:</b> 9. The system informs the nutritionist. 12. The nutritionist creates a new patient profile. 13. The nutritionist creates a new patient record. 14. The system validates and securely stores the new record.</li> <li>4. The nutritionist generates a progress report.</li> <li>5. The system compiles and presents the report for review.</li> <li>6. The nutritionist finalizes and saves the report.</li> </ol>
<b>Description of the Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If the patient has no existing dietary records, the nutritionist creates a new dietary record.</li> <li>2. If invalid data is entered, the system prompts the nutritionist to correct it.</li> <li>3. If the patient profile is incomplete, the system notifies the nutritionist.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system should allow real-time access to dietary records.</li> <li>- Data security measures must be implemented to ensure patient confidentiality.</li> <li>- The system should support a user-friendly interface with role-based access control.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The dietary records are updated and saved in the system.</li> <li>- A personalized dietary plan is created.</li> <li>- A progress report is generated for monitoring dietary intake.</li> </ul>

UC Name	Healthcare Collaboration
UC Code and Name	UC_NUT_02: Healthcare Collaboration

## Electronic Healthcare System Requirements Specification

<b>Summary</b>	The system shall enable the nutritionist to share patient data with other healthcare professionals for coordinated care.
<b>Dependency</b>	This use case may depend on "Dietary Record Management & Monitoring" for accessing patient dietary records.
<b>Actors</b>	<p><b>Primary Actor:</b> Nutritionist</p> <p><b>Secondary Actors:</b> Other Healthcare Professionals (Doctors etc)</p>
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The nutritionist must be logged into the system.</li> <li>- The patient's dietary records must exist in the system.</li> </ul>
<b>Description of the Main Sequence</b>	<ol style="list-style-type: none"> <li>1. The nutritionist logs into the system.</li> <li>2. The nutritionist selects the "Share Patient Data" option.</li> <li>3. The system displays a list of available patient records.</li> <li>4. The nutritionist selects a patient record and chooses a healthcare professional to share it with.</li> <li>5. The system verifies access permissions and confirms data-sharing authorization.</li> <li>6. The system securely shares the data with the selected healthcare professional.</li> <li>7. The system notifies the recipient of the shared patient data.</li> </ol>
<b>Description of the Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If the healthcare professional is not authorized, the system denies access.</li> <li>2. If no patient records are available, the system notifies the nutritionist.</li> <li>3. If network issues occur, the system prompts the user to retry later.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must ensure secure data sharing using encryption and access control mechanisms.</li> <li>- Only authorized healthcare professionals should access patient data.</li> <li>- The system should log all data-sharing activities for auditing purposes.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- Patient data is securely shared with authorized healthcare professionals.</li> </ul>

## ***Electronic Healthcare System Requirements Specification***

	-The recipient healthcare professional receives a notification about the shared data.
--	---

<b>UC Name</b>	<b>Consultation Management</b>
<b>UC Code and Name</b>	UC_NUT_03: Consultation Management
<b>Summary</b>	The system shall allow the nutritionist to schedule, update, and conduct consultations, both in-person and virtual, for flexible patient care.
<b>Dependency</b>	This use case may depend on "Patient Profile Management" for retrieving patient details.
<b>Actors</b>	<p><b>Primary Actor:</b> Nutritionist</p> <p><b>Secondary Actors:</b> Patient</p>
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The nutritionist must be logged into the system.</li> <li>- The patient must have an active profile in the system.</li> </ul>
<b>Description of the Main Sequence</b>	<ol style="list-style-type: none"> <li>1. The nutritionist logs into the system.</li> <li>2. The nutritionist accesses the "Consultation Management" module.</li> <li>3. The system displays a calendar with available consultation slots.</li> <li>4. The nutritionist selects an available time slot and schedules a consultation.</li> <li>5. The system confirms the appointment and notifies the patient.</li> <li>6. On the scheduled date, the nutritionist conducts the consultation (in-person or virtual).</li> <li>7. The nutritionist records notes and recommendations after the consultation.</li> </ol>

## ***Electronic Healthcare System Requirements Specification***

<b>Description of the Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If the patient requests rescheduling, the system allows the nutritionist to modify the appointment.</li> <li>2. If the patient does not show up, the system logs a missed consultation.</li> <li>3. If network issues occur in virtual consultations, the system suggests rescheduling.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system should provide real-time scheduling updates to prevent conflicts.</li> <li>- Secure communication must be ensured for virtual consultations.</li> <li>- The system should provide automated reminders to both the nutritionist and patient.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The consultation is successfully completed.</li> <li>- The system updates the consultation records.</li> </ul>

<b>UC Name</b>	<b>Patient Engagement &amp; Compliance</b>
<b>UC Code and Name</b>	UC_NUT_04: Patient Engagement & Compliance
<b>Summary</b>	The nutritionist shall be able to provide patients with educational materials, meal plans, send reminders, and track adherence to support patient engagement and compliance with dietary plans.
<b>Dependency</b>	This use case may depend on "Dietary Record Management & Monitoring" for accessing patient dietary data.
<b>Actors</b>	<b>Primary Actor:</b> Nutritionist <b>Secondary Actor:</b> Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The nutritionist must be logged into the system.</li> <li>- The patient must have a registered profile in the system.</li> </ul>

## **Electronic Healthcare System Requirements Specification**

<b>Description of the Main Sequence</b>	<ol style="list-style-type: none"><li>1. The nutritionist logs into the system.</li><li>2. The nutritionist accesses the "Patient Engagement" module.</li><li>3. The system displays a list of patients and their dietary plans.</li><li>4. The nutritionist selects a patient and provides educational materials or updates meal plans.</li><li>5. The system sends reminders to the patient.</li><li>6. The patient logs dietary intake and progress updates.</li><li>7. The system tracks patient adherence and generates a compliance report.</li><li>8. The nutritionist reviews the report and adjusts the dietary plan if needed.</li></ol>
<b>Description of the Alternative Sequence</b>	<ol style="list-style-type: none"><li>1. If the patient does not engage with reminders, the system escalates the notification frequency.</li><li>2. If the patient misses logging their intake, the system prompts them to enter data.</li><li>3. If the nutritionist updates a dietary plan, the system automatically notifies the patient.</li></ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"><li>- The system must ensure secure messaging for patient communication.</li><li>- The system should provide automated reminders via multiple channels (email, SMS, app notifications).</li><li>- Data analytics should track patient adherence trends.</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- The patient receives educational materials and meal plans.</li><li>- The system tracks and records patient adherence.</li><li>- The nutritionist can assess patient compliance and modify plans accordingly.</li></ul>

### **4.1.4 Pharmacy Staff Use Cases**

## ***Electronic Healthcare System Requirements Specification***

<b>UC Name</b>	<b>Prescription Management</b>
<b>UC Code and Name</b>	UC_PH_01: Prescription Management
<b>Summary</b>	The system shall manage prescriptions by allowing pharmacy staff to securely access, verify, process, and update patient prescriptions while checking for drug interactions, duplicate prescriptions, and ensuring secure communication with doctors.
<b>Dependency</b>	This use case may depend on "Patient Profile Management" and "Doctor Prescription Entry."
<b>Actors</b>	<p><b>Primary Actor:</b> Pharmacy Staff</p> <p><b>Secondary Actors:</b> Doctors, Patients</p>
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The pharmacy staff must be logged into the system.</li> <li>- The patient must have a valid prescription from a doctor.</li> </ul>
<b>Description of the Main Sequence</b>	<ol style="list-style-type: none"> <li>1. The pharmacy staff logs into the system using secure credentials.</li> <li>2. The pharmacy staff navigates to the "Prescription Management" module.</li> <li>3. The system displays a list of active prescriptions.</li> <li>4. The pharmacy staff searches for a prescription using patient ID, prescription number, or doctor name.</li> <li>5. The system retrieves the prescription details, including: <ul style="list-style-type: none"> <li>o Patient information</li> <li>o Medication details (name, dosage, quantity)</li> <li>o Prescribing doctor details</li> <li>o Expiry date of prescription</li> </ul> </li> <li>6. The pharmacy staff verifies prescription validity and checks for: <ul style="list-style-type: none"> <li>o Drug interactions</li> <li>o Duplicate prescriptions</li> <li>o Dosage errors</li> </ul> </li> <li>7. If no issues are found, the pharmacy staff processes the prescription and dispenses the medication.</li> <li>8. The system updates the prescription status to "Fulfilled."</li> <li>9. The system notifies the patient that their medication is ready for collection.</li> </ol>

## ***Electronic Healthcare System Requirements Specification***

	<p>10. If verification from the doctor is required, the system notifies the doctor.</p>
<b>Description of the Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If a drug interaction is detected, the system alerts the pharmacy staff and suggests alternatives.</li> <li>2. If a duplicate prescription is found, the system prompts for manual review.</li> <li>3. If the prescription is incomplete or invalid, the system requests clarification from the doctor.</li> <li>4. If the patient does not collect the prescription within a defined period, the system sends a reminder.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must encrypt prescription data to ensure security.</li> <li>- Secure messaging should be available for doctor communication.</li> <li>- The system should maintain a complete audit log of prescription handling.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The prescription is processed, updated, and marked as fulfilled.</li> <li>- The patient is notified to collect their medication.</li> </ul>

<b>UC Code and Name</b>	UC_PH_02: Pharmacy Inventory Management
<b>Summary</b>	The system shall track and manage pharmacy inventory in real-time, generating automatic reorder alerts for low stock and providing reports on dispensed medications, stock levels, and expiry dates.
<b>Dependency</b>	This use case may depend on "Prescription Management" for tracking dispensed medications.

## ***Electronic Healthcare System Requirements Specification***

<b>Actors</b>	Primary Actor: Pharmacy Staff
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The pharmacy staff must be logged into the system.</li> <li>- The inventory database must be updated with current stock levels.</li> </ul>
<b>Description of the Main Sequence</b>	<ol style="list-style-type: none"> <li>1. The pharmacy staff logs into the system using secure credentials.</li> <li>2. The pharmacy staff navigates to the "Inventory Management" module.</li> <li>3. The system displays current stock levels, expiry dates, and medication categories.</li> <li>4. The pharmacy staff updates stock levels as medications are dispensed.</li> <li>5. The system tracks stock depletion and generates low-stock alerts.</li> <li>6. If stock is low, the system generates an automatic reorder request.</li> <li>7. The pharmacy staff reviews the reorder request and submits it for approval.</li> <li>8. The system updates inventory records to reflect pending orders.</li> <li>9. The system generates inventory reports for regulatory compliance.</li> </ol>
<b>Description of the Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If an expired medication is detected, the system alerts pharmacy staff for removal.</li> <li>2. If stock levels are incorrect, the system allows manual correction.</li> <li>3. If a medication is discontinued, the system suggests alternatives.</li> <li>4. If a reorder request is delayed, the system notifies pharmacy staff.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system should provide real-time inventory updates.</li> <li>- Secure access control must prevent unauthorized stock modifications.</li> <li>- Automated alerts should notify pharmacy staff of critical stock levels.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- Stock levels are updated in real-time.</li> <li>- Reorder requests are generated for low-stock medications.</li> </ul>

<b>Use Case ID</b>	UC_PH_03: Customer Loyalty & Discount Program
--------------------	---

## ***Electronic Healthcare System Requirements Specification***

<b>Summary</b>	The system allows pharmacy staff to enroll patients in a loyalty program, track purchase history, apply discounts for regular customers, and generate reports on high-demand medications and customer purchasing trends.
<b>Dependency</b>	May depend on "Pharmacy Inventory Management" for tracking dispensed medications.
<b>Actors</b>	<p><b>Primary Actor:</b> Pharmacy Staff</p> <p><b>Secondary Actor:</b> Patients</p>
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- Pharmacy staff must be logged into the system.</li> <li>- The patient must have a registered profile in the system.</li> </ul>
<b>Description of the Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>Pharmacy staff logs into the system</b> using secure credentials.</li> <li>2. <b>Pharmacy staff navigates to the "Customer Loyalty Program" module</b> from the system dashboard.</li> <li>3. <b>Pharmacy staff searches for an existing patient profile</b> using name, patient ID, or contact details. <ul style="list-style-type: none"> <li>o If the patient exists, their profile is displayed.</li> <li>o If the patient does not exist, staff must register the patient before proceeding.</li> </ul> </li> <li>4. <b>Pharmacy staff enrolls the patient in the loyalty program</b> by selecting the enrollment option and confirming participation.</li> <li>5. <b>The system generates a unique loyalty ID</b> and links it to the patient profile.</li> <li>6. <b>Patient makes a purchase</b> at the pharmacy.</li> <li>7. <b>The system tracks the patient's purchase history, including:</b> <ul style="list-style-type: none"> <li>o Medication names and quantities</li> <li>o Date of purchase</li> <li>o Total amount spent</li> <li>o Any applicable insurance or discounts used</li> </ul> </li> <li>8. <b>The system calculates loyalty points based on the purchase</b> and updates the patient's account.</li> </ol>

## **Electronic Healthcare System Requirements Specification**

	<ol style="list-style-type: none"><li>9. <b>The system applies a discount if the patient has accumulated sufficient loyalty points.</b><ul style="list-style-type: none"><li>o If the patient qualifies for a discount, it is applied automatically.</li><li>o If not, the points are saved for future use.</li></ul></li><li>10. <b>The system generates a transaction receipt</b>, displaying purchase details, loyalty points earned, and available discounts.</li><li>11. <b>The pharmacy staff provides the receipt to the patient</b> and confirms the successful transaction.</li><li>12. <b>The system updates inventory records</b> to reflect the dispensed medications.</li><li>13. <b>The system periodically generates reports on:</b><ul style="list-style-type: none"><li>• High-demand medications</li><li>• Customer purchasing trends</li><li>• Effectiveness of the loyalty program</li></ul></li><li>14. <b>Pharmacy staff reviews the reports</b> to make data-driven decisions regarding inventory and promotions.</li></ol>
<b>Description of the Alternative Sequence</b>	<ul style="list-style-type: none"><li>- If the patient is not registered, the system prompts pharmacy staff to create a new profile.</li><li>- If there's a discount calculation error, the system notifies pharmacy staff to apply the discount manually or adjust the loyalty balance.</li><li>- If the patient opts out, the system removes their loyalty status and stops tracking purchases for rewards.</li></ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"><li>- Secure handling of customer data and compliance with privacy regulations.</li><li>- Real-time discount calculations at checkout.</li><li>- Automated reports with analytics on customer trends and loyalty program effectiveness.</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- Patient is enrolled in the loyalty program (if new).</li><li>- Purchase is processed, loyalty points are updated, and discounts are applied.</li><li>- System tracks purchasing trends and generates reports for decision-making.</li></ul>

### **4.1.5 Laboratory Staff Use Cases**

***Electronic Healthcare System Requirements Specification***

<b>UC Name</b>	UC_LAB_01 – Manage Lab Test Requests and Results
<b>Summary</b>	Lab staff manage lab test requests, input results, and the system automatically notifies doctors, patients, and lab personnel when results are posted.
<b>Dependency</b>	Depends on successful login
<b>Actors</b>	Primary: Laboratory Staff Secondary: Doctor, Patient (as notification recipients)
<b>Preconditions</b>	Laboratory staff must be logged into the system.
<b>Description of the Main Sequence</b>	Step 1: Lab staff logs into the system. Step 2: Lab staff views pending test requests. Step 3: Conducts the lab test or receives results. Step 4: Uploads test results to the system. Step 5: System automatically notifies doctors and patients about result availability.
<b>Description of the Alternative Sequence</b>	Step 1: If test result is delayed, lab staff marks status as “pending”. Step 2: If upload fails, system prompts for re-upload.
<b>Non functional requirements</b>	Performance: Results upload should take under 3 seconds. Security: Lab staff have permission to upload doctors and patients can view, but not upload or edit. Notifications should be sent within 10 seconds of result submission.
<b>Postconditions</b>	Test result is stored in the system and notifications are sent to relevant users.

<b>UC Name</b>	UC_LAB_02 – Handle Imaging Files
<b>Summary</b>	Laboratory staff uploads, stores, and manages patient imaging files (X-rays, MRIs, CT scans) securely in the system for authorized access.
<b>Dependency</b>	Depends on successful login

## ***Electronic Healthcare System Requirements Specification***

<b>Actors</b>	Primary: Laboratory Staff Secondary: Doctor, Patient (as notification recipients)
<b>Preconditions</b>	Laboratory staff is authenticated and logged in. Imaging data is available in digital format.
<b>Description of the Main Sequence</b>	Step 1: Staff logs into the system. Step 2: Navigates to the “Upload Imaging” module. Step 3: Selects and uploads the imaging file (X-ray, MRI, etc). Step 4: Enters metadata (patient ID, scan type, date). Step 5: Confirms and submits. System stores file securely.
<b>Description of the Alternative Sequence</b>	Step 1: System detects invalid file type – prompts user to reupload. Step 2: Upload is interrupted – system retries or saves draft. Step 3: Missing metadata – system alerts user to complete required fields.
<b>Non functional requirements</b>	Imaging files must be encrypted and stored securely. System must ensure upload completes within 10 seconds for typical file sizes. Maximum file size per upload: 200MB
<b>Postconditions</b>	Imaging files are securely stored, linked to the correct patient, and accessible by authorized healthcare professionals.

### **4.1.6 Patient Use Cases**

<b>UC_PT_01</b>	<b>Patient Appointment Management</b>
<b>Summary</b>	The system shall allow patients to schedule, modify, and cancel appointments.
<b>Dependency</b>	None
<b>Actors</b>	Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-The patient must be logged into the system.</li> <li>-The patient must have an active user account.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>Patient logs into the system</b> using secure login credentials.</li> <li>2. <b>System verifies the login</b> and grants access to the patient dashboard.</li> </ol>

	<ol style="list-style-type: none"><li>3. <b>Patient navigates to the "Appointment Management" module from the dashboard.</b></li><li>4. <b>System displays options</b> to schedule, modify, or cancel appointments.</li><li>5. <b>Patient selects an action</b> based on their need.</li></ol> <p>If the patient chooses to schedule an appointment:</p> <ol style="list-style-type: none"><li>6. <b>System displays available time slots</b> for the selected provider and date range.</li><li>7. <b>Patient selects a preferred time slot</b> from the list.</li><li>8. <b>System checks time availability</b> in real time.</li><li>9. If the slot is available:<ol style="list-style-type: none"><li>a) Patient confirms the booking.</li><li>b) <b>System saves the appointment</b> and updates the scheduling database.</li><li>c) <b>System sends a confirmation notification</b> to the patient.</li></ol></li></ol> <p>If the patient chooses to modify an appointment:</p> <p>Patient views upcoming appointments.</p> <ol style="list-style-type: none"><li>6. Patient views upcoming appointments.</li><li>7. <b>Patient selects the appointment</b> to be modified.</li><li>8. System shows alternative available time slots.</li><li>9. Patient selects a new time slot.</li><li>10. System verifies the new slot's availability.</li><li>11. Patient confirms the change.</li><li>12. <b>System updates the appointment</b> and sends a reschedule confirmation notification.</li></ol> <p>If the patient chooses to cancel an appointment:</p> <ol style="list-style-type: none"><li>6. Patient views their appointment list.</li><li>7. Patient selects an appointment to cancel.</li><li>8. <b>System checks cancellation policy</b> (e.g., timing before appointment).</li><li>9. If cancellation is allowed:</li></ol>
--	--

***Electronic Healthcare System Requirements Specification***

	<ul style="list-style-type: none"> <li>a) Patient confirms the cancellation.</li> <li>b) <b>System deletes the appointment</b> and sends a cancellation confirmation notification.</li> </ul>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If the patient selects an unavailable time slot: <ul style="list-style-type: none"> <li>a) <b>System detects that the slot is already booked.</b></li> <li>b) System displays a "No time available" message.</li> <li>c) <b>Patient selects a new time slot</b>, and the main sequence resumes from step 8 (scheduling).</li> </ul> </li> <li>2. If the patient attempts to cancel too close to the appointment time: <ul style="list-style-type: none"> <li>a) <b>System checks the time remaining until the scheduled appointment.</b></li> <li>b) System determines that cancellation is restricted based on policy.</li> <li>c) <b>System displays a warning message</b> explaining the cancellation restriction.</li> <li>d) <b>Patient is prompted to acknowledge the policy</b> or return to the appointment view without canceling.</li> </ul> </li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must ensure real-time appointment availability updates.</li> <li>- Patients must receive automated notifications for appointment actions</li> <li>.- The interface must be user-friendly and accessible.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The appointment database is updated.</li> <li>- The patient receives a confirmation or reschedule notification.</li> </ul>

<b>UC_PT_02</b>	<b>Patient Medical Record Access</b>
<b>Summary</b>	The system shall provide patients with secure access to their medical records and test results.
<b>Dependency</b>	None

***Electronic Healthcare System Requirements Specification***

<b>Actors</b>	Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-The patient must be logged into the system.</li> <li>- The patient must provide valid authentication credentials.</li> <li>- The system must have existing records for the patient (if viewing).</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li><b>1 Patient logs into the system</b> using secure credentials..</li> <li><b>2 System verifies login credentials</b> and checks authentication validity.</li> <li><b>3</b> If credentials are valid:             <ol style="list-style-type: none"> <li><b>a) System grants access</b> and redirects the patient to the dashboard.</li> </ol> </li> <li><b>4 Patient navigates to the “Medical Records” module</b> from the dashboard.</li> <li><b>5 System initiates a request</b> to retrieve the patient's medical records and test results from the database.</li> <li><b>6 System checks if records exist</b> for the logged-in patient.</li> <li><b>7</b> If records are available:             <ol style="list-style-type: none"> <li><b>a) System displays the medical records and test results,</b> including:                     <ul style="list-style-type: none"> <li>• Personal information (e.g., name, birthdate)</li> <li>• Medical history</li> <li>• Lab test results</li> <li>• Diagnoses and treatments</li> </ul> </li> </ol> </li> <li><b>8 Patient views the records</b> through the interface.</li> <li><b>9 Patient may download or export records</b> if needed.</li> <li><b>10 System logs the access</b> and records the timestamp for auditing purposes.</li> </ol>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li><b>1</b> If login credentials are invalid:             <ol style="list-style-type: none"> <li><b>a) System shows an authentication error.</b></li> <li><b>b) Patient is prompted to re-enter correct login credentials.</b></li> </ol> </li> <li><b>2</b> If no records are available:             <ol style="list-style-type: none"> <li><b>a) System shows a "No records found" message.</b></li> <li><b>b) Patient is redirected back to the dashboard or remains in the module.</b></li> </ol> </li> </ol>

## ***Electronic Healthcare System Requirements Specification***

<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must ensure encrypted storage and secure access.</li> <li>- Patients must be notified of any changes in their records.</li> <li>- Audit logs should track record access history.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The patient successfully views or downloads their medical records.</li> <li>- The system logs the access history for security and compliance.</li> </ul>

<b>UC_PT_03</b>	<b>Prescription Refill Management</b>
<b>Summary</b>	The system shall allow patients to request prescription refills through the platform.
<b>Dependency</b>	None
<b>Actors</b>	Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-The patient must be logged into the system.</li> <li>- The patient must have an <b>active prescription</b> listed in their records.</li> <li>- The pharmacy system must be connected to receive and process requests.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1 <b>Patient logs into the system</b> using valid credentials.</li> <li>2 System verifies the login credentials.</li> <li>3 If the login is valid:             <ol style="list-style-type: none"> <li>a) <b>System grants access</b> and redirects the patient to the dashboard.</li> </ol> </li> <li>4. <b>Patient navigates to the “Prescription Refill” module</b> from the dashboard.</li> <li>5. <b>System displays a list of active prescriptions</b> eligible for refill.</li> <li>6. <b>Patient selects a prescription</b> from the list.</li> <li>7. Patient submits a refill request.</li> <li>8. <b>System checks the prescription validity</b> (e.g., not expired, refill limit not reached)</li> <li>9. If the prescription is valid:             <ol style="list-style-type: none"> <li>a) System forwards the refill request to the designated pharmacist.</li> <li>b) <b>System sends a confirmation message</b> to the patient.</li> </ol> </li> </ol>

***Electronic Healthcare System Requirements Specification***

	<p align="center"><b>10. Patient views the request status in the system interface.</b></p>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1 If login credentials are invalid:           <ol style="list-style-type: none"> <li>a) <b>System displays a login error</b> and prompts the patient to re-enter credentials.</li> </ol> </li> <li>2 If the selected prescription is invalid (e.g., expired or limit exceeded):           <ol style="list-style-type: none"> <li>a) <b>System shows an error message</b> indicating the issue.</li> </ol> </li> <li>3 If the pharmacy is unable to fulfill the request (e.g., out of stock):           <ol style="list-style-type: none"> <li>a) System notifies the patient of the issue.</li> </ol> </li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must ensure secure and accurate prescription management.</li> <li>- Patients must receive timely updates on refill status.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The pharmacy receives the refill request</li> <li>.- The patient is notified of the prescription status.</li> </ul>

<b>UC_PT_04</b>	<b>Telemedicine Consultation</b>
<b>Summary</b>	The system shall provide access to telemedicine consultations through the platform.
<b>Dependency</b>	None
<b>Actors</b>	<b>Primary Actor:</b> Patient <b>Secondary Actor:</b> Doctor
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-The patient must be logged into the system.</li> <li>-The doctor must have available consultation time in the schedule.</li> <li>-Both the patient and doctor must have access to a device with audio/video capability.</li> </ul>

## ***Electronic Healthcare System Requirements Specification***

<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>Patient logs into the system</b> using valid credentials.</li> <li>2. <b>System verifies login credentials</b> and redirects the patient to the dashboard.</li> <li>3. Patient navigates to the “Telemedicine” module.</li> <li>4. <b>System displays a list of available doctors</b> along with their open time slots.</li> <li>5. <b>Patient selects a doctor</b> and chooses an available time for the consultation.</li> <li>6. <b>System verifies the doctor's availability</b> and confirms the appointment.</li> <li>7. <b>System sends confirmation notifications</b> to both the patient and the doctor.</li> <li>8. <b>At the scheduled time</b>, the system initiates a secure video/audio session between the patient and the doctor.</li> <li>9. Patient and doctor conduct the consultation.</li> <li>10. <b>Upon completion, the session is closed</b> and the system logs the interaction.</li> <li>11. <b>Doctor may issue follow-up instructions or prescriptions</b> through the system.</li> <li>12. <b>Patient receives a summary or next steps</b> via notification or in-app message.</li> </ol>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1 If the selected doctor is unavailable (e.g., slot already booked):             <ol style="list-style-type: none"> <li>a) System displays a conflict message.</li> <li>b)</li> </ol> </li> <li>2 If the patient's connection is lost during consultation:             <ol style="list-style-type: none"> <li>a) <b>System attempts to reconnect automatically.</b></li> </ol> </li> <li>3 If the doctor ends the session early due to emergency or technical issue:             <ol style="list-style-type: none"> <li>a) <b>System notifies the patient</b> and reschedules if needed.</li> </ol> </li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must ensure encrypted and high-quality video communication.</li> <li>- Patients must receive timely reminders before consultations.</li> </ul>

## ***Electronic Healthcare System Requirements Specification***

<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The consultation is successfully completed and logged.</li> <li>- The patient receives follow-up instructions or prescriptions if needed.</li> </ul>
-----------------------	---

<b>UC_PT_05</b>	<b>Patient Feedback and Rating</b>
<b>Summary</b>	The system shall allow patients to provide feedback and rate their care experience.
<b>Dependency</b>	None
<b>Actors</b>	Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-The patient must have received a healthcare service (e.g., appointment, consultation, treatment).</li> <li>-The system must have a record of the completed service linked to the patient's account.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1 <b>Patient logs into the system</b> using secure credentials.</li> <li>2 <b>System verifies login credentials</b> and grants access to the dashboard.</li> <li>3 <b>Patient navigates to the “Feedback and Rating” module</b> from the dashboard.</li> <li>4 <b>System displays a feedback form.</b></li> <li>5 Patient enters a rating and provides optional written feedback.</li> <li>6 Patient submits the feedback form.</li> <li>7 <b>System validates the submission</b> (e.g., required fields, content length).</li> <li>8 <b>System stores the feedback</b> in the database.</li> </ol>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1 If the feedback submission fails, the system notifies the patient and allows retrying.</li> <li>2 If the feedback is inappropriate, the system flags it for review.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- - The system must ensure confidentiality of patient feedback.</li> <li>- Feedback must be structured and categorized for analysis.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The feedback is stored and available for service improvement.</li> </ul>

***Electronic Healthcare System Requirements Specification***

	- The healthcare provider receives anonymous reports on ratings.
--	--

UC_PT_06	Emergency Service Access
<b>Summary</b>	The system shall allow patients to quickly access emergency services, enabling one-click alerts to emergency responders.
<b>Dependency</b>	None
<b>Actors</b>	Patient
<b>Preconditions</b>	- The patient must be logged into the system.
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1 <b>Patient logs into the system</b> using valid credentials.</li> <li>2 <b>System verifies the login</b> and redirects the patient to the dashboard.</li> <li>3 <b>Patient accesses the “Emergency Services” module</b> from the dashboard.</li> <li>4 <b>System displays a one-click emergency alert button</b> with clear emergency labeling.</li> <li>5 Patient clicks the emergency alert button.</li> <li>6 <b>System captures the patient's location</b> (via GPS or profile information).</li> <li>7 <b>System compiles the patient's emergency profile</b>, including: <ul style="list-style-type: none"> <li>• Full name and age</li> <li>• Medical history or critical conditions (if available)</li> <li>• Emergency contact info</li> <li>• Location coordinates</li> </ul> </li> <li>8 <b>System immediately sends the alert</b> to pre-designated emergency responders or local authorities.</li> <li>9 <b>System displays confirmation</b> to the patient that the alert has been successfully sent.</li> <li>10 <b>Patient receives real-time updates</b> on responder status (e.g., “Help is on the way”).</li> </ol>

## ***Electronic Healthcare System Requirements Specification***

<b>Alternative Sequence</b>	<p>4. If location data is not accessible (e.g., GPS disabled):</p> <ul style="list-style-type: none"> <li>a) System prompts the patient to manually enter their location.</li> <li>b) <b>Patient enters location</b>, and the system continues the alert process.</li> </ul> <p>5. If the emergency request fails (e.g., no internet connection or server error):</p> <ul style="list-style-type: none"> <li>a) System retries sending the alert up to a set number of times.</li> <li>b) If all retries fail, <b>system notifies the patient</b> and attempts to alert a <b>backup contact</b> (e.g., via SMS or secondary channel)..</li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The emergency alert feature must operate with <b>low latency and high reliability</b>.</li> <li>- All transmissions must be <b>encrypted and secure</b>, especially patient identity and location.</li> <li>-The system must maintain <b>uptime guarantees</b> for this critical service.</li> <li>-Alerts should be <b>timestamped and logged</b> for auditing and response tracking.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- <b>Emergency responders receive a complete alert</b> containing the patient's data and location.</li> <li>-The patient is informed about response status.</li> <li>-<b>All activity is logged</b> for future reporting and accountability.</li> </ul>

### **4.1.7 IT Support Use Cases**

UC_IT_01	IT Support Manages User Accounts
<b>Summary</b>	The system shall allow IT support to manage user accounts, including creating, modifying, and deactivating user accounts.
<b>Dependency</b>	None
<b>Actors</b>	Primary Actor: IT Support
<b>Preconditions</b>	-IT support must be logged into the admin panel with appropriate permissions.

## ***Electronic Healthcare System Requirements Specification***

<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1 IT support accesses the "User Account Management" module.</li> <li>2 The system displays a list of existing user accounts.</li> <li>3 IT support selects the account to be managed.</li> <li>4 IT support can create, modify, or deactivate the account.</li> <li>5 The system confirms the action and updates the user account details.</li> <li>6 IT support receives confirmation of the action.</li> </ol>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If IT support attempts to create a duplicate account, the system notifies them and suggests merging or modifying the existing account.</li> <li>2. If IT support attempts to deactivate an account with active sessions, the system asks for confirmation before proceeding.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must ensure that <b>user account changes are logged</b> for auditing purposes.</li> <li>- Changes to user accounts should be <b>reflected in real time</b>.</li> <li>-The interface must be <b>secure and user-friendly</b> for IT support staff.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- <b>User account data is updated or deactivated</b> as per IT support's actions.</li> <li>-<b>IT support receives confirmation</b> of the completed action.</li> </ul>

<b>UC_IT_02</b>	<b>System Monitors Performance</b>
<b>Summary</b>	The system shall continuously monitor its performance, including CPU, memory, disk usage, and application status, and notify stakeholders of any performance issues.
<b>Dependency</b>	None
<b>Actors</b>	Primary Actor: System Secondary Actor: IT Support (when intervention is required)
<b>Preconditions</b>	-The system must be running, and performance monitoring tools must be active.

## ***Electronic Healthcare System Requirements Specification***

<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. The system starts monitoring performance metrics such as CPU, memory usage, disk space, and network activity.</li> <li>2. The system collects data at regular intervals and stores it in a log.</li> <li>3. If a performance threshold (e.g., high CPU usage) is exceeded, the system triggers an alert.</li> <li>4. IT support receives an alert with performance details.</li> <li>5. IT support investigates the issue and takes corrective action, if necessary (e.g., scaling resources, terminating processes).</li> <li>6. The system logs the resolution of the performance issue.</li> </ol>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If the system detects a minor issue (e.g., temporary spike in CPU usage), it can either attempt automatic recovery (e.g., load balancing) or escalate the alert to IT support.</li> <li>2. If IT support is unavailable, the system continues to monitor and escalate issues through automated channels (e.g., emails, SMS alerts).</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Performance data should be logged securely and with high accuracy.</li> <li>- Alerts must be timely and provide sufficient data for IT support to diagnose the issue.</li> <li>-The system should be designed for minimal impact on performance during monitoring.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- Performance data is recorded and accessible for review.</li> <li>-IT support resolves or acknowledges the performance issue.</li> <li>-The system continues its performance monitoring process.</li> </ul>

#### **4.1.8 Emergency Service Use Cases**

<b>UC Name:</b>	<b>UC_ES_01 Real-time Patient Vitals Streaming</b>
<b>Summary:</b>	The system provides emergency responders with real-time streaming of patient vitals from wearable health devices, integrates with GPS for automatic ETA updates, and pre-notifies hospital staff about incoming cases.
<b>Actors:</b>	<ul style="list-style-type: none"> <li>• <b>Primary Actor:</b> Emergency responders</li> <li>• <b>Secondary Actor:</b> Hospital staff</li> </ul>
<b>Preconditions:</b>	<p>-The emergency responder must be logged into the system with valid credentials.</p> <p>-Wearable health devices must be functional and paired with the system.</p> <p>-GPS tracking must be enabled.</p>
<b>Description of the Main Sequence:</b>	<ol style="list-style-type: none"> <li>1. The emergency responder logs into the system using secure credentials.</li> <li>2. The system verifies the login details and grants access to the dashboard.</li> <li>3. The responder selects the 'Real-time Patient Vitals' module from the dashboard.</li> <li>4. The responder searches for the patient by entering their unique ID or scanning the wearable device pairing code.</li> <li>5. The system retrieves the patient's details and confirms a connection with the wearable device.</li> <li>6. The system starts receiving real-time vital signs from the patient's device (e.g., heart rate, blood pressure, oxygen levels).</li> <li>7. The responder reviews the incoming vitals on their interface.</li> <li>8. The system streams the vitals in real-time to the emergency response team's dashboard.</li> </ol>

***Electronic Healthcare System Requirements Specification***

	<p>9. The system calculates the estimated time of arrival (ETA) using GPS data and updates it dynamically.</p> <p>10. The hospital staff receives a pre-notification with the patient's details, current vitals, and ETA.</p> <p>11. The emergency responder continuously monitors the vitals and updates necessary records.</p> <p>12. The system securely stores the patient's vitals and logs for future reference.</p>
<b>Description of the Alternative Sequence:</b>	<p>1. If the wearable device fails to transmit data, responders are alerted to check vitals and input them manually.</p> <p>2. If GPS tracking is lost, responders manually update ETA and notify hospital staff.</p>
<b>Non-functional Requirements:</b>	<ul style="list-style-type: none"> <li>- Secure data transmission.</li> <li>- Must support low-latency data transmission.</li> </ul>
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>- Hospital staff receive the pre-notification with updated patient status.</li> <li>-Patient vitals are securely stored for reference.</li> </ul>

<b>UC Name:</b>	<b>UC_ES_02 Digital Emergency Checklist</b>
<b>Summary:</b>	The system provides a digital checklist that guides responders based on patient conditions and enables automated report generation through voice commands and sensor data.
<b>Actors:</b>	<ul style="list-style-type: none"> <li>• <b>Primary Actor: Emergency responders</b></li> </ul>
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>-The emergency responder must be logged into the system with valid credentials.</li> <li>-The digital checklist system must be functional.</li> <li>-The system should have voice recognition enabled.</li> </ul>
<b>Description of the Main Sequence:</b>	<ol style="list-style-type: none"> <li>1. The emergency responder logs into the system using secure credentials.</li> <li>2. The system verifies the credentials and grants access.</li> <li>3. The responder navigates to the 'Digital Emergency Checklist' module.</li> </ol>

***Electronic Healthcare System Requirements Specification***

	<ol style="list-style-type: none"> <li>4. The responder inputs or selects the patient's condition or symptoms.</li> <li>5. The system customizes the checklist based on the patient's symptoms and displays step-by-step guidance.</li> <li>6. The responder follows the guided steps, checking off completed tasks.</li> <li>7. If applicable, the system collects real-time sensor data to update the checklist dynamically.</li> <li>8. The responder provides voice inputs for additional details (e.g., consciousness level, pain response).</li>   <li>9. The system records the voice input and automatically generates a report.</li> <li>10. The system securely saves the report and sends it to relevant hospital staff.</li> </ol>
<b>Description of the Alternative Sequence:</b>	<ol style="list-style-type: none"> <li>1. If voice recognition fails, responders manually input checklist data.</li> <li>2. If automated report generation is not possible, responders complete reports manually.</li> </ol>
<b>Non-functional Requirements:</b>	<ul style="list-style-type: none"> <li>- The checklist must be accessible offline.</li> <li>- Support voice recognition.</li> <li>- Integrate with hospital records.</li> </ul>
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>-A completed emergency report is generated and available for medical staff review.</li> </ul>

<b>UC Name:</b>	<b>UC_ES_03 Dynamic Ambulance Rerouting</b>
<b>Summary:</b>	The system allows ambulances to reroute dynamically based on real-time traffic conditions and enables automatic hospital bed availability checks.
<b>Actors:</b>	<ul style="list-style-type: none"> <li>• <b>Primary Actor:</b> Emergency responder</li> </ul>
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>- The emergency responder must be logged into the system with valid credentials.</li> </ul>

***Electronic Healthcare System Requirements Specification***

	<ul style="list-style-type: none"> <li>-GPS tracking must be active.</li> <li>- Hospital databases must be accessible.</li> </ul>
<b>Description of the Main Sequence:</b>	<ol style="list-style-type: none"> <li>1. The emergency responder logs into the system using secure credentials.</li> <li>2. The system verifies the login and grants access.</li> <li>3. The responder accesses the 'Dynamic Ambulance Rerouting' module.</li> <li>4. The system continuously monitors real-time traffic conditions using GPS data.</li> <li>5. The system suggests the fastest available route based on traffic updates.</li> <li>6. The responder confirms the suggested route or selects an alternative if necessary.</li> <li>7. The system checks hospital bed availability in the nearest facilities.</li> <li>8. If the initially intended hospital has available capacity, the system confirms the route.</li> <li>9. If the preferred hospital is full, the system automatically suggests the next best option.</li> <li>10. The system reroutes the ambulance accordingly and updates ETA.</li> <li>11. The hospital staff is notified about the incoming patient and estimated arrival time.</li> <li>12. The responder follows the optimized route while monitoring patient condition updates.</li> </ol>
<b>Description of the Alternative Sequence:</b>	<ol style="list-style-type: none"> <li>1. If GPS tracking fails, responders manually select a route.</li> <li>2. If no nearby hospitals have available beds, the system suggests the next best option.</li> </ol>
<b>Non-functional Requirements:</b>	<ul style="list-style-type: none"> <li>- Must support real-time updates.</li> <li>- Ensure accurate data processing.</li> </ul>

	<ul style="list-style-type: none"> <li>- Provide high system reliability.</li> </ul>
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>- The patient is transported to the most suitable hospital efficiently.</li> <li>- The ambulance reaches the destination using an optimized route.</li> </ul>

## Extension Use Cases:

<b>UC_PT_07</b>	Manage Appointments
<b>Summary</b>	The system shall allow patients to schedule new appointments by selecting a date, time, and healthcare provider based on availability.
<b>Dependency</b>	Extends <b>UC_PT_01: Manage Appointments</b>
<b>Actors</b>	Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-The patient must be logged into the system.</li> <li>-The patient must have an active user account.</li> <li>-Available appointment slots must be defined in the system.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>Patient logs into the system</b> using secure login credentials.</li> <li>2. <b>System verifies credentials</b> and grants access to the patient dashboard.</li> <li>3. <b>Patient navigates to the "Appointment Management" module.</b></li> <li>4. <b>Patient selects "Schedule New Appointment."</b></li> <li>5. <b>System displays available time slots</b> filtered by doctor, date, and location.</li> <li>6. <b>Patient selects a time slot</b> based on preference.</li> <li>7. <b>System checks the real-time availability</b> of the selected slot.</li> <li>8. <b>If available, patient confirms the booking.</b></li> </ol>

***Electronic Healthcare System Requirements Specification***

	<p>9. <b>System reserves the appointment</b>, saves it in the schedule, and updates the database.</p> <p>10 <b>System sends a confirmation notification</b> to the patient.</p>
<b>Alternative Sequence</b>	<p><b>1.If the selected slot is no longer available:</b></p> <ul style="list-style-type: none"> <li>a. System displays a "Slot Unavailable" message.</li> <li>b. System suggests alternative time slots.</li> <li>c. Patient selects a new time slot and proceeds with the scheduling process.</li> </ul> <p><b>2. If internet connection fails during submission:</b></p> <ul style="list-style-type: none"> <li>a. System prompts the patient to retry.</li> <li>b. Upon reconnection, system resumes the process from the last action.</li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Appointment availability data must be updated in <b>real time</b>.</li> <li>- The interface must be <b>user-friendly and responsive</b> on both desktop and mobile.</li> <li>- The scheduling action must be <b>completed within 2 seconds</b> under normal conditions.</li> <li>- A <b>confirmation message</b> must be sent immediately after scheduling.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The appointment is successfully scheduled and logged in the system.</li> <li>-The patient receives a confirmation via email, SMS, or in-app notification.</li> <li>- The healthcare provider receives anonymous reports on ratings.</li> </ul>

UC_PT_08	Modify Appointment
<b>Summary</b>	The system shall allow patients to modify existing appointments by selecting a new date and time based on provider availability.

***Electronic Healthcare System Requirements Specification***

<b>Dependency</b>	Extends <b>UC_PT_01: Manage Appointments</b>
<b>Actors</b>	Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-The patient must be logged into the system.</li> <li>-An active appointment must exist in the patient's schedule.</li> <li>-Available alternative time slots must be retrievable for modification.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li><b>1. Patient logs into the system using valid credentials.</b></li> <li><b>2. System verifies credentials</b> and grants access to the patient dashboard.</li> <li><b>3. Patient navigates to the "Appointment Management" module.</b></li> <li>4. Patient selects the appointment to be modified.</li> <li><b>5. System displays the details of the selected appointment.</b></li> <li><b>6. System shows alternative available time slots for the same doctor or service.</b></li> <li><b>7. Patient selects a new preferred time slot.</b></li> <li><b>8. System verifies slot availability in real time.</b></li> <li><b>9. Patient confirms the rescheduling.</b></li> <li><b>10 System updates the appointment record, replaces the old time slot, and notifies the patient.</b></li> </ol>
<b>Alternative Sequence</b>	<ol style="list-style-type: none"> <li>1. If no alternative slots are available: <ul style="list-style-type: none"> <li>a. <b>System displays a message</b> stating that no new slots are currently available.</li> <li>b. <b>Patient is prompted to try again later</b> or cancel the appointment instead.</li> </ul> </li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must <b>prevent double bookings</b> during slot selection.</li> <li>- Rescheduling must <b>complete within 3 seconds</b> of confirmation.</li> </ul>

***Electronic Healthcare System Requirements Specification***

	<ul style="list-style-type: none"> <li>- Notifications must be sent <b>instantly upon update</b>.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The appointment is updated in the system and reflected on the patient's dashboard</li> <li>-The patient receives a reschedule confirmation.</li> <li>- The previous slot is released and made available for other users.</li> </ul>

<b>UC_PT_09</b>	Cancel Appointment
<b>Summary</b>	The system shall allow patients to cancel their scheduled appointments, subject to cancellation policies.
<b>Dependency</b>	Extends <b>UC_PT_01: Manage Appointments</b>
<b>Actors</b>	Patient
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The patient must be logged into the system.</li> <li>• The patient must have at least one active appointment scheduled.</li> <li>• The system must check cancellation rules based on time of cancellation.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li><b>1. Patient logs into the system</b> using secure credentials.</li> <li><b>2. System verifies credentials</b> and grants dashboard access.</li> <li><b>3. Patient opens the “Appointment Management” module.</b></li> <li><b>4. Patient views a list of their upcoming appointments.</b></li> <li><b>5. Patient selects an appointment to cancel.</b></li> <li><b>6. System checks the cancellation policy</b> (e.g., whether cancellation is permitted this close to the appointment).</li> <li><b>7. If allowed</b>, patient confirms cancellation.</li> <li><b>8. System deletes the appointment record</b> from the schedule.</li> <li><b>9. System notifies the patient</b> and logs the cancellation.</li> </ol>

***Electronic Healthcare System Requirements Specification***

<b>Alternative Sequence</b>	<b>If there is a system error or failed submission:</b> <ul style="list-style-type: none"> <li>a. <b>System prompts the patient to retry.</b></li> <li>b. <b>If retry fails</b>, patient is advised to contact support.</li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- The system must <b>enforce cancellation policies automatically</b>.</li> <li>- The cancellation process must complete within <b>2 seconds</b>.</li> <li>- Notifications must be sent <b>immediately</b> via program or SMS.</li> <li>- All cancellations must be <b>timestamped and logged</b> for auditing.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The appointment is removed from the patient's upcoming schedule.</li> <li>- The patient receives a cancellation confirmation.</li> <li>- The slot becomes available for other patients.</li> <li>- The system logs the cancellation for historical tracking.</li> </ul>

UC_PH_04	Check Drug Interactions
<b>Summary</b>	The system shall allow pharmacy staff to automatically check for drug interactions when verifying or processing prescriptions.
<b>Dependency</b>	Extends Manage Prescription
<b>Actors</b>	<b>Primary Actor:</b> Pharmacy Staff
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• Pharmacy staff must be logged into the system.</li> <li>• A prescription must be selected and visible to the staff.</li> <li>• The system must have access to the patient's medication history.</li> </ul>

***Electronic Healthcare System Requirements Specification***

<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. Pharmacy staff selects a prescription to verify.</li> <li>2. System retrieves the patient's active medication history.</li> <li>3. System automatically checks for drug interactions, including:           <ul style="list-style-type: none"> <li>• Conflicting medications</li> <li>• Drug-allergy interactions</li> <li>• Known contraindications</li> </ul> </li> <li>4. System presents the interaction results to pharmacy staff.</li> <li>5. Pharmacy staff reviews the alert and decides whether to proceed.</li> <li>6. If necessary, staff contacts the prescribing doctor for clarification or alternative recommendations.</li> <li>7. System logs the interaction check and outcome.</li> </ol>
<b>Alternative Sequence</b>	<p><b>If no interaction is found:</b></p> <ol style="list-style-type: none"> <li>a. System confirms prescription is safe to process.</li> </ol> <p><b>If drug interaction is detected:</b></p> <ol style="list-style-type: none"> <li>a. System displays a detailed alert with explanation and risk level.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Drug interaction database must be <b>updated regularly</b>.</li> <li>- Interaction check must complete within <b>1–2 seconds</b>.</li> <li>- Alerts must be <b>clear and color-coded</b> by severity.</li> <li>- System must <b>log all interaction checks</b> for legal/audit purposes.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The system logs the interaction check and outcome.</li> <li>• The prescription is marked as cleared, modified, or flagged for review.</li> <li>• The pharmacist is informed of any risks and actions taken.</li> </ul>

UC_PH_05	Check Dosage Errors
<b>Summary</b>	The system shall automatically check prescription dosages against standard medical guidelines to identify potential dosage errors during prescription verification.
<b>Dependency</b>	Extends Manage Prescription

## ***Electronic Healthcare System Requirements Specification***

<b>Actors</b>	<b>Primary Actor: Pharmacy Staff</b>
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The pharmacy staff must be logged into the system.</li> <li>• A prescription must be selected and ready for verification.</li> <li>• The system must have access to clinical dosage guidelines and patient information (e.g., age, weight).</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. Pharmacy staff selects a prescription to verify.</li> <li>2. System retrieves the prescribed dosage and relevant patient data (e.g., age, weight, diagnosis).</li> <li>3. System compares the dosage against safe dosage ranges from medical databases.</li> <li>4. If the dosage is within the acceptable range, the system confirms safety.</li> <li>5. If the dosage appears abnormal, the system highlights it as a warning.</li> <li>6. Pharmacy staff reviews the warning and determines whether to proceed.</li> <li>7. If needed, the pharmacist contacts the prescribing doctor for clarification.</li> <li>8. System logs the check outcome for compliance tracking.</li> </ol>
<b>Alternative Sequence</b>	<p><b>If dosage exceeds recommended limits:</b></p> <ol style="list-style-type: none"> <li>a. System displays a red alert with severity level.</li> <li>b. Pharmacy staff is required to review and confirm override or contact the doctor.</li> </ol> <p><b>If the dosage is below therapeutic range:</b></p> <ol style="list-style-type: none"> <li>a. System suggests a reevaluation of the dose.</li> <li>b. Pharmacy staff is prompted to verify with the doctor if the dosage is intentional.</li> </ol> <p><b>If patient information is missing or incomplete:</b></p> <ol style="list-style-type: none"> <li>a. System prompts pharmacy staff to complete or request additional details before continuing.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- System must access <b>up-to-date dosage guidelines</b> from verified clinical sources.</li> <li>- The check must complete in <b>under 2 seconds</b>.</li> <li>- Alerts must be <b>clear, color-coded</b>, and accompanied by explanations.</li> <li>- The interaction must be <b>securely logged and time-stamped</b>.</li> </ul>

<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The system logs the dosage verification outcome.</li> <li>• The pharmacist is informed of any concerns and actions taken.</li> <li>• If confirmed safe, the prescription proceeds to processing.</li> <li>• If incorrect, the prescription is corrected before fulfillment.</li> </ul>
-----------------------	---

Nutritionist

<b>UC_NUT_05</b>	Provide Educational Materials and Track Adherence
<b>Summary</b>	The system shall allow the nutritionist to provide patients with educational materials and monitor adherence to prescribed dietary plans through reminders and tracking tools.
<b>Dependency</b>	Extends Patient Engagement & Compliance
<b>Actors</b>	<b>Primary Actor: Nutritionist</b> <b>Secondary Actor: Patient</b>
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The nutritionist must be logged into the system.</li> <li>• The patient must have a registered profile and an active dietary plan.</li> <li>• The system must have communication and tracking tools enabled.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>Nutritionist logs into the system</b> and accesses the “Patient Engagement” module.</li> <li>2. <b>System displays the patient list and their dietary adherence status.</b></li> <li>3. <b>Nutritionist selects a patient</b> and views their current dietary plan.</li> <li>4. <b>Nutritionist attaches educational materials</b>, such as articles, videos, or PDFs.</li> <li>5. <b>System sends the materials to the patient</b> via the patient portal or app.</li> <li>6. <b>System enables daily/weekly tracking</b>, prompting the patient to log meals, progress, or symptoms.</li> <li>7. <b>System compiles adherence data</b> and updates it in the patient’s progress record.</li> <li>8. <b>Nutritionist reviews adherence logs</b> and adjusts the dietary plan if necessary.</li> </ol>
<b>Alternative Sequence</b>	<b>If the patient does not open or engage with materials:</b> <ol style="list-style-type: none"> <li>a. <b>System increases reminder frequency.</b></li> <li>b. <b>System notifies the nutritionist of non-engagement.</b></li> </ol>

## ***Electronic Healthcare System Requirements Specification***

	<p><b>If patient data logging is incomplete or inconsistent:</b></p> <ul style="list-style-type: none"> <li>a. System sends prompts to the patient.</li> <li>b. Nutritionist may reach out for clarification.</li> </ul> <p><b>If educational content fails to upload or send:</b></p> <ul style="list-style-type: none"> <li>a. <b>System provides retry options.</b></li> </ul>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Educational materials must be stored and delivered <b>securely</b>.</li> <li>- System must support <b>reminder automation and multilingual content</b>.</li> <li>- All patient interactions and logs must be <b>tracked with time stamps</b>.</li> <li>- User interface must be <b>accessible on mobile and desktop</b>.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Patient receives educational content tailored to their dietary plan.</li> <li>• Patient adherence data is collected and logged in the system.</li> <li>• The nutritionist can evaluate and adjust treatment based on tracked behavior.</li> </ul>

Emergenza:

<b>UC_EME_04</b>	<b>Check Hospital Bed Availability</b>
<b>Summary</b>	The system shall allow emergency responders to check real-time hospital bed availability during ambulance routing to ensure the patient is taken to a facility with available capacity.
<b>Dependency</b>	Extends <b>UC_ES_03: Dynamic Ambulance Rerouting</b>
<b>Actors</b>	<b>Primary Actor:</b> Emergency Responder <b>Seconday Actor :</b> Hospital
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The emergency responder must be logged into the system.</li> <li>• GPS and routing modules must be active.</li> <li>• The system must have access to real-time hospital bed availability data.</li> </ul>

## Electronic Healthcare System Requirements Specification

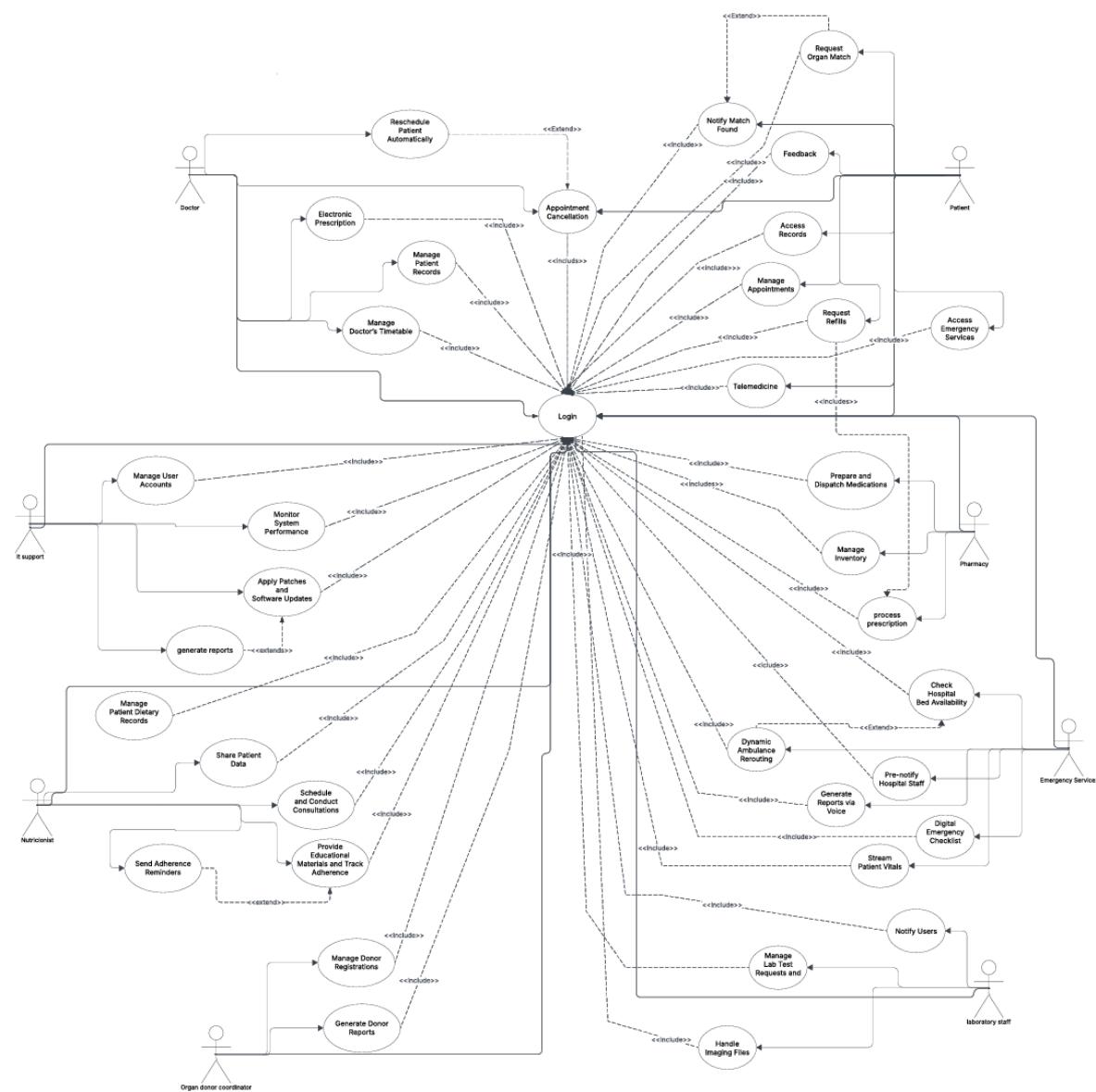
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>Emergency responder initiates ambulance routing</b> using the “Dynamic Ambulance Rerouting” module.</li> <li>2. <b>System identifies nearby hospitals</b> based on GPS location.</li> <li>3. <b>System queries bed availability</b> in real time from hospital databases.</li> <li>4. <b>System displays a list of facilities</b> ranked by distance and availability.</li> <li>5. <b>Responder selects the most suitable hospital.</b></li> <li>6. <b>System confirms the selection and updates the ambulance route.</b></li> <li>7. <b>Hospital staff is automatically notified</b> about the incoming patient and expected arrival time.</li> </ol>
<b>Alternative Sequence</b>	<p><b>If no nearby hospitals have available beds:</b></p> <ol style="list-style-type: none"> <li>a. <b>System expands the search radius</b> to include more distant hospitals.</li> <li>b. <b>System suggests the next best options</b>, even if further away.</li> </ol> <p><b>If a hospital's availability data is temporarily unavailable:</b></p> <ol style="list-style-type: none"> <li>a. <b>System flags it as unknown</b>, and suggests verified alternatives.</li> </ol> <p><b>If GPS tracking fails:</b></p> <ol style="list-style-type: none"> <li>a. <b>Responder manually selects a destination hospital</b> from the list.</li> <li>b. <b>System resumes route guidance once GPS reconnects.</b></li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Bed availability data must be <b>updated in real time</b> and securely transmitted.</li> <li>- The lookup process must complete in <b>under 2 seconds</b> to avoid delays.</li> <li>- The UI must present options <b>clearly and quickly</b>, optimized for mobile use in emergency vehicles.</li> <li>- All decisions and routing changes must be <b>logged for accountability</b>.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• A hospital with confirmed bed availability is selected for patient transport.</li> <li>• The ambulance is routed to the selected hospital.</li> <li>• Hospital staff are notified and can prepare for the patient’s arrival.</li> <li>• The system logs the route and destination decision.</li> </ul>

UC_SYS_01	User Login
<b>Summary</b>	The system shall allow users to securely log in using their credentials to access their respective dashboards and functionalities.
<b>Dependency</b>	None

## **Electronic Healthcare System Requirements Specification**

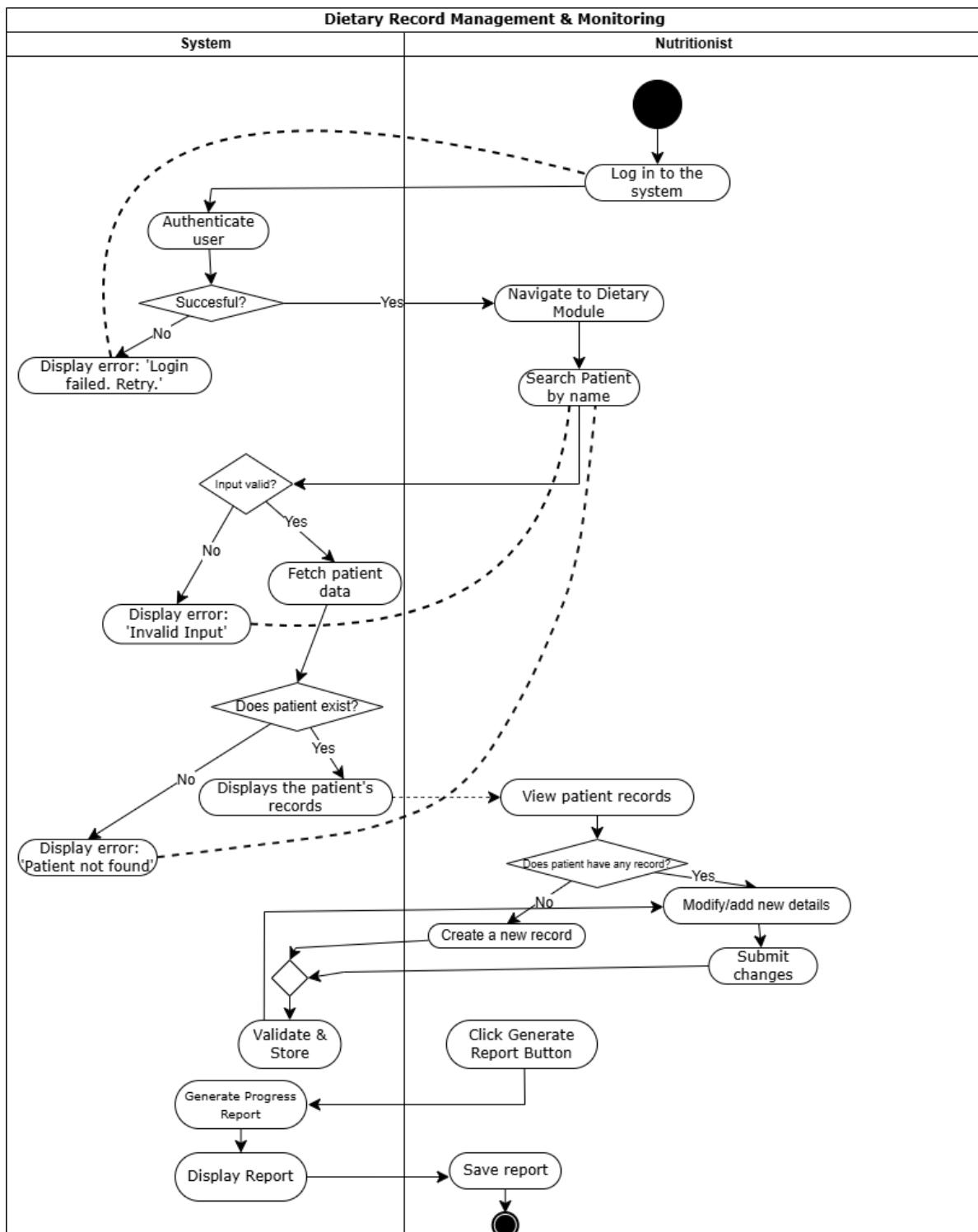
<b>Actors</b>	<b>Primary Actor:</b> Any System User (e.g., Patient, Doctor, Nutritionist, Pharmacy Staff, IT Support)
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The user must have a registered account.</li> <li>• The system must be online and accessible.</li> </ul>
<b>Main Sequence</b>	<ol style="list-style-type: none"> <li>1. <b>User accesses the login page</b> via web or mobile platform.</li> <li>2. <b>User enters login credentials</b>, including username/email and password.</li> <li>3. <b>System validates the credentials</b> against the user database.</li> <li>4. <b>If credentials are valid:</b> <ol style="list-style-type: none"> <li>a. <b>System authenticates the user.</b></li> <li>b. <b>System redirects the user to their personalized dashboard</b> (based on role).</li> </ol> </li> <li>5. <b>System logs the login event</b> with a timestamp for audit purposes.</li> </ol>
<b>Alternative Sequence</b>	<p><b>If credentials are invalid:</b></p> <ol style="list-style-type: none"> <li>a. <b>System displays an error message.</b></li> <li>b. <b>User is prompted to retry</b> or use "Forgot Password" to reset credentials.</li> </ol> <p><b>If login fails due to system error or network issue:</b></p> <ol style="list-style-type: none"> <li>a. <b>System notifies the user of a temporary issue.</b></li> <li>b. <b>User is advised to try again later.</b></li> </ol> <p><b>If multiple failed attempts occur:</b></p> <ol style="list-style-type: none"> <li>a. <b>System may temporarily lock the account</b> and prompt for identity verification.</li> </ol>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>- Login must be secured with <b>end-to-end encryption</b>.</li> <li>- System must support <b>multi-factor authentication</b> where applicable.</li> <li>- Login should complete in <b>under 3 seconds</b> under normal conditions.</li> <li>- <b>Failed login attempts</b> must be logged and monitored for security.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The user is authenticated and granted access to the system.</li> <li>• The system logs the login event and user role.</li> <li>• All role-based permissions are initialized for the active session.</li> </ul>

## 5. Diagrams

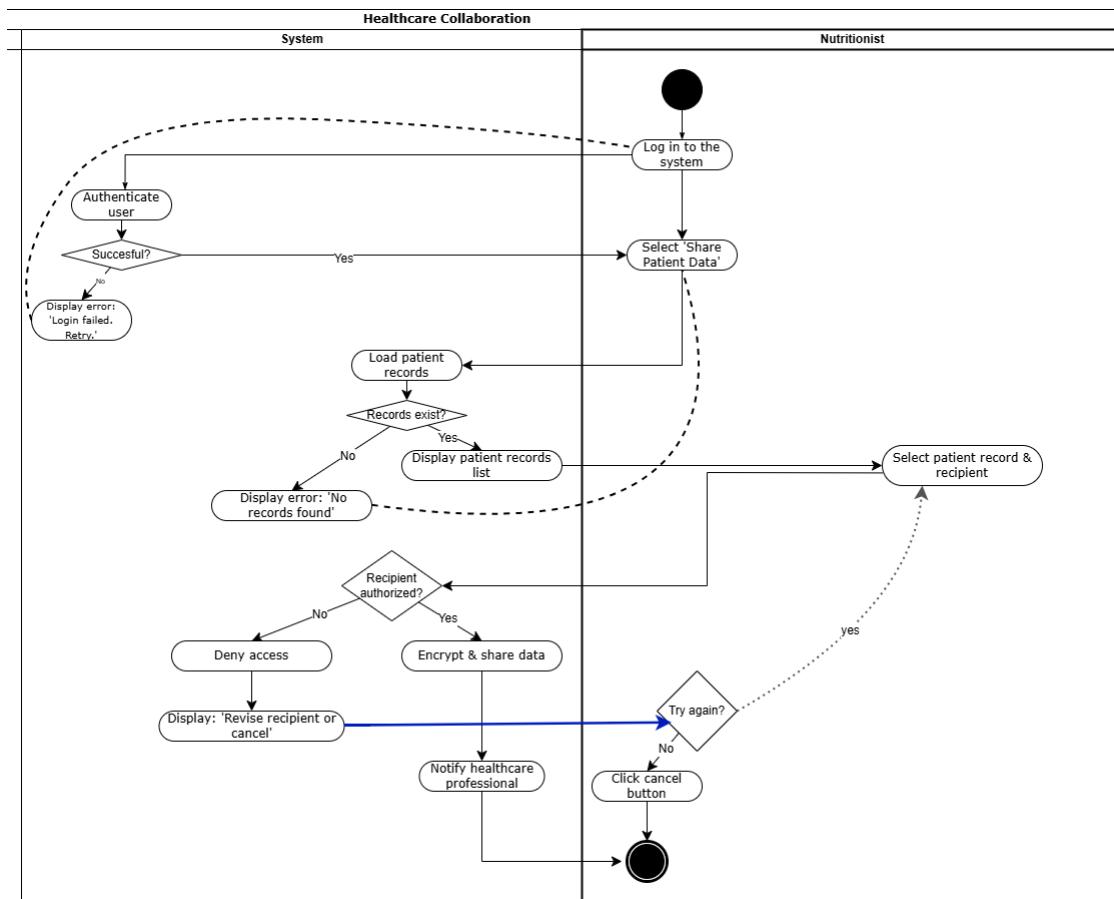


## 5.2 Activity Diagram

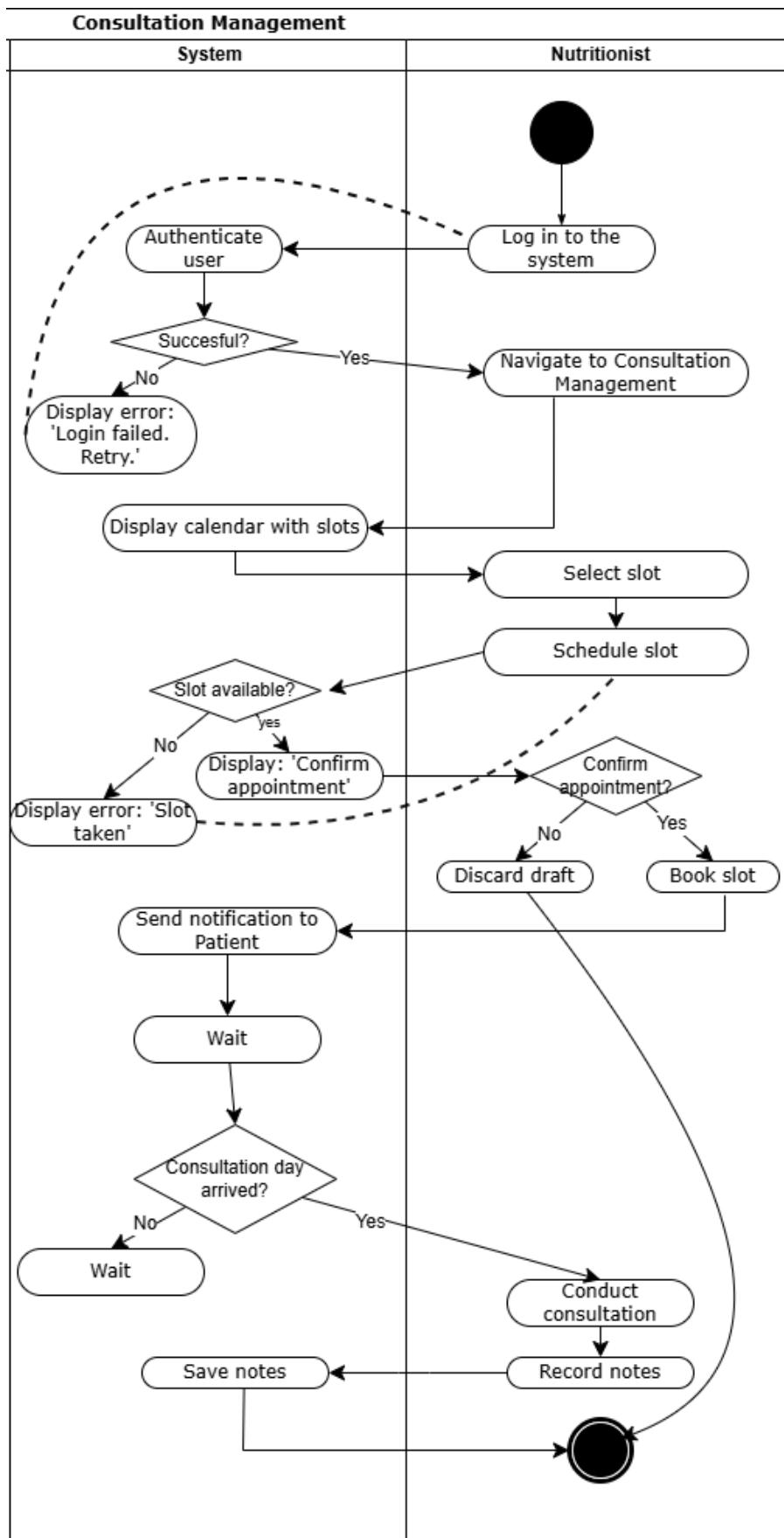
### Nutritionist -Belina Durmishi



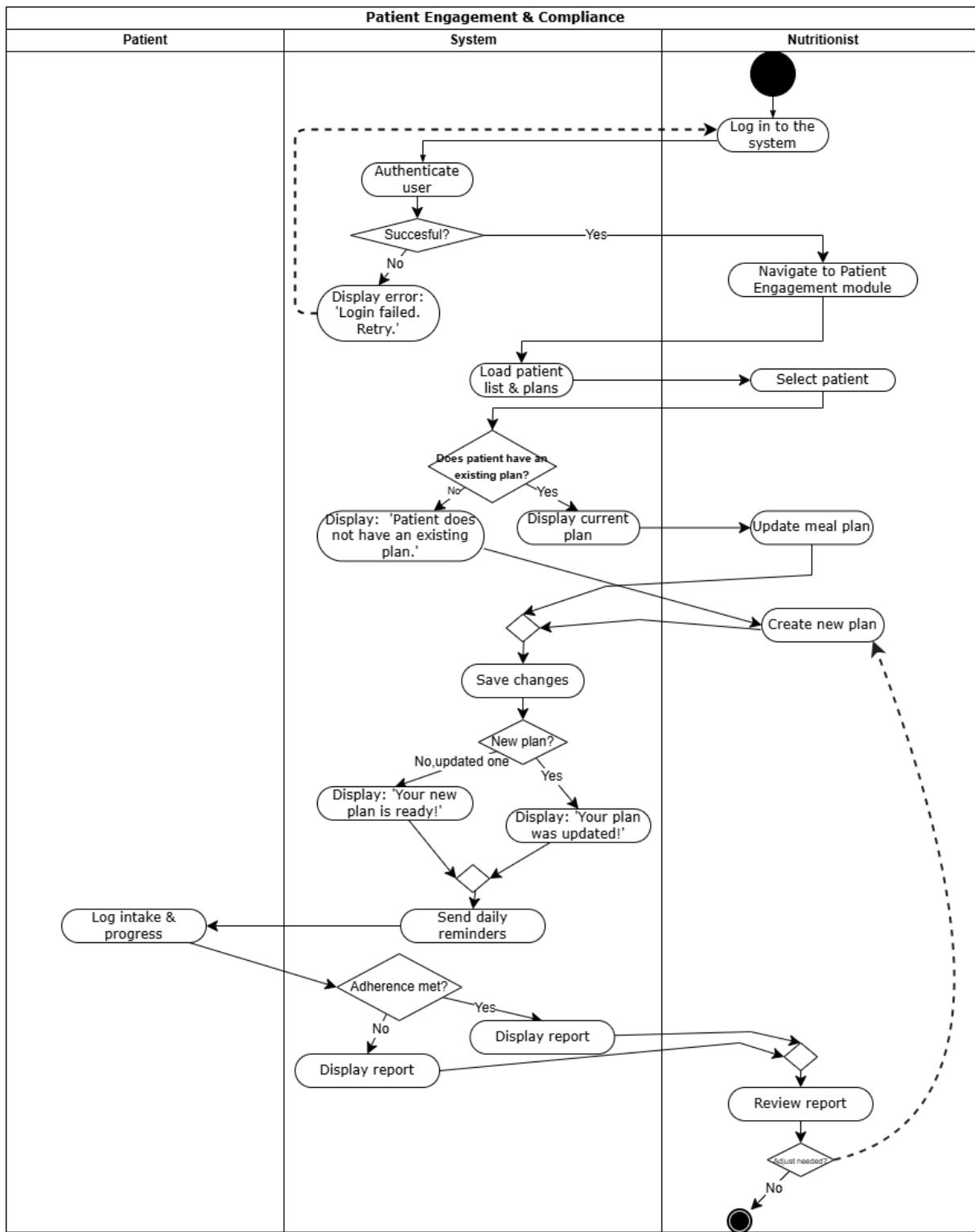
## Electronic Healthcare System Requirements Specification



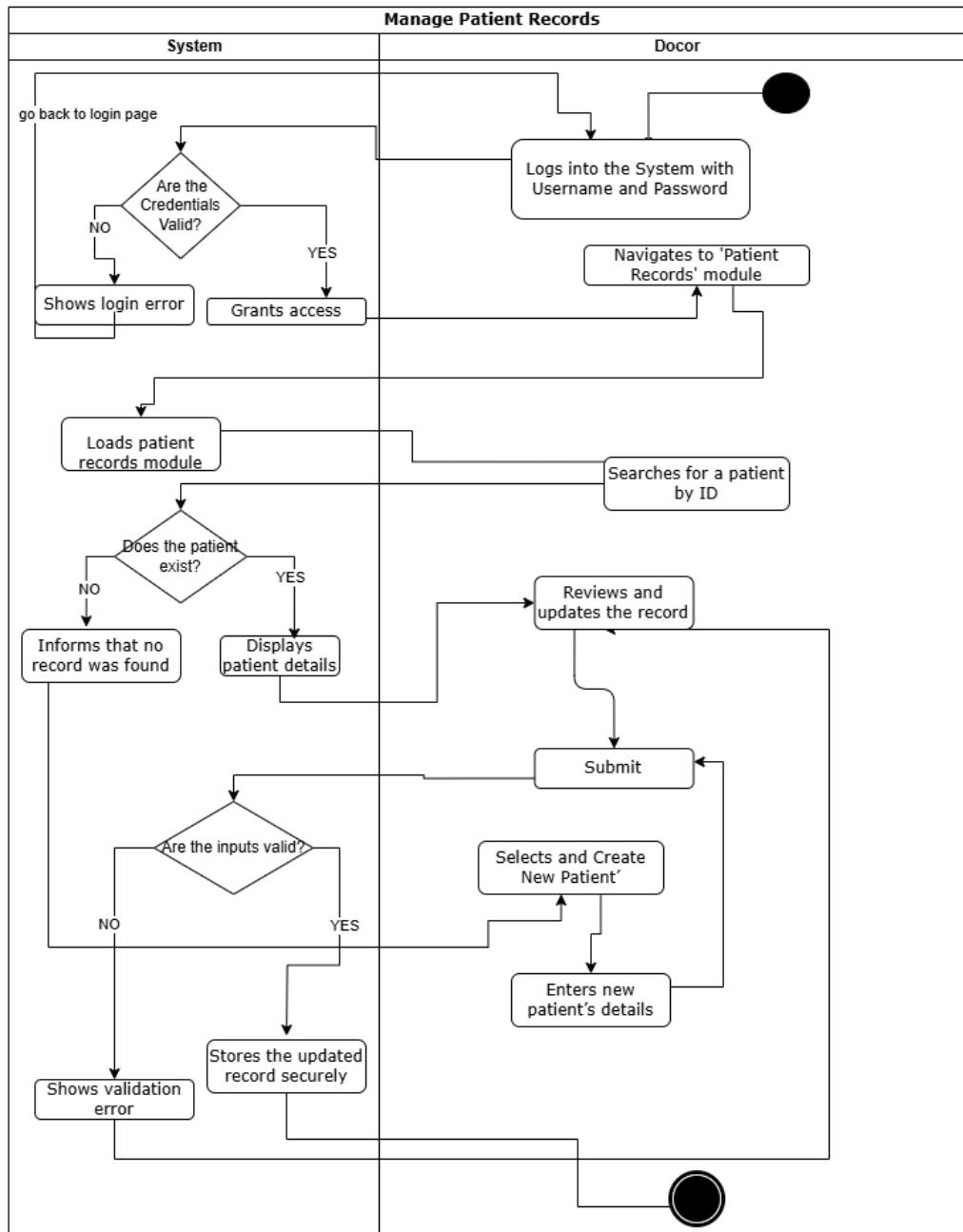
## Electronic Healthcare System Requirements Specification



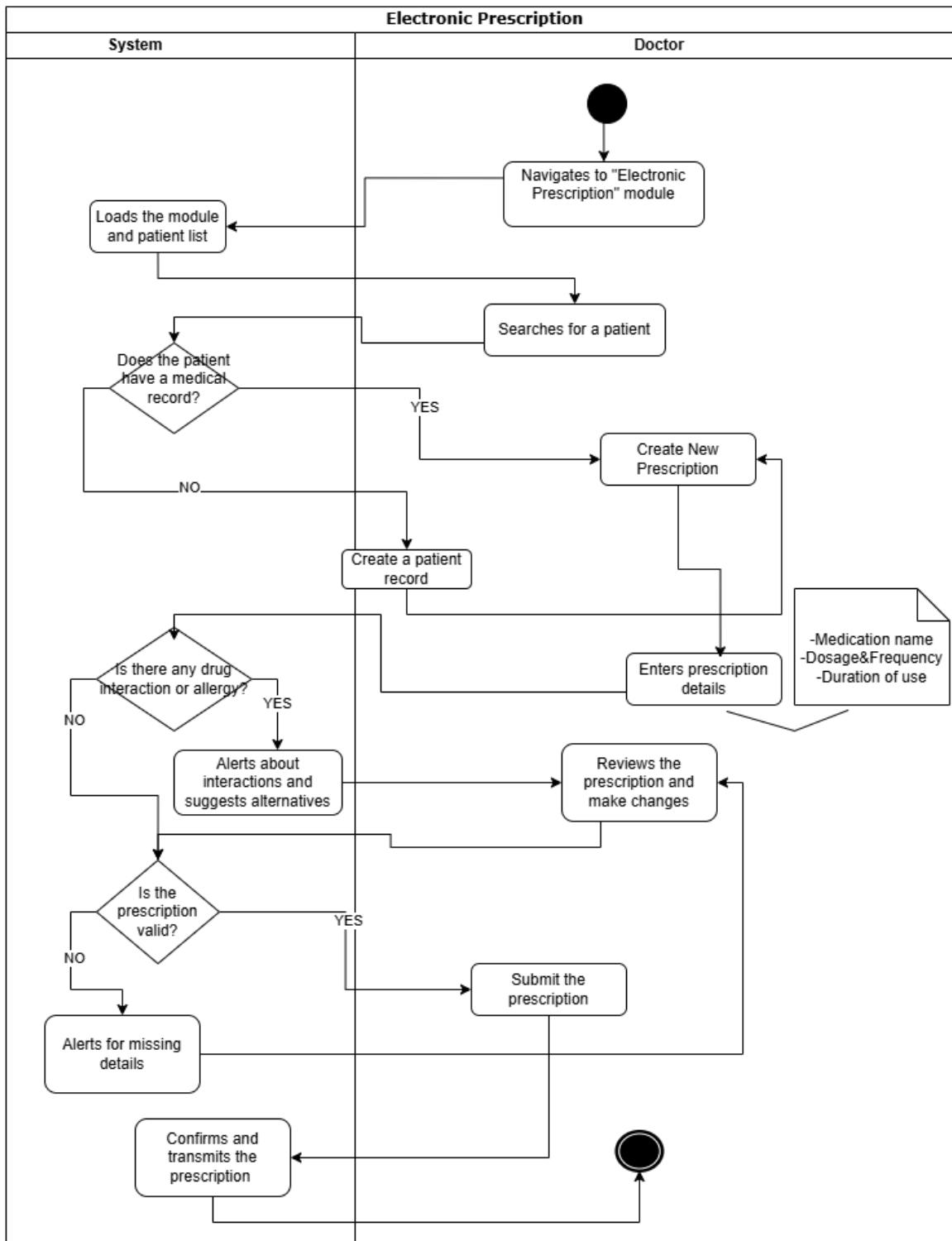
## Electronic Healthcare System Requirements Specification



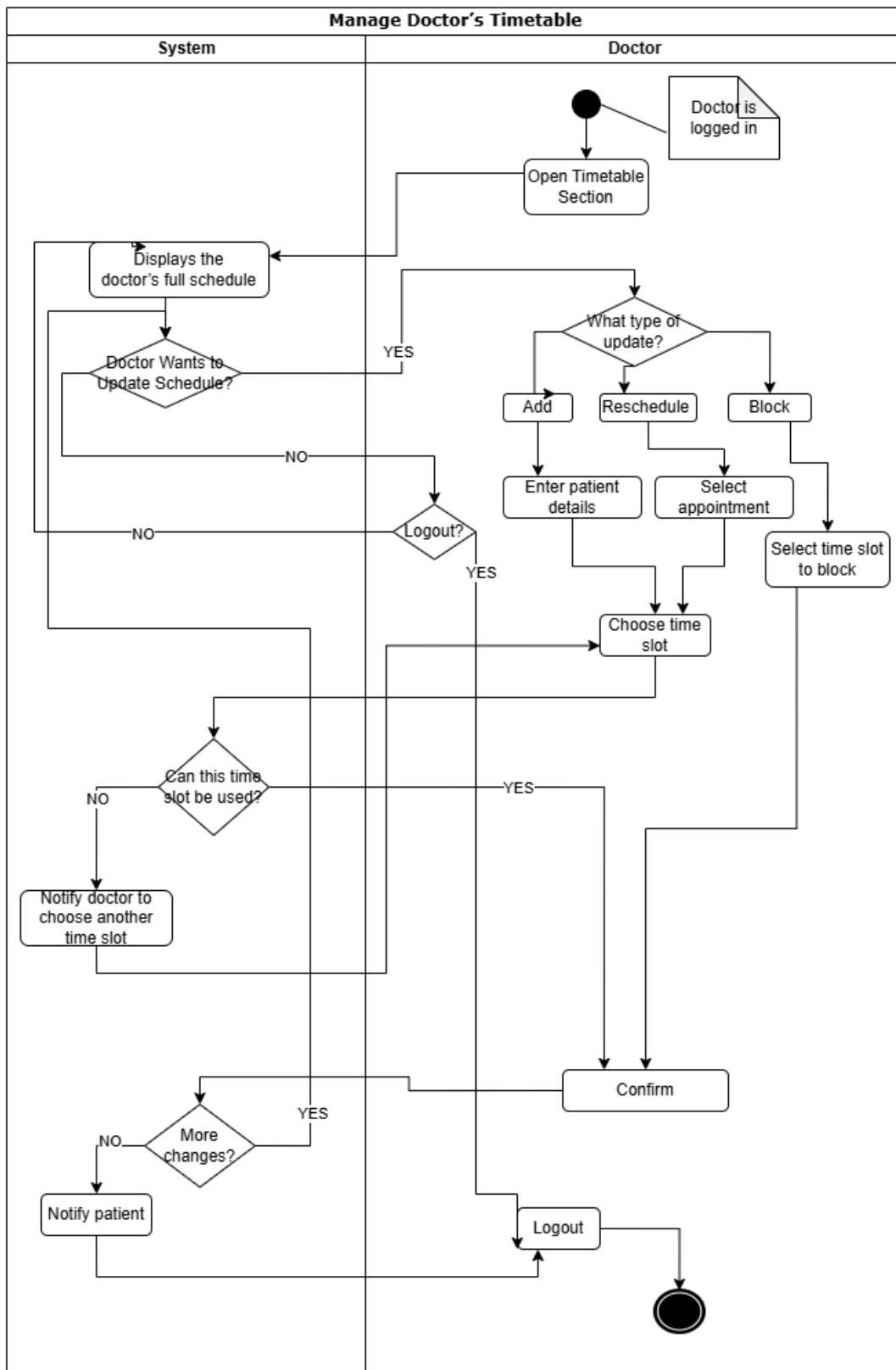
## **Doctor-Antea Koxherri**



## Electronic Healthcare System Requirements Specification

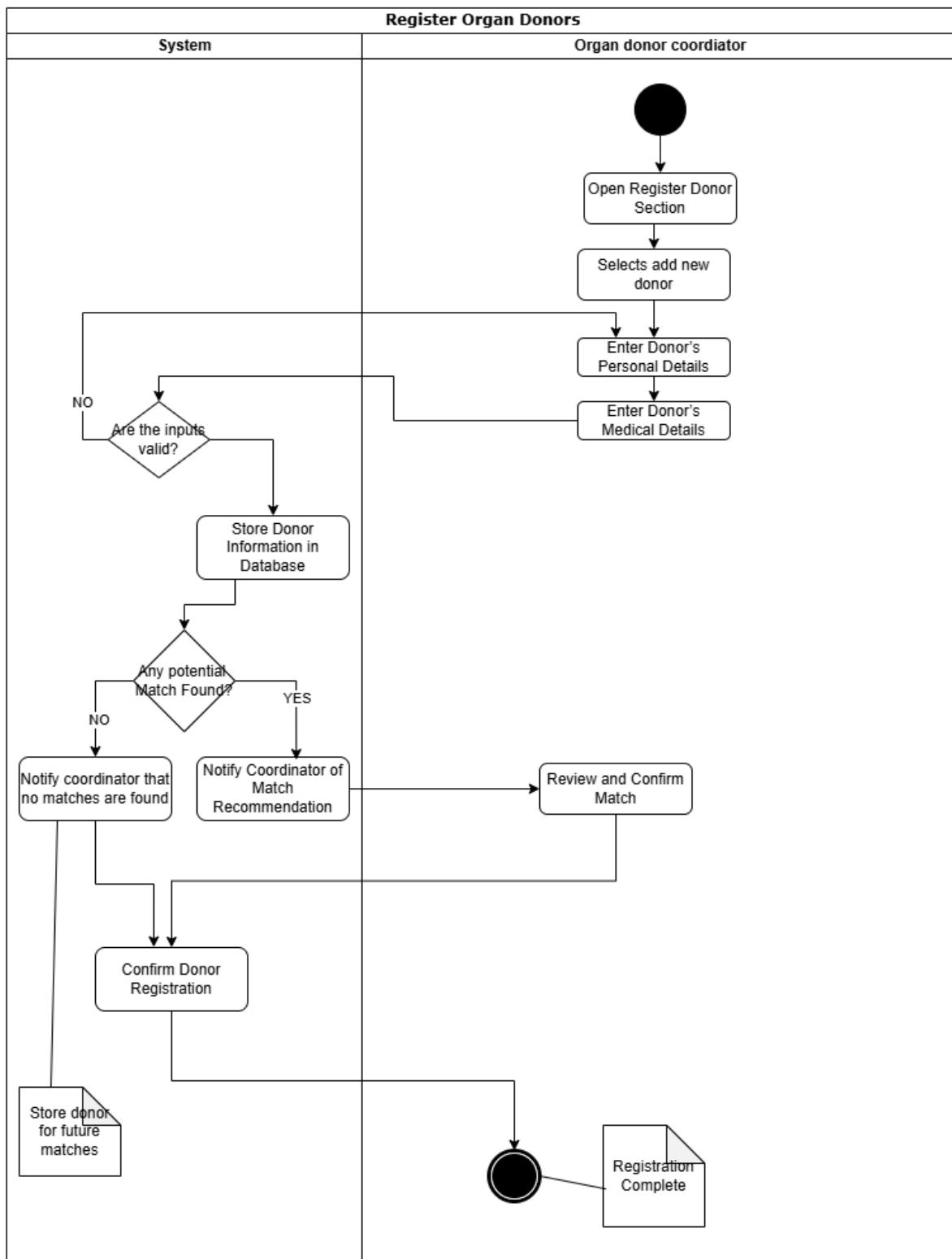


## Electronic Healthcare System Requirements Specification

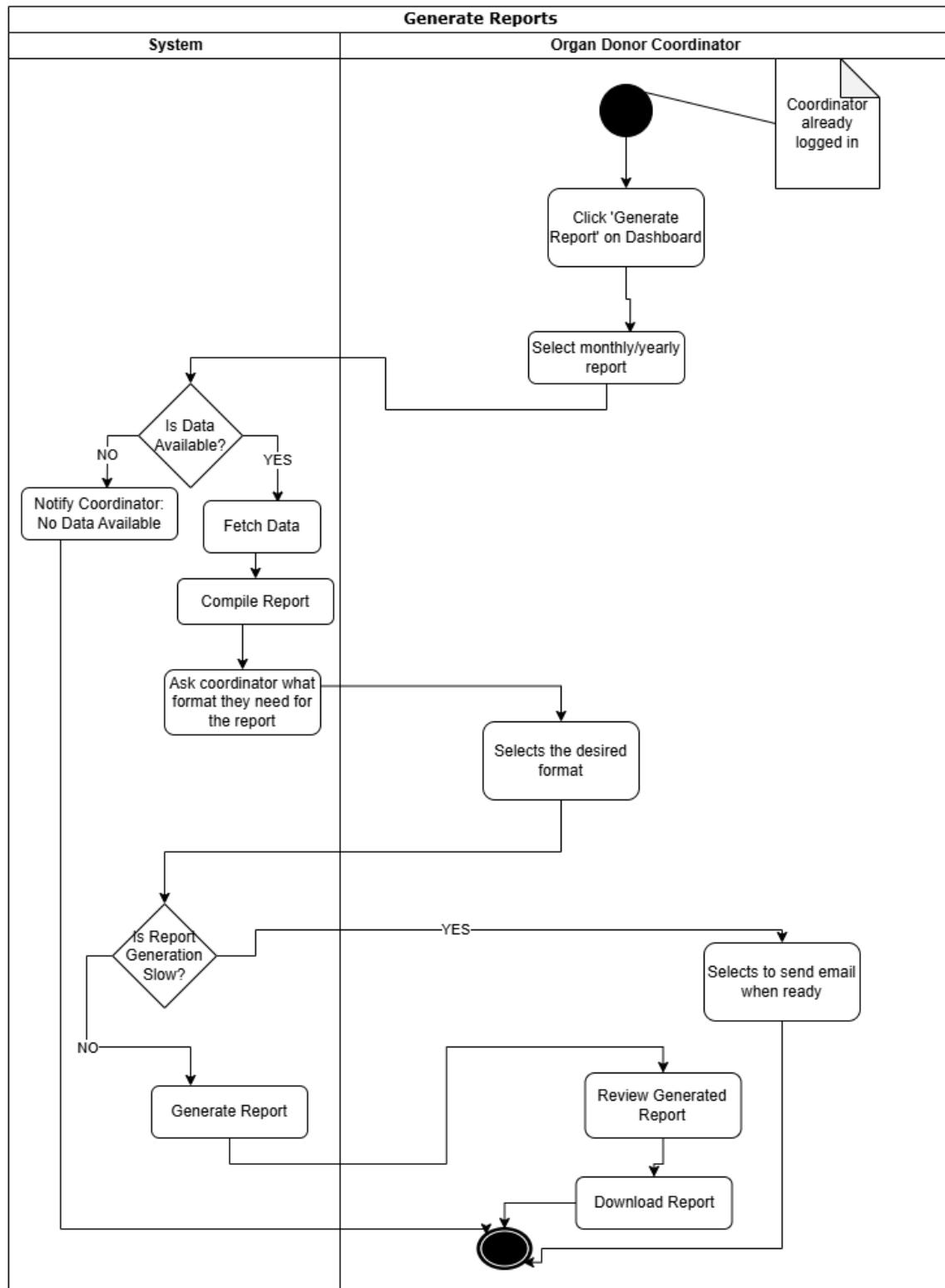


**Organ Donor Coordinator-Antea Koxherri**

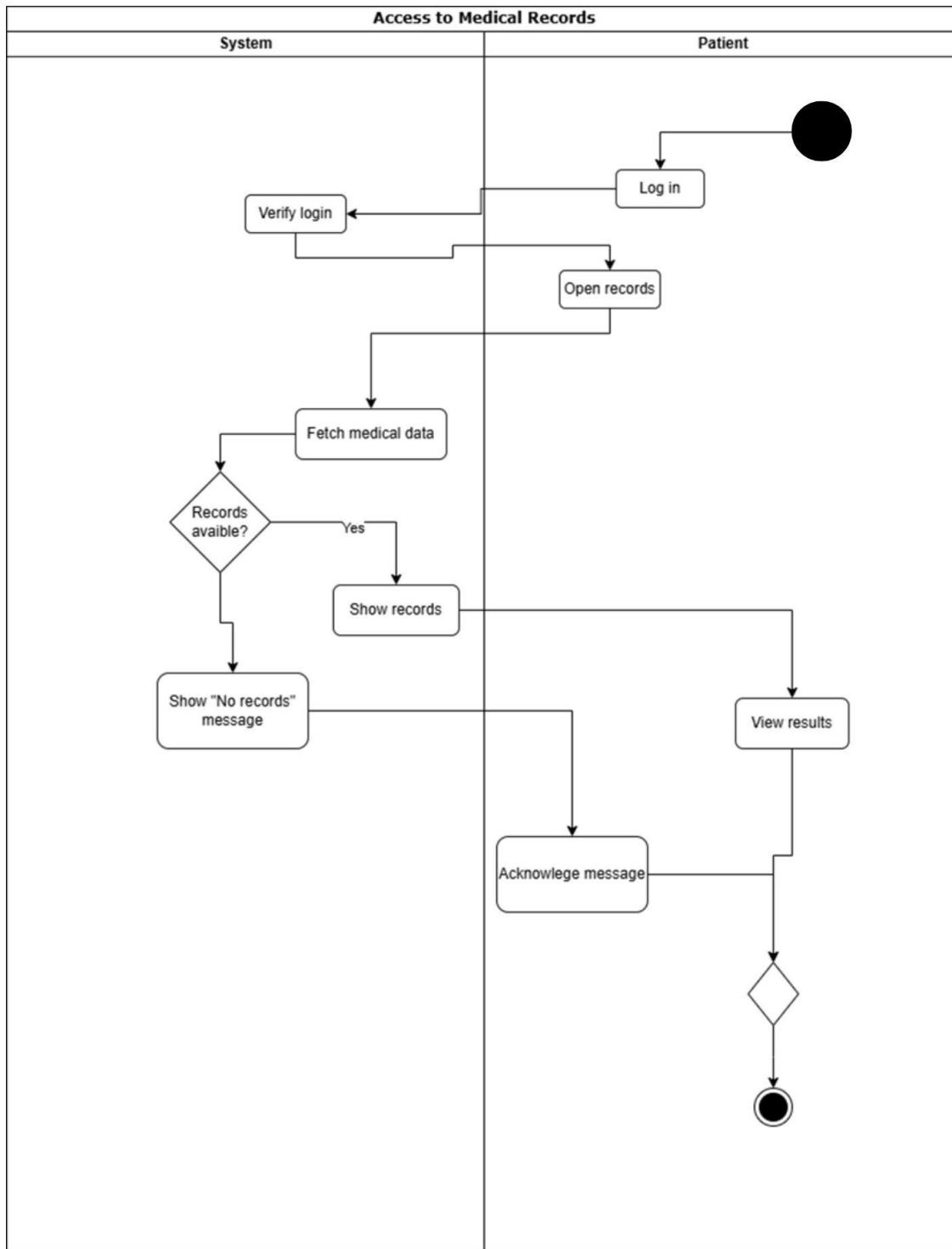
## Electronic Healthcare System Requirements Specification



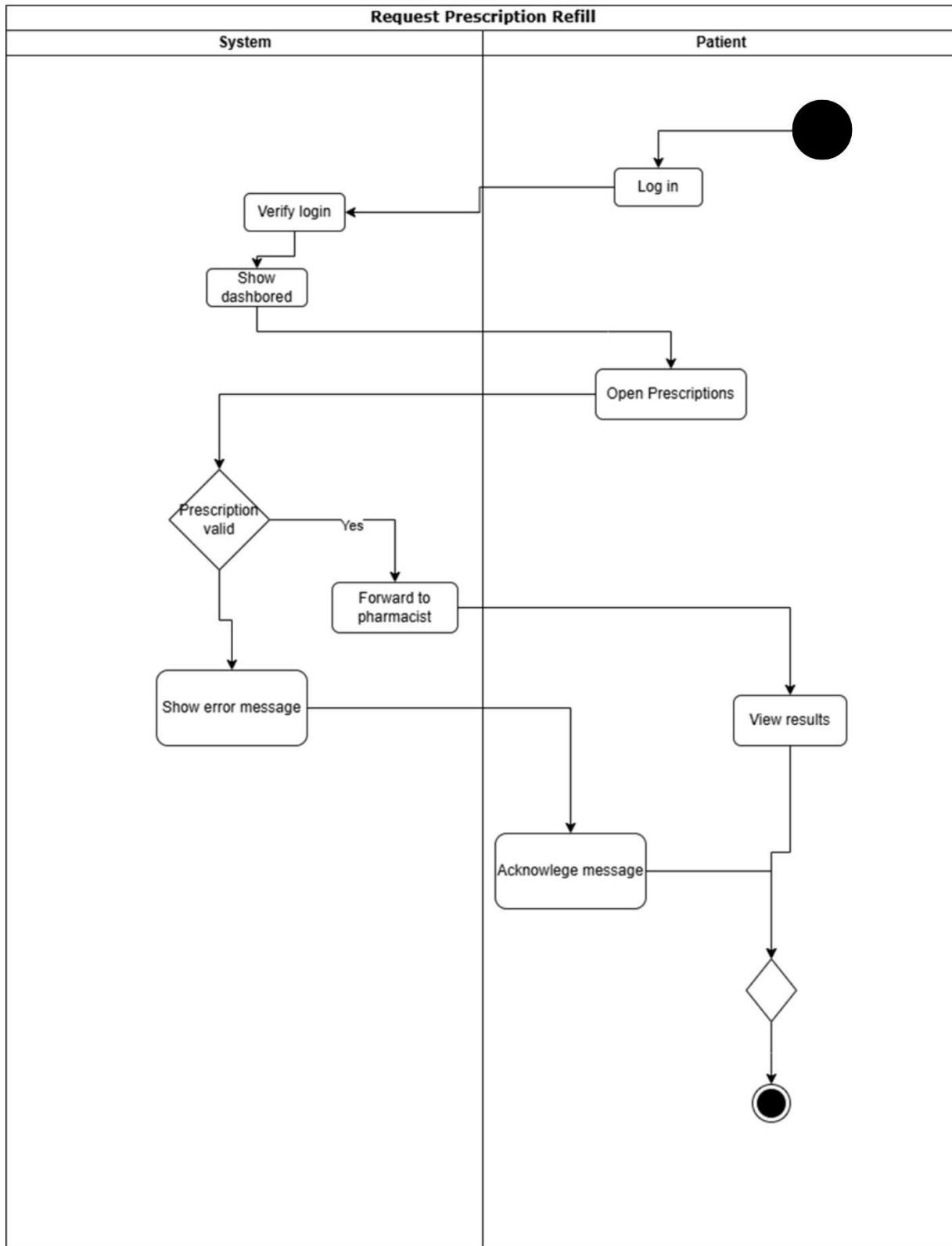
## Electronic Healthcare System Requirements Specification



**Patient- Elisona Doku**

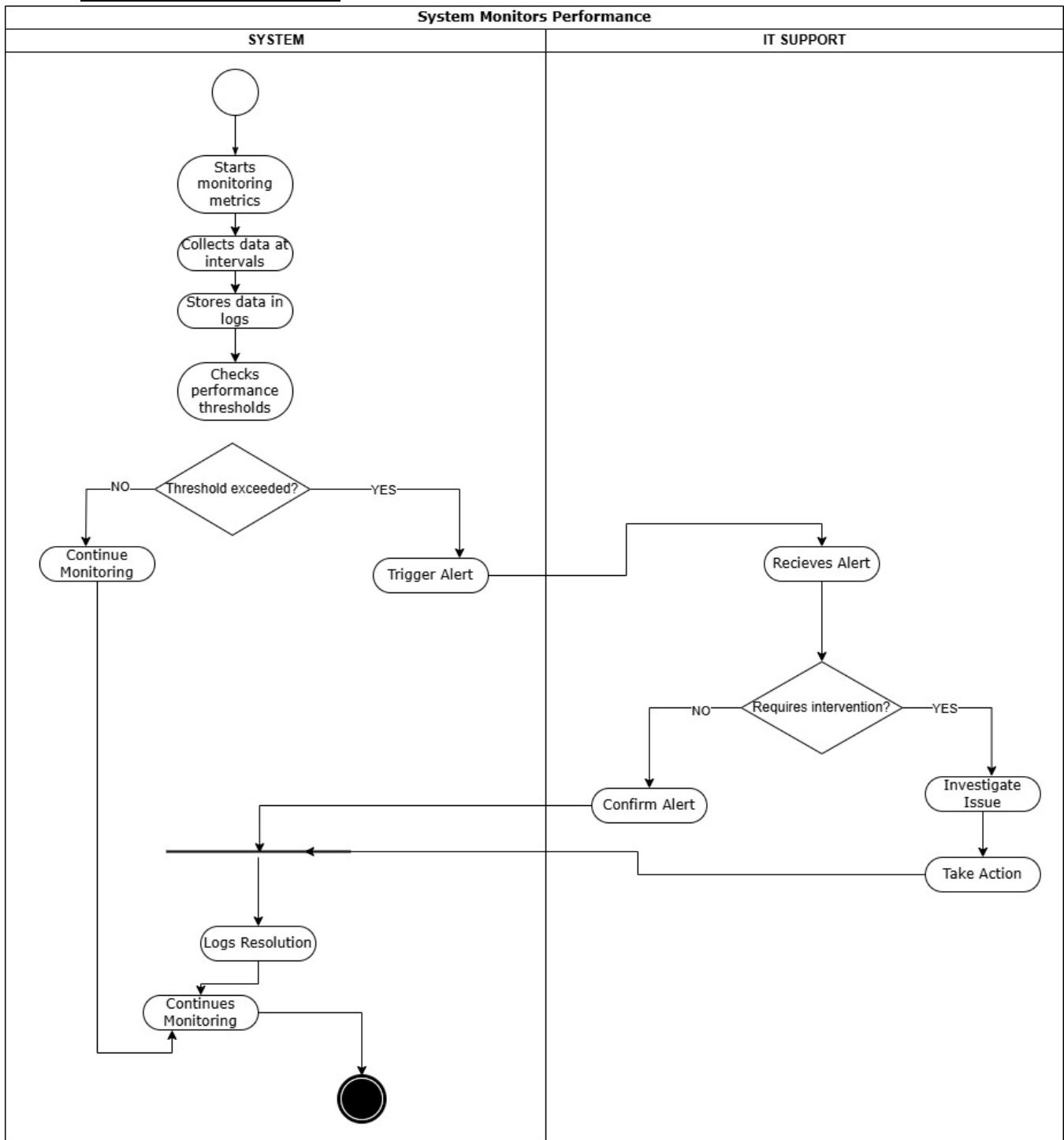


## Electronic Healthcare System Requirements Specification

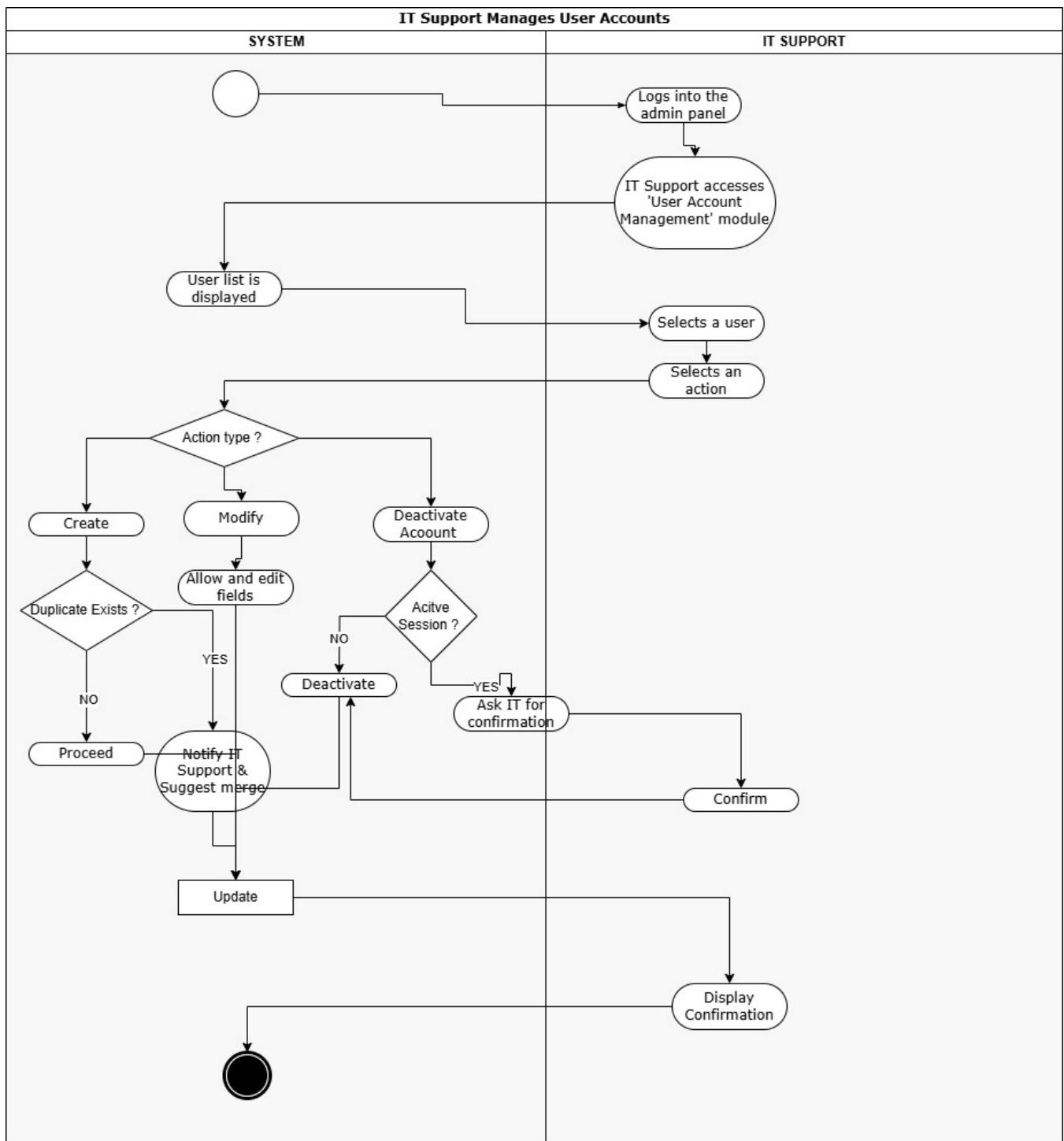


## Electronic Healthcare System Requirements Specification

### IT Support-Elkier Ago

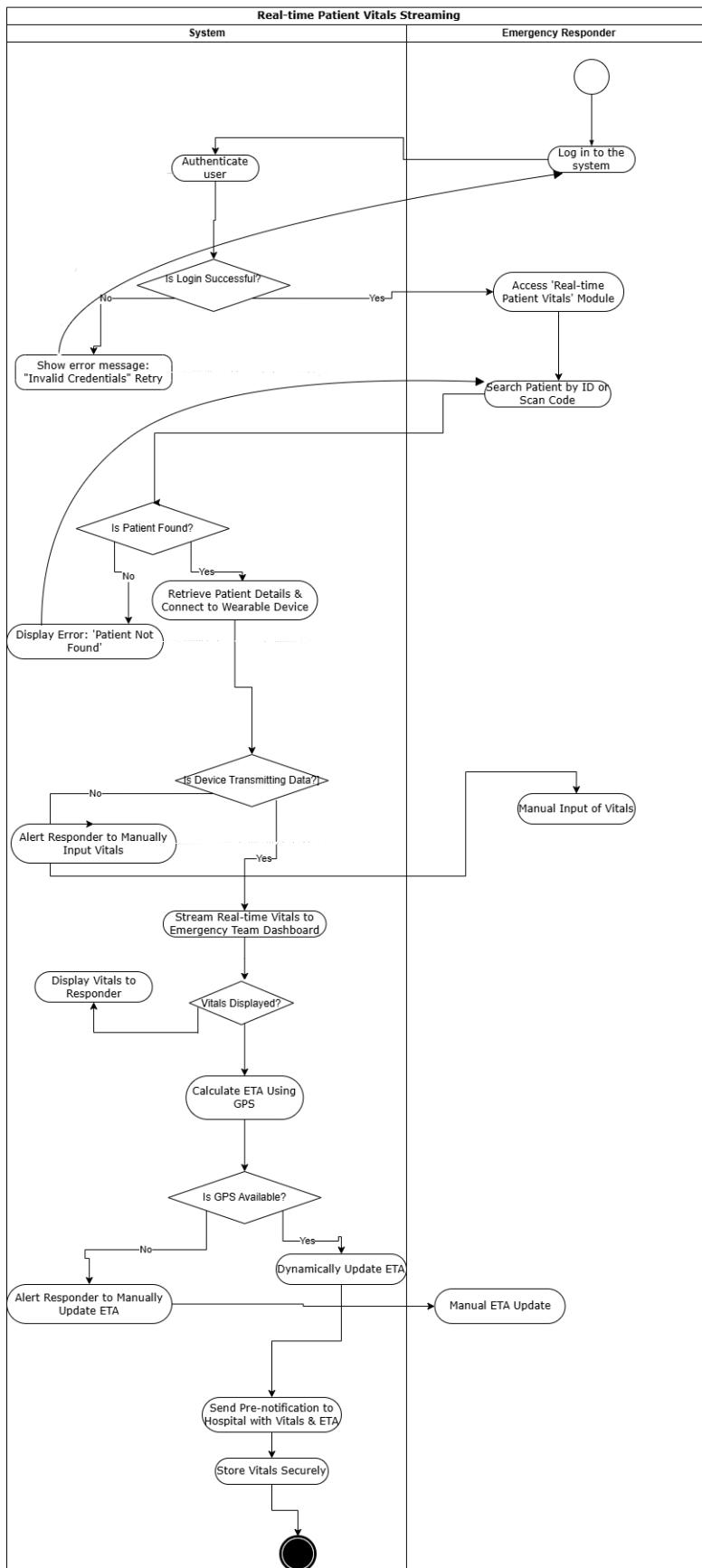


## Electronic Healthcare System Requirements Specification

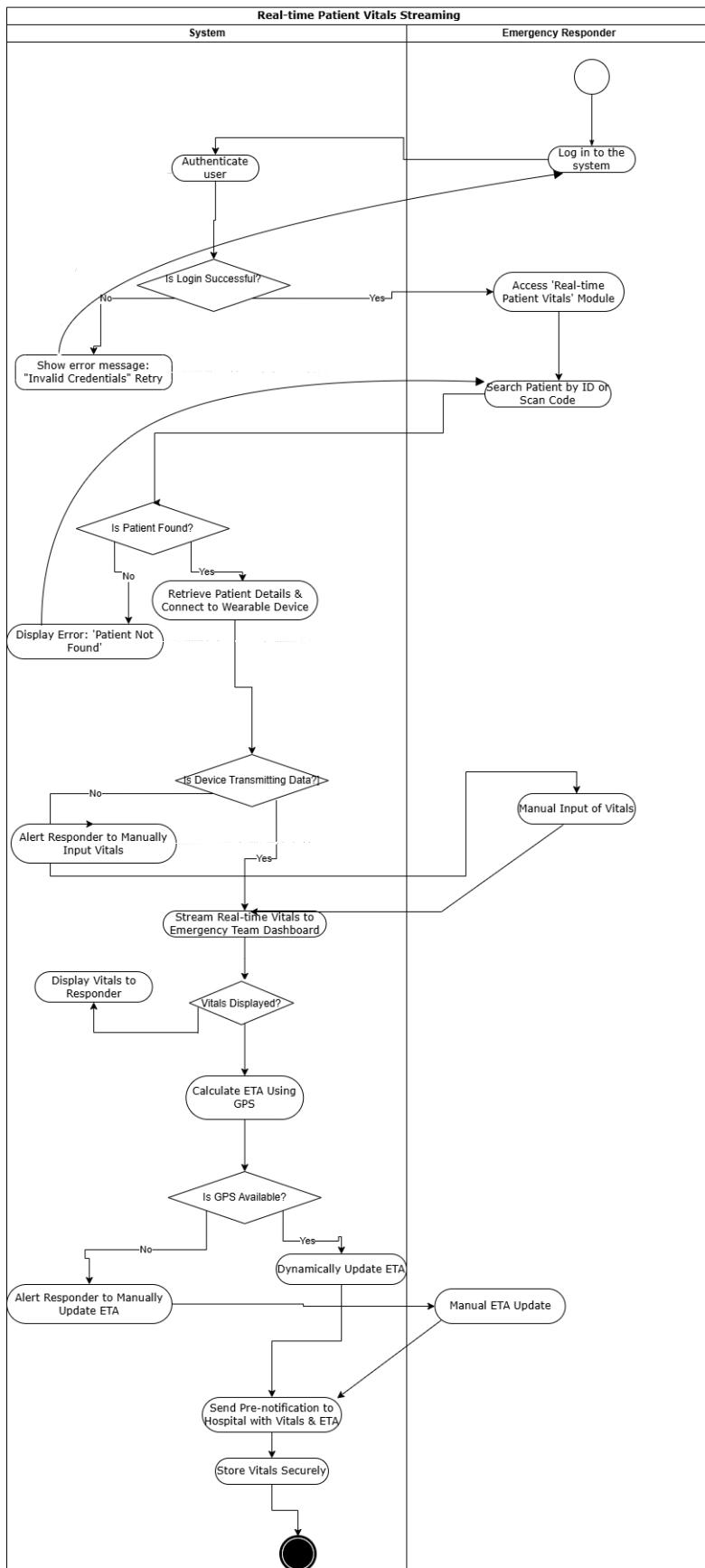


# Electronic Healthcare System Requirements Specification

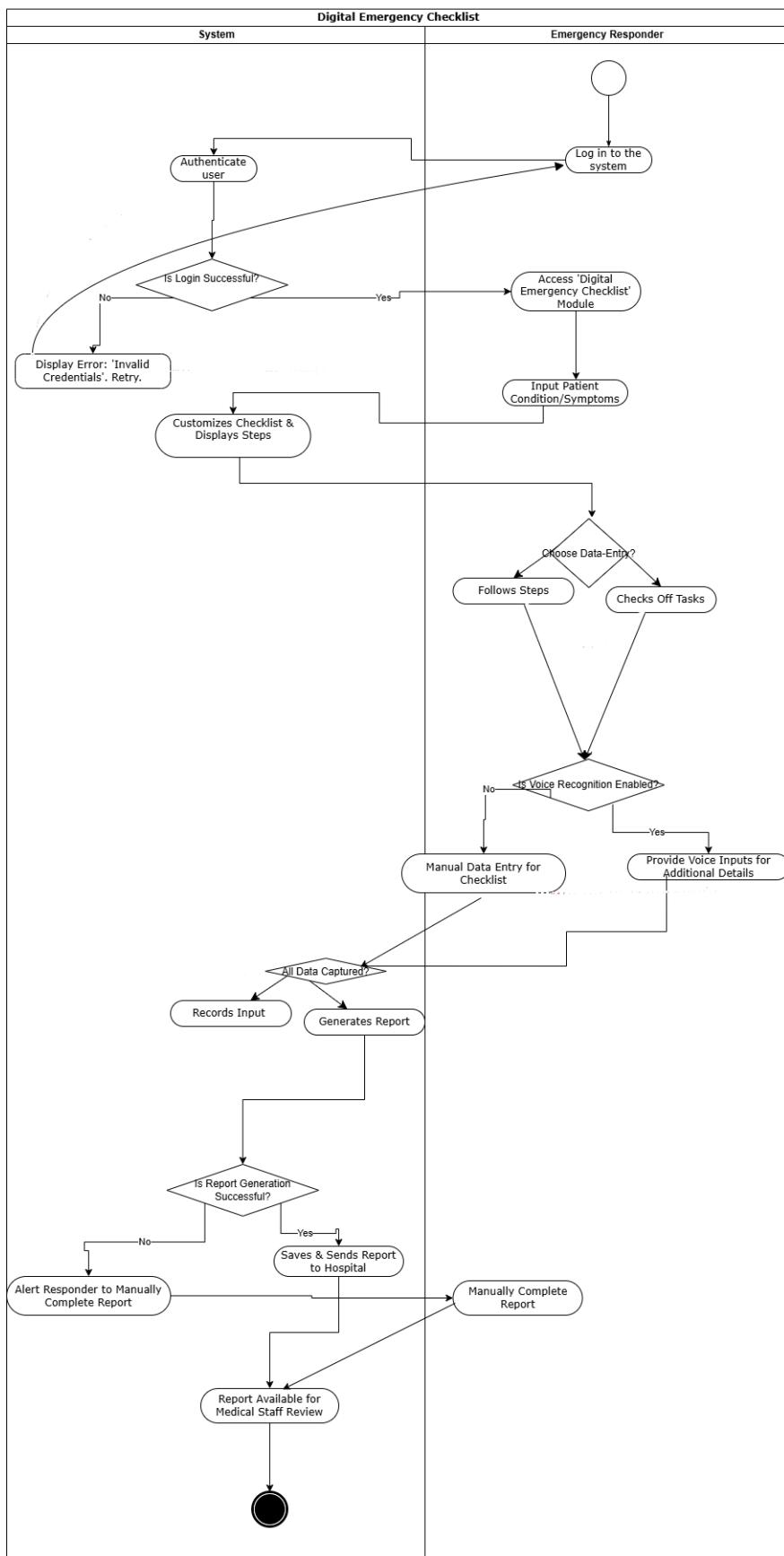
## Emergency Service -Evelina Gace



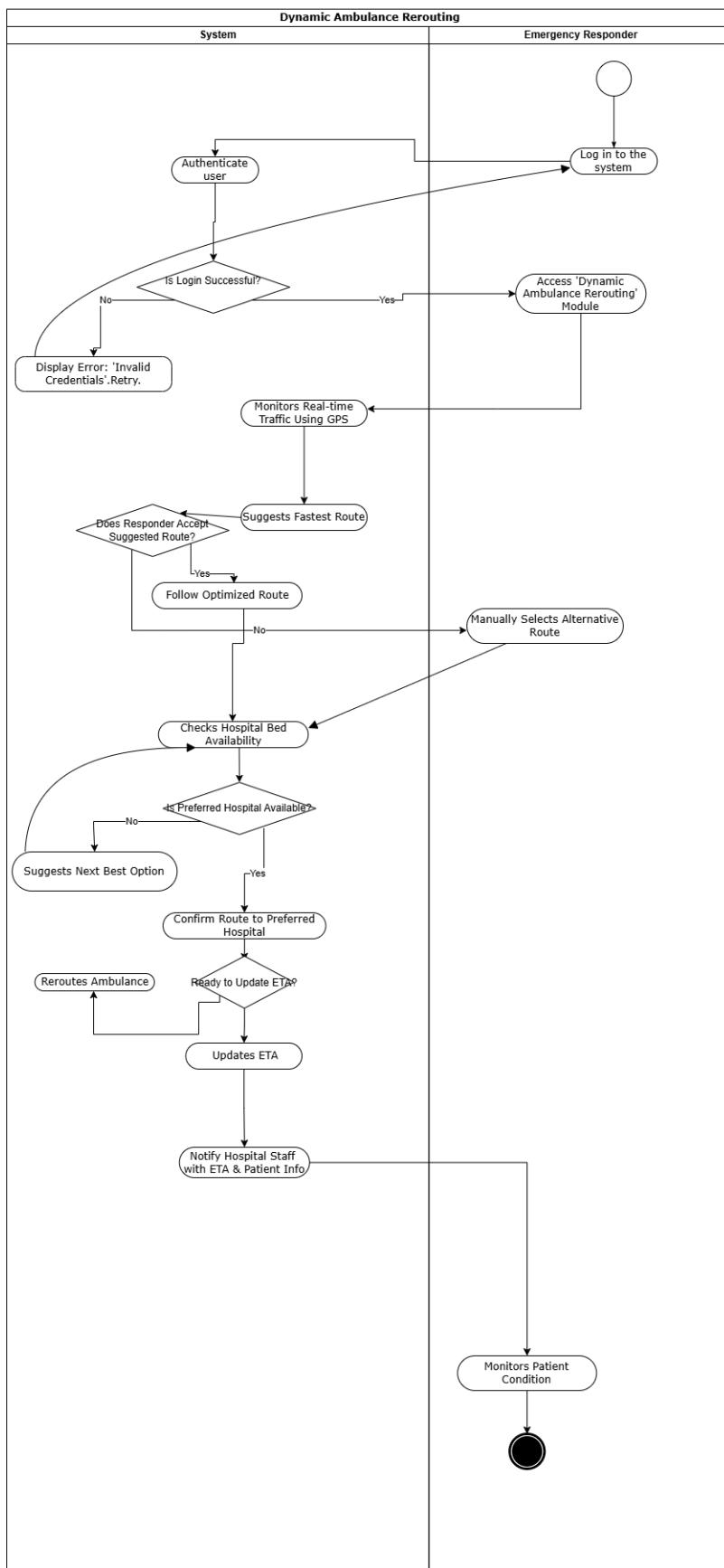
## Electronic Healthcare System Requirements Specification



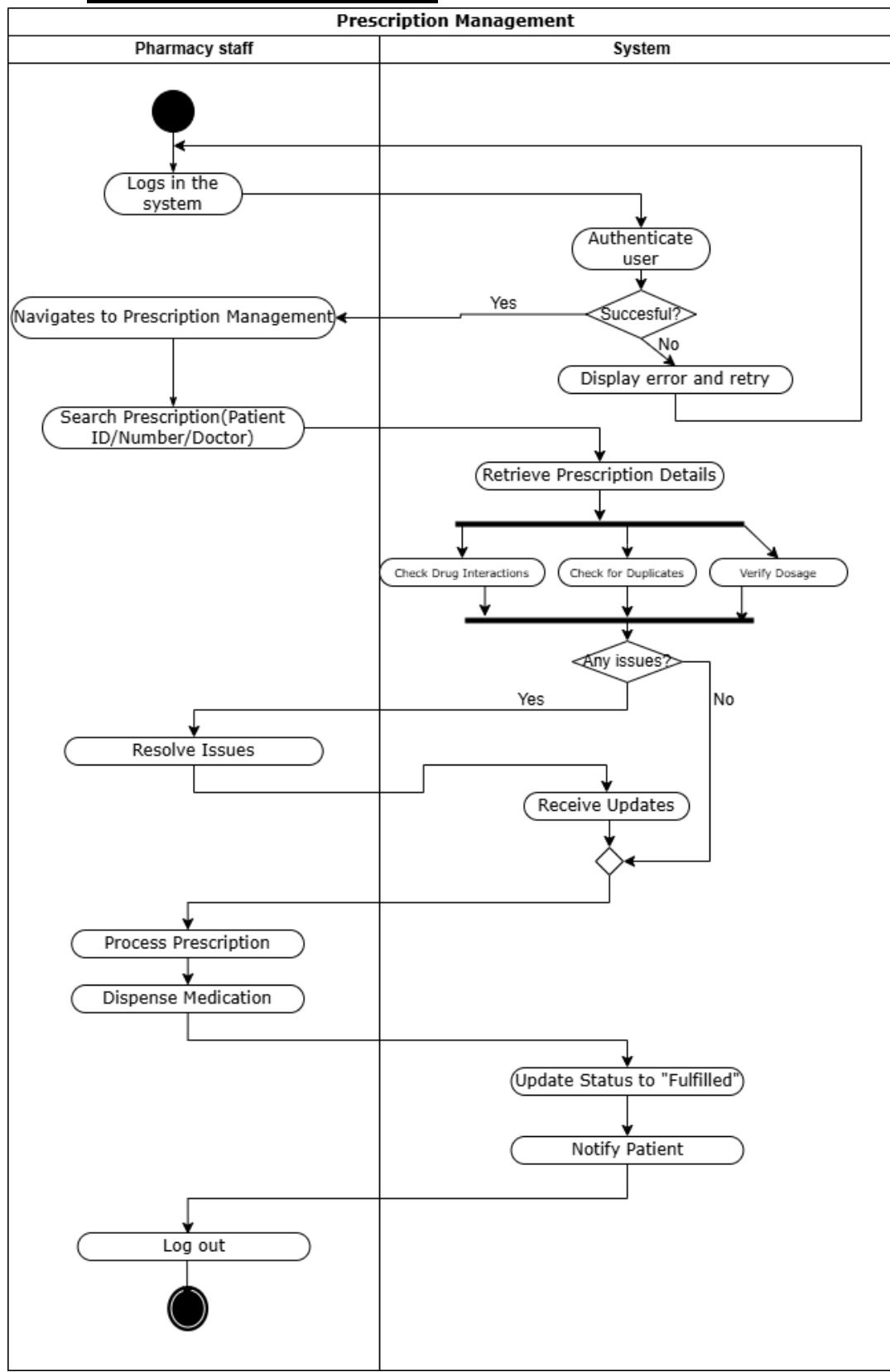
## Electronic Healthcare System Requirements Specification



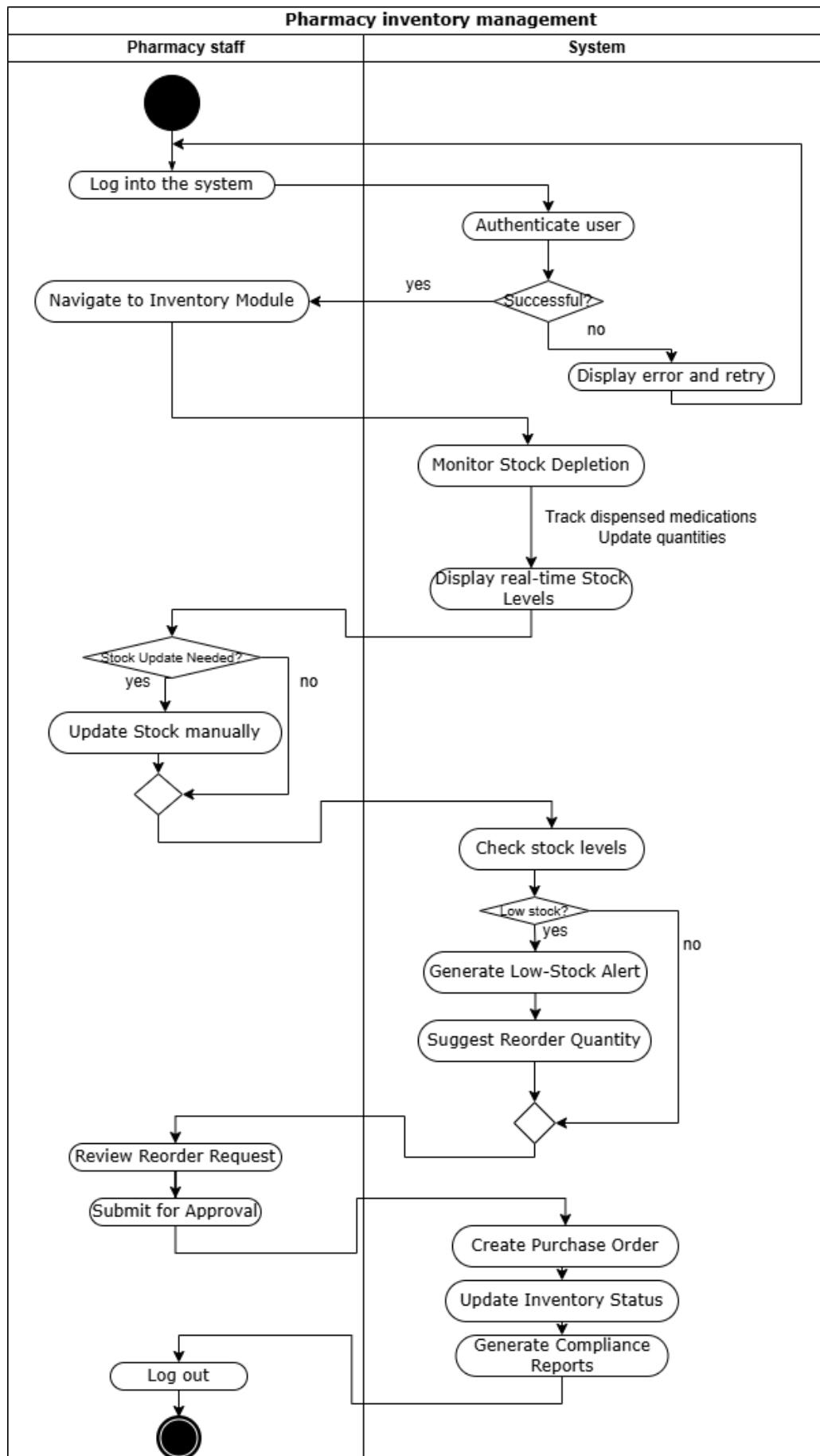
## Electronic Healthcare System Requirements Specification



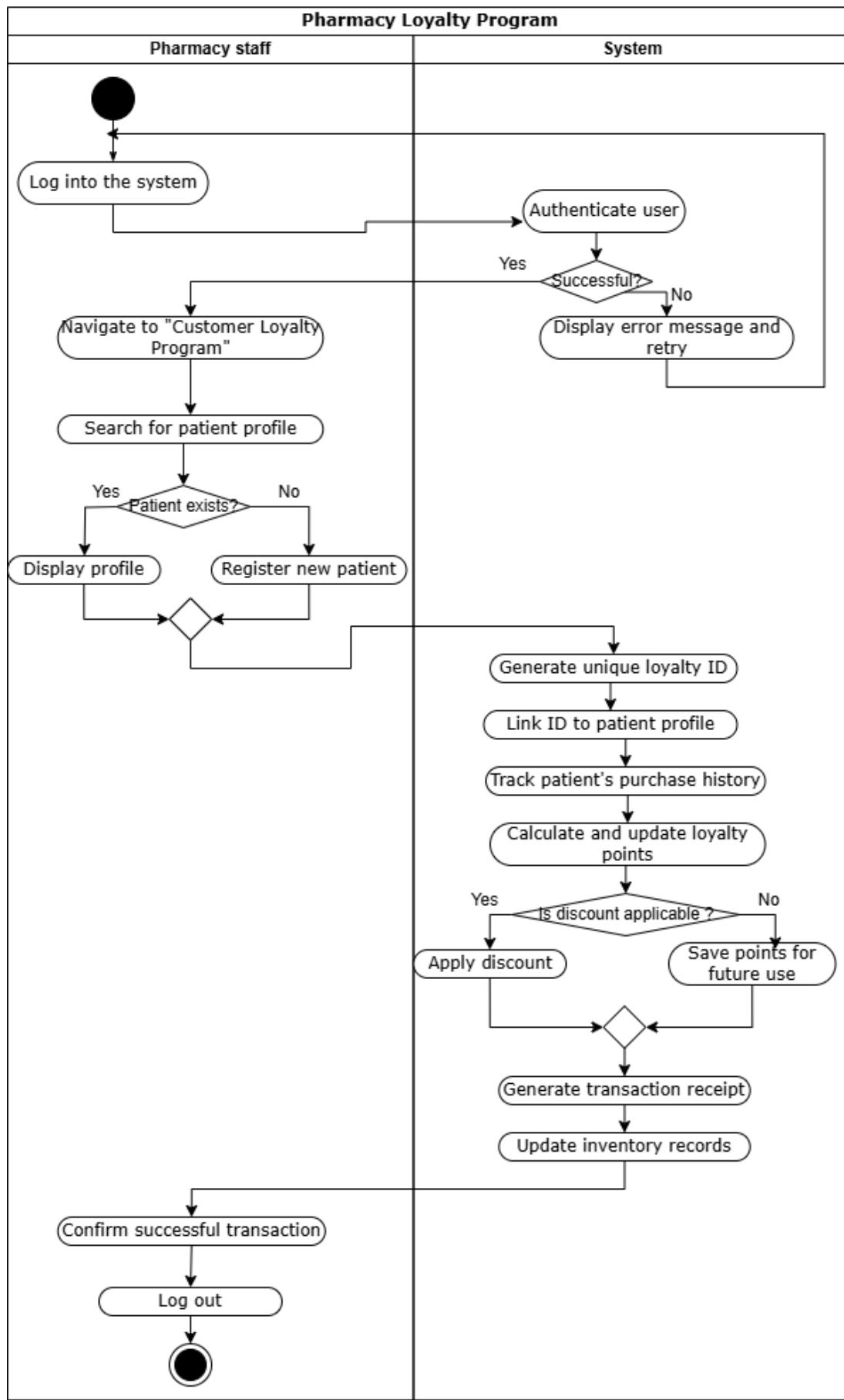
### **Pharmacy Staff-Flavia Koco**



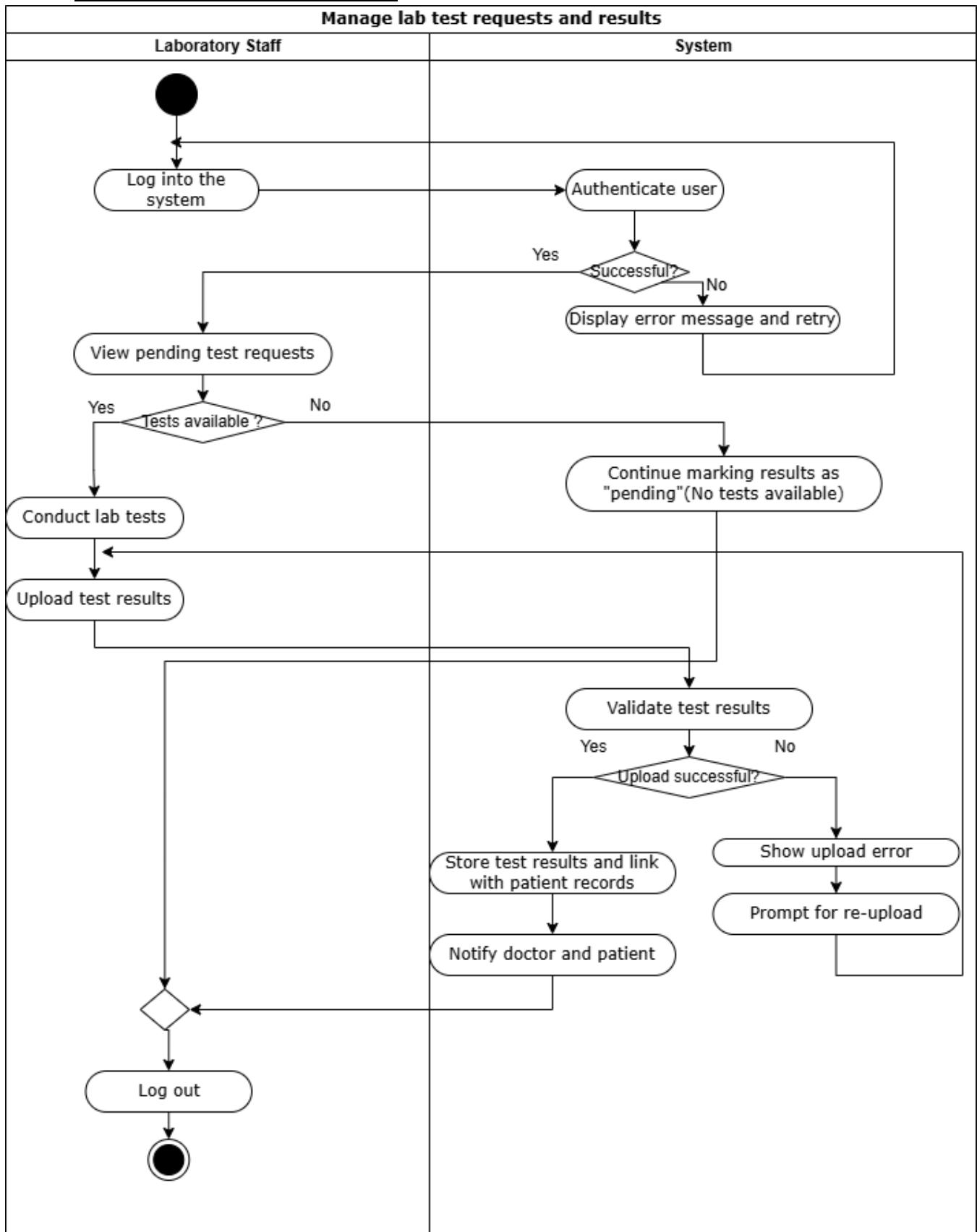
## Electronic Healthcare System Requirements Specification



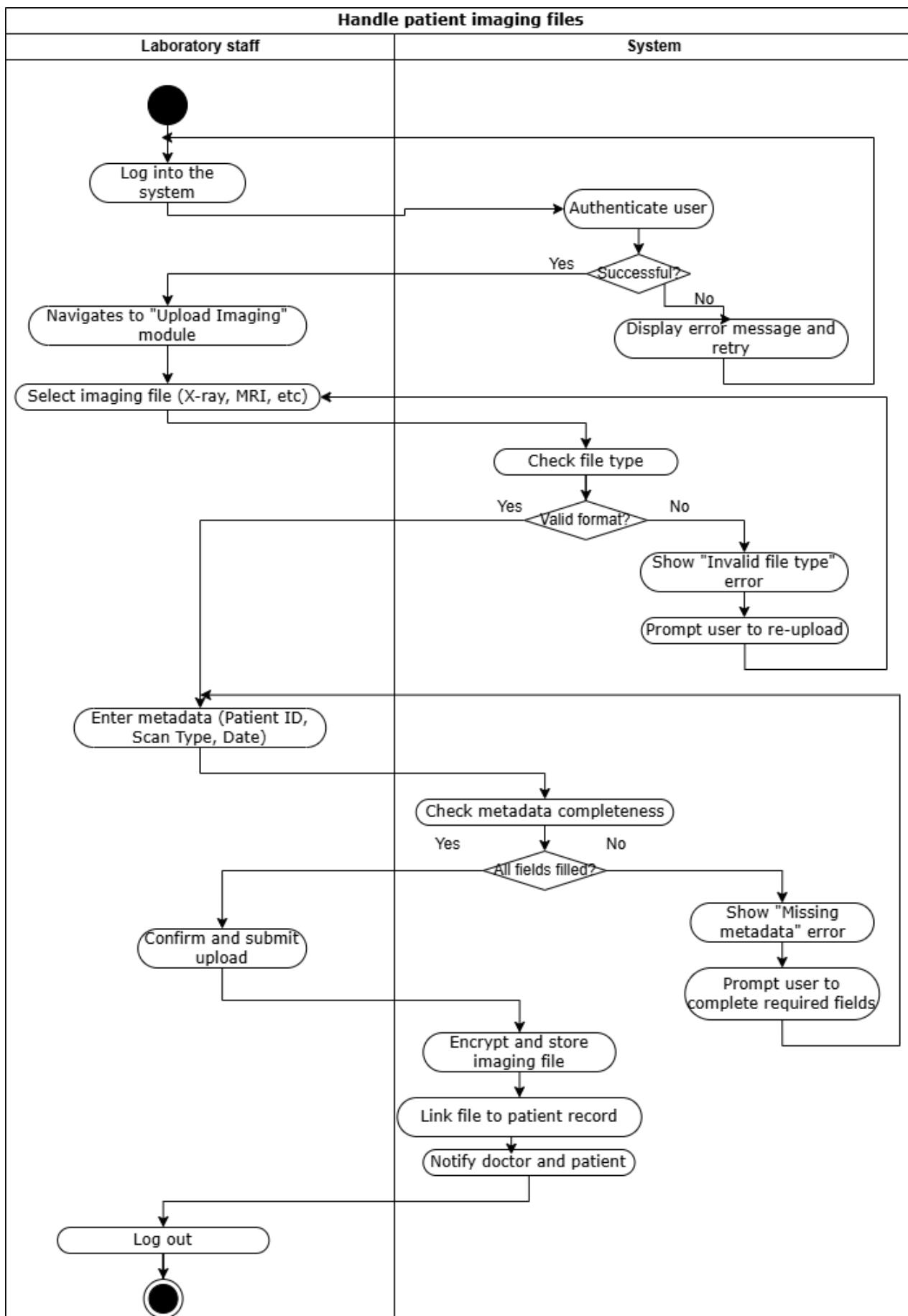
## Electronic Healthcare System Requirements Specification



**Laboratory Staff- Flavia Koco**



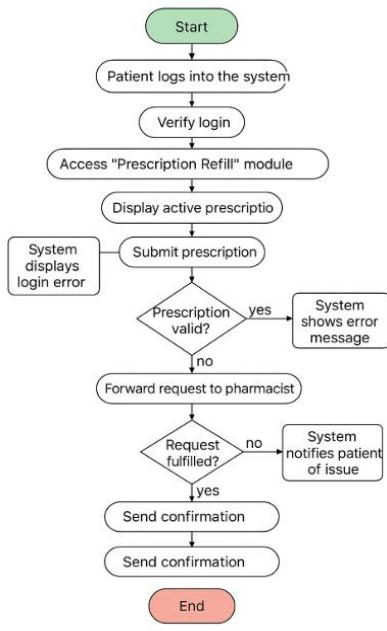
## Electronic Healthcare System Requirements Specification



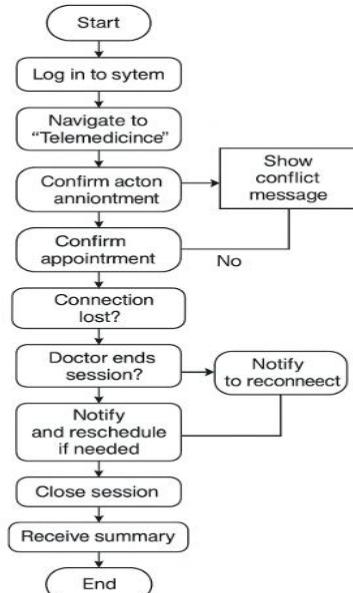
## Electronic Healthcare System Requirements Specification

**Liza Koliqi:**

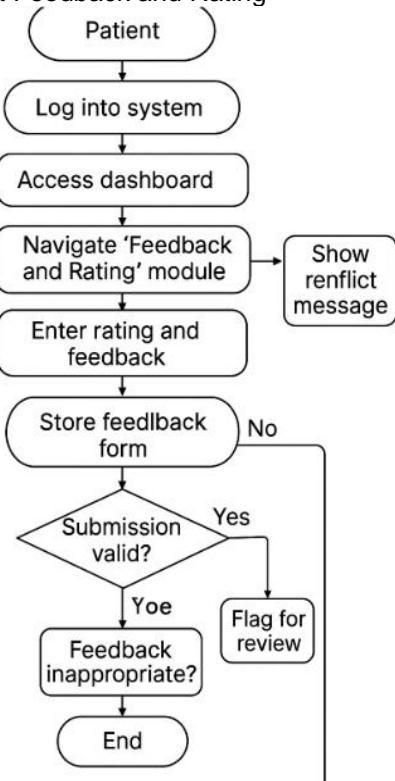
Prescription Refill Management



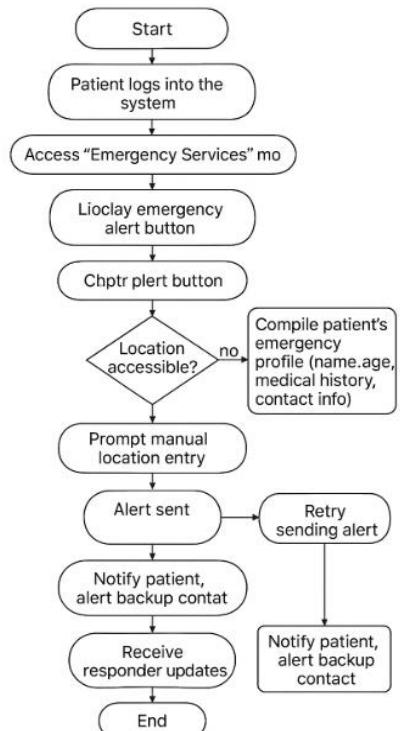
Telemedicine Consultation



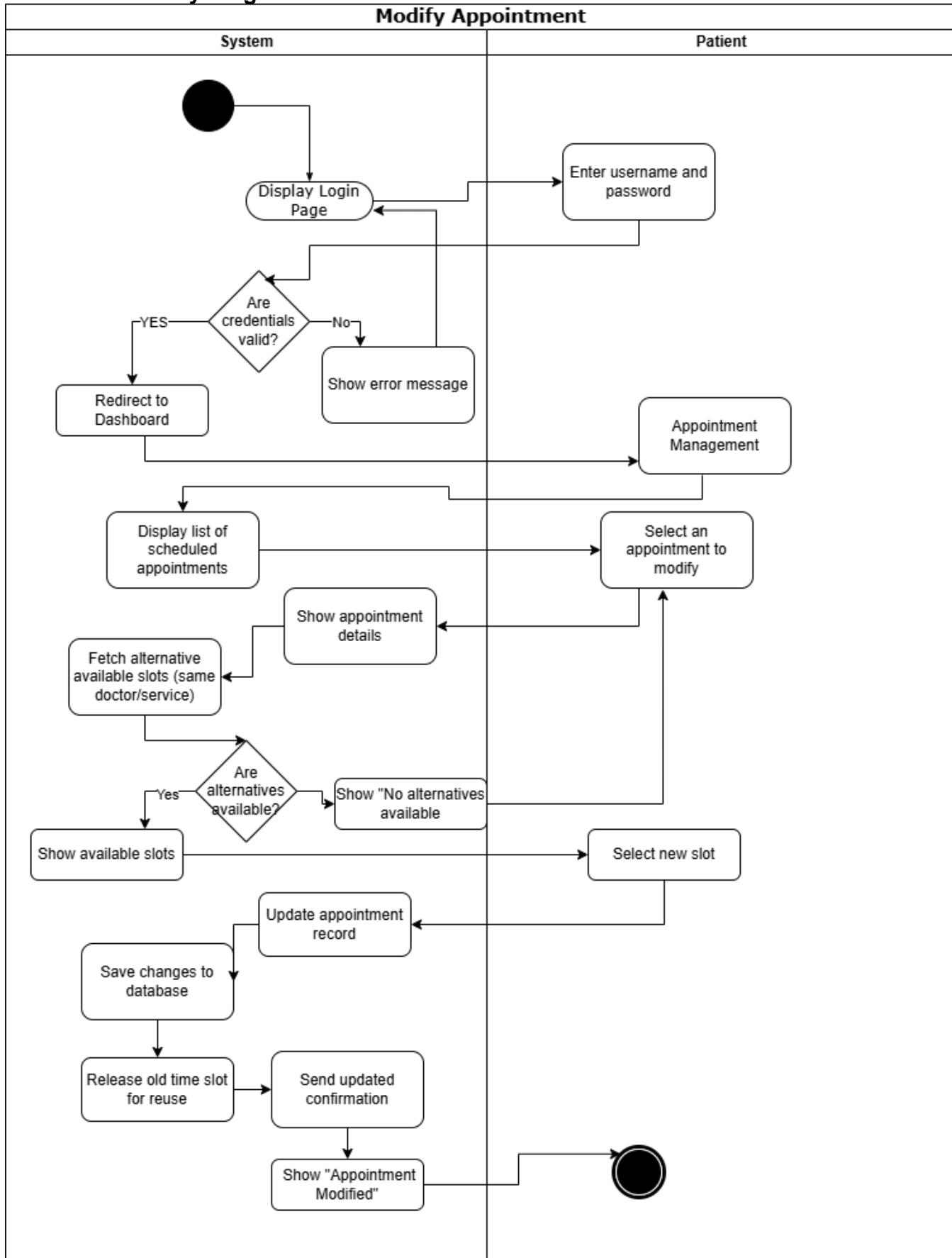
Patient Feedback and Rating



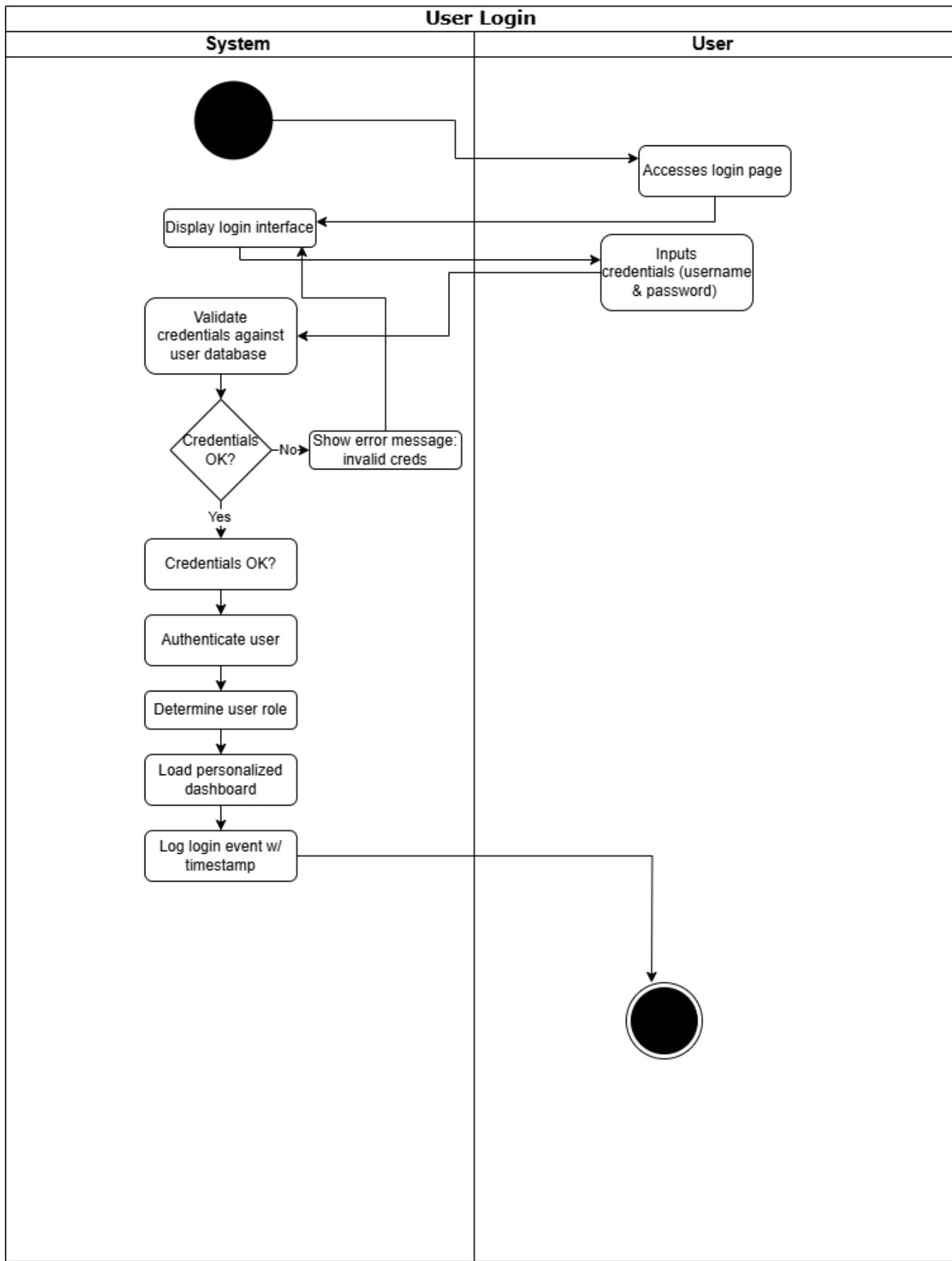
Emergency Service Access



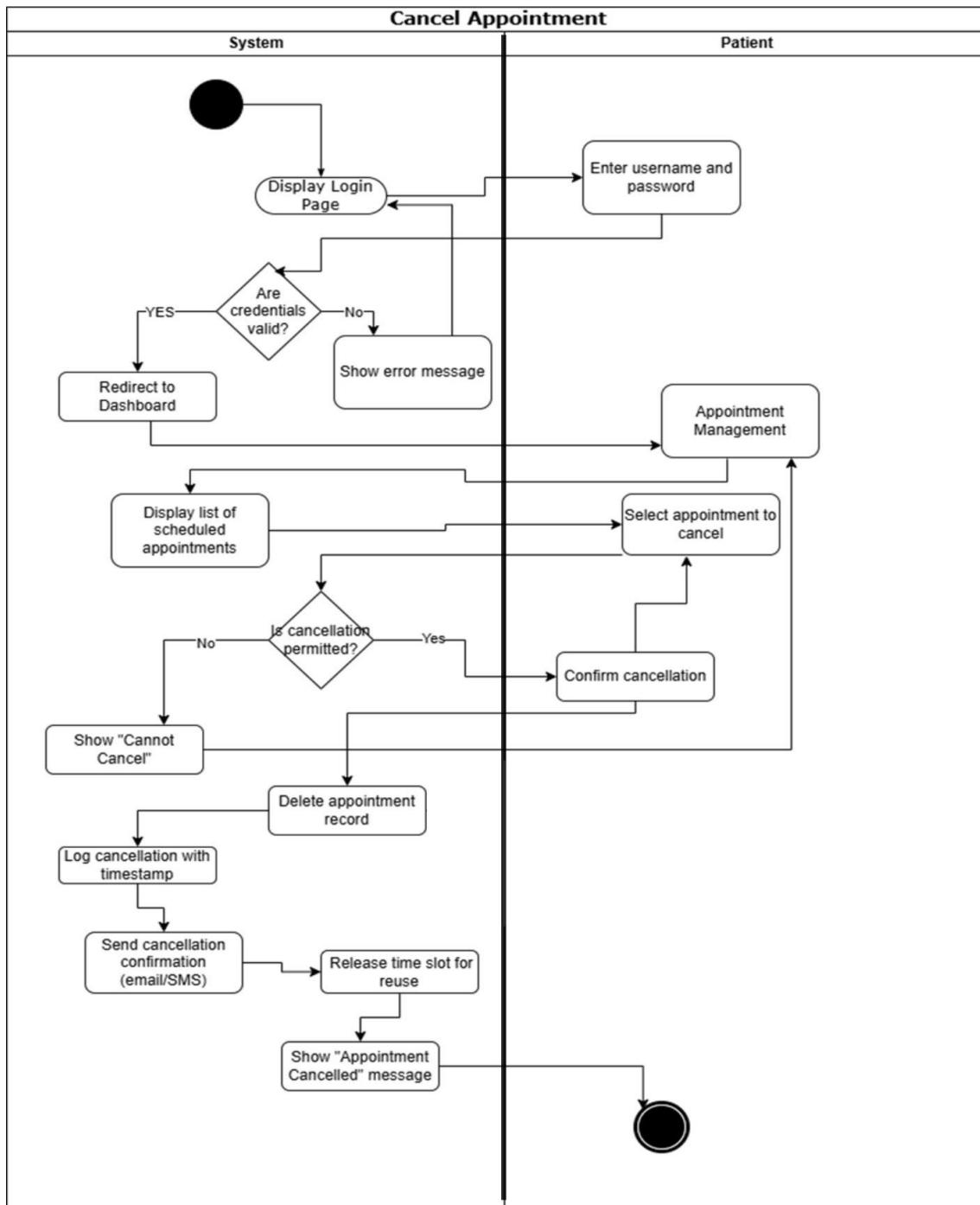
## Extension Activity Diagrams- Elisona Doku



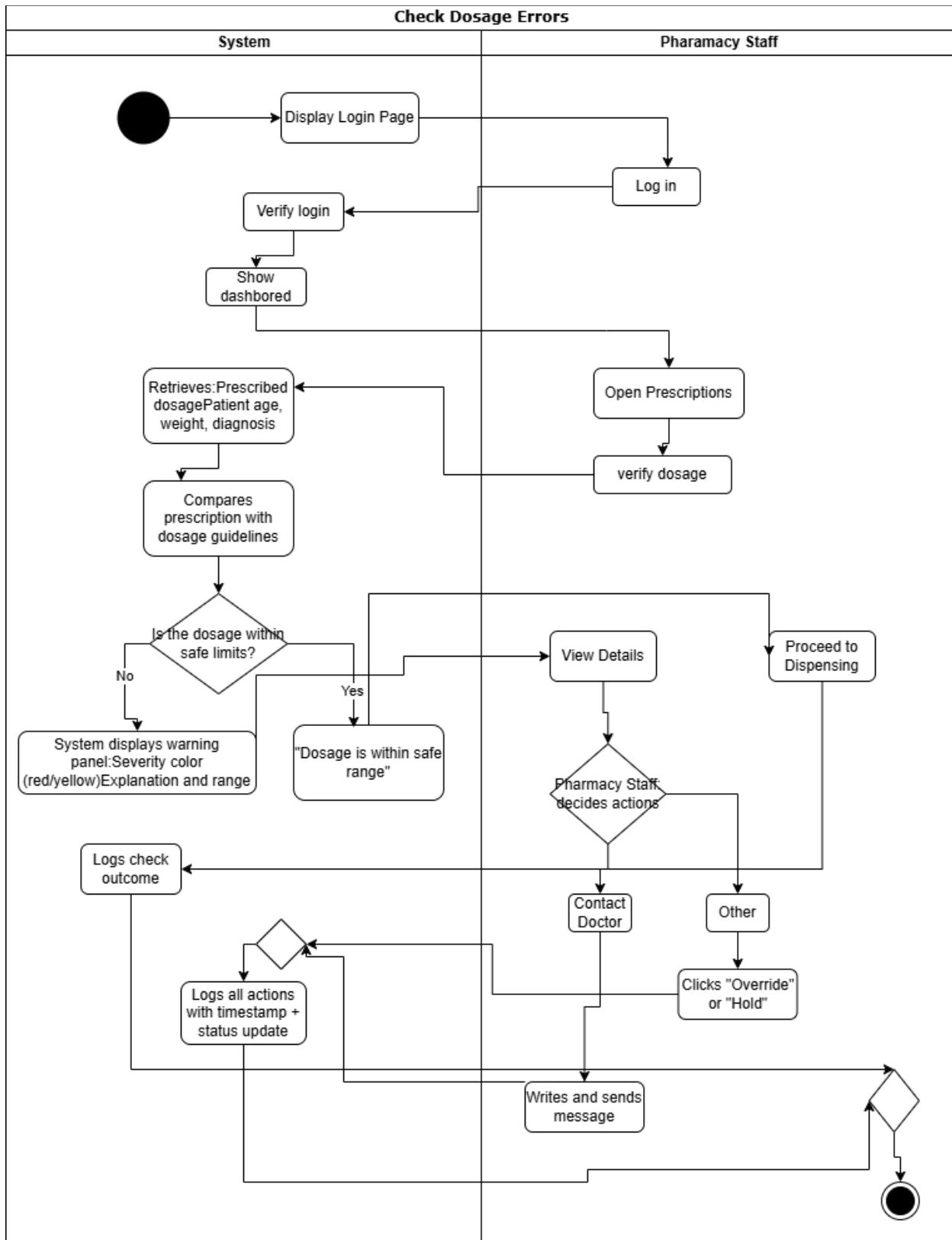
## Electronic Healthcare System Requirements Specification



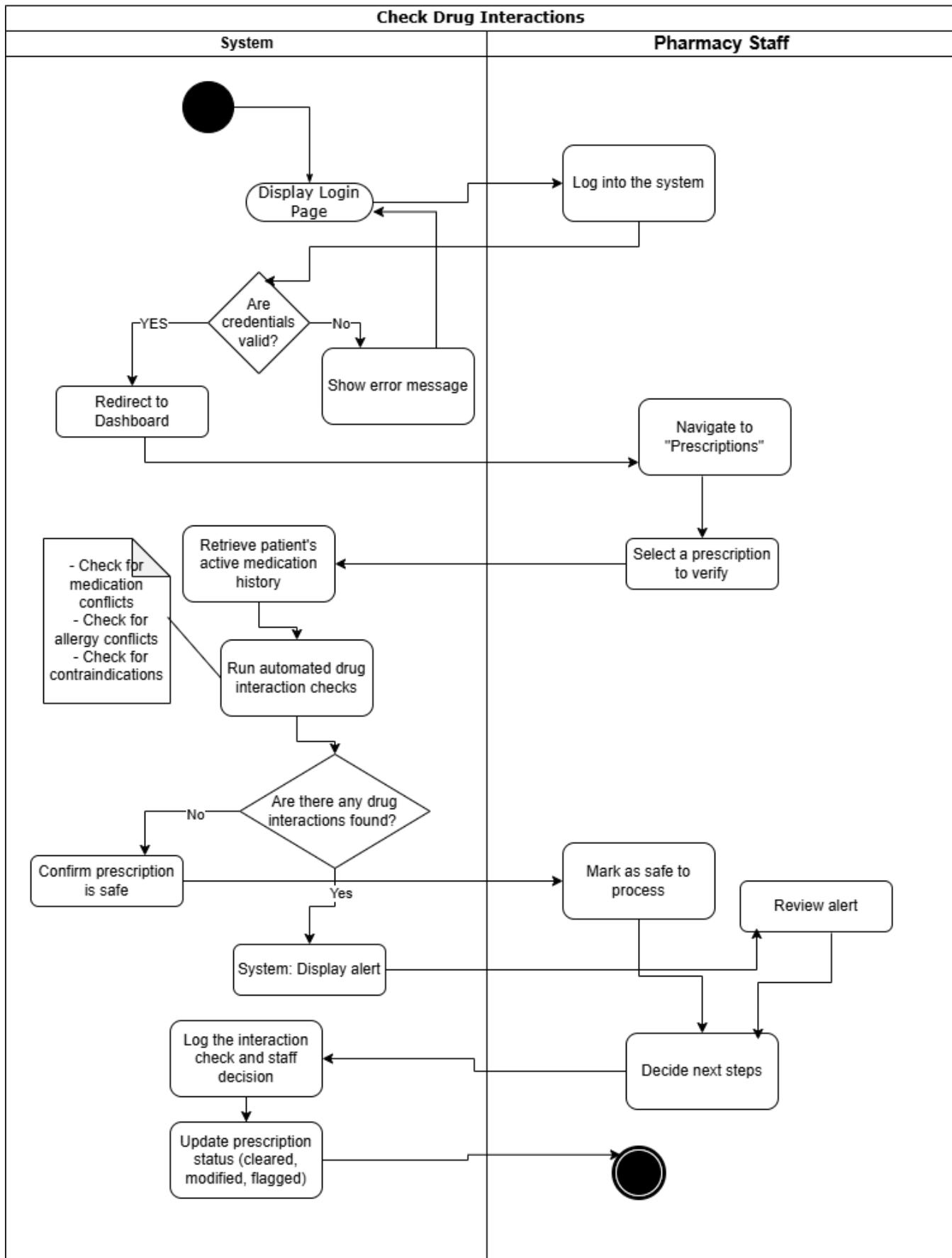
## Electronic Healthcare System Requirements Specification



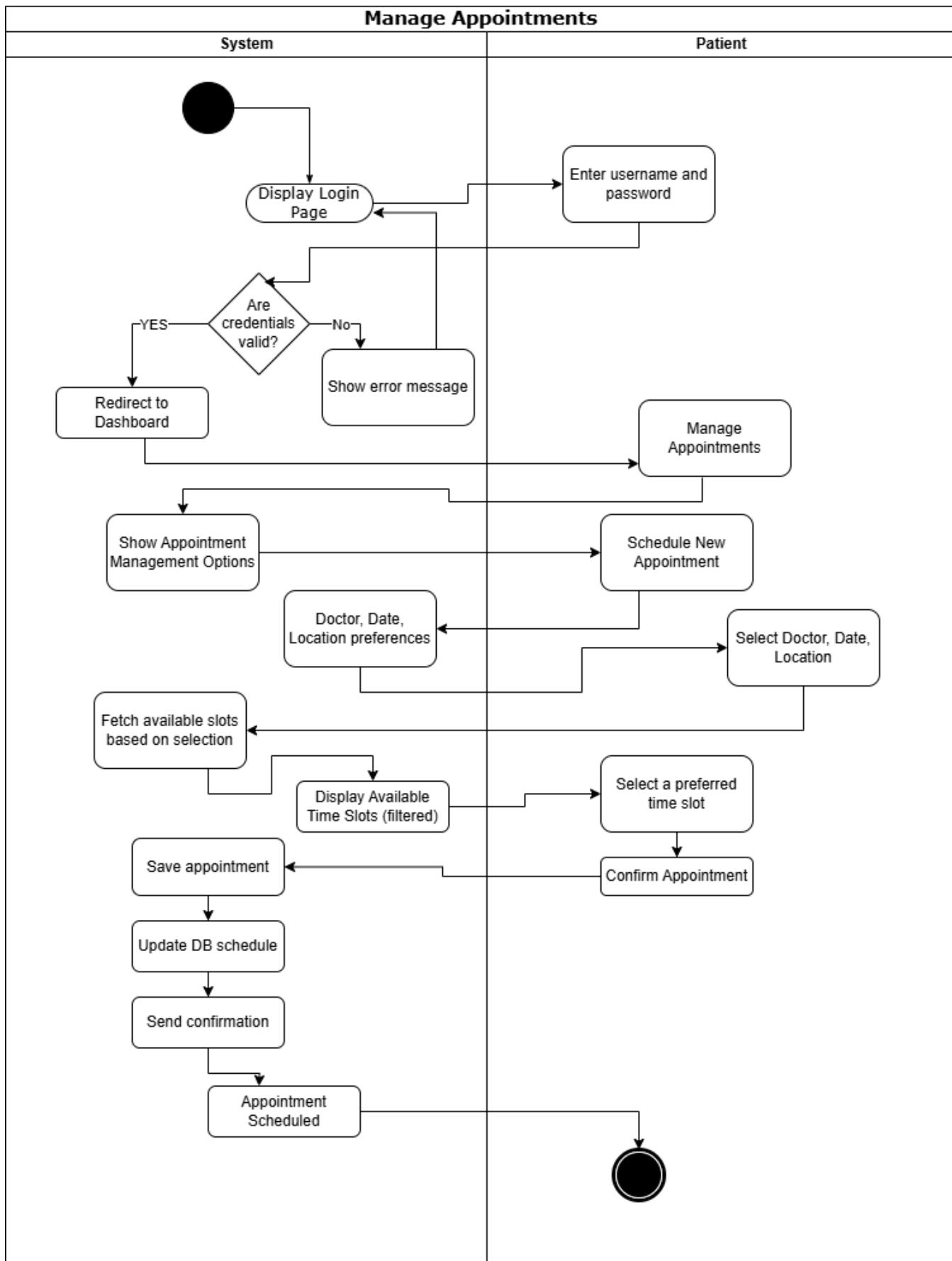
## Electronic Healthcare System Requirements Specification



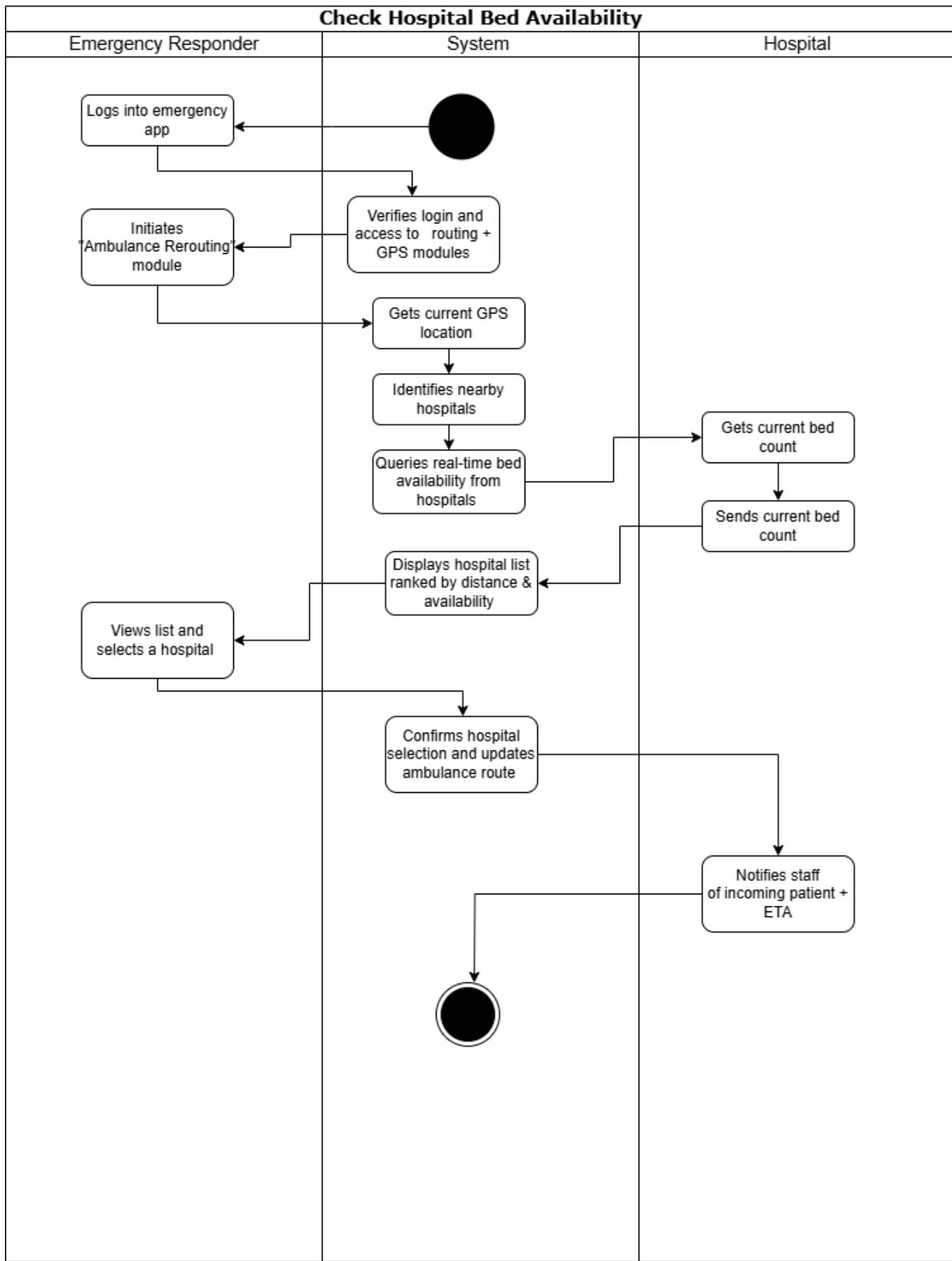
## Electronic Healthcare System Requirements Specification



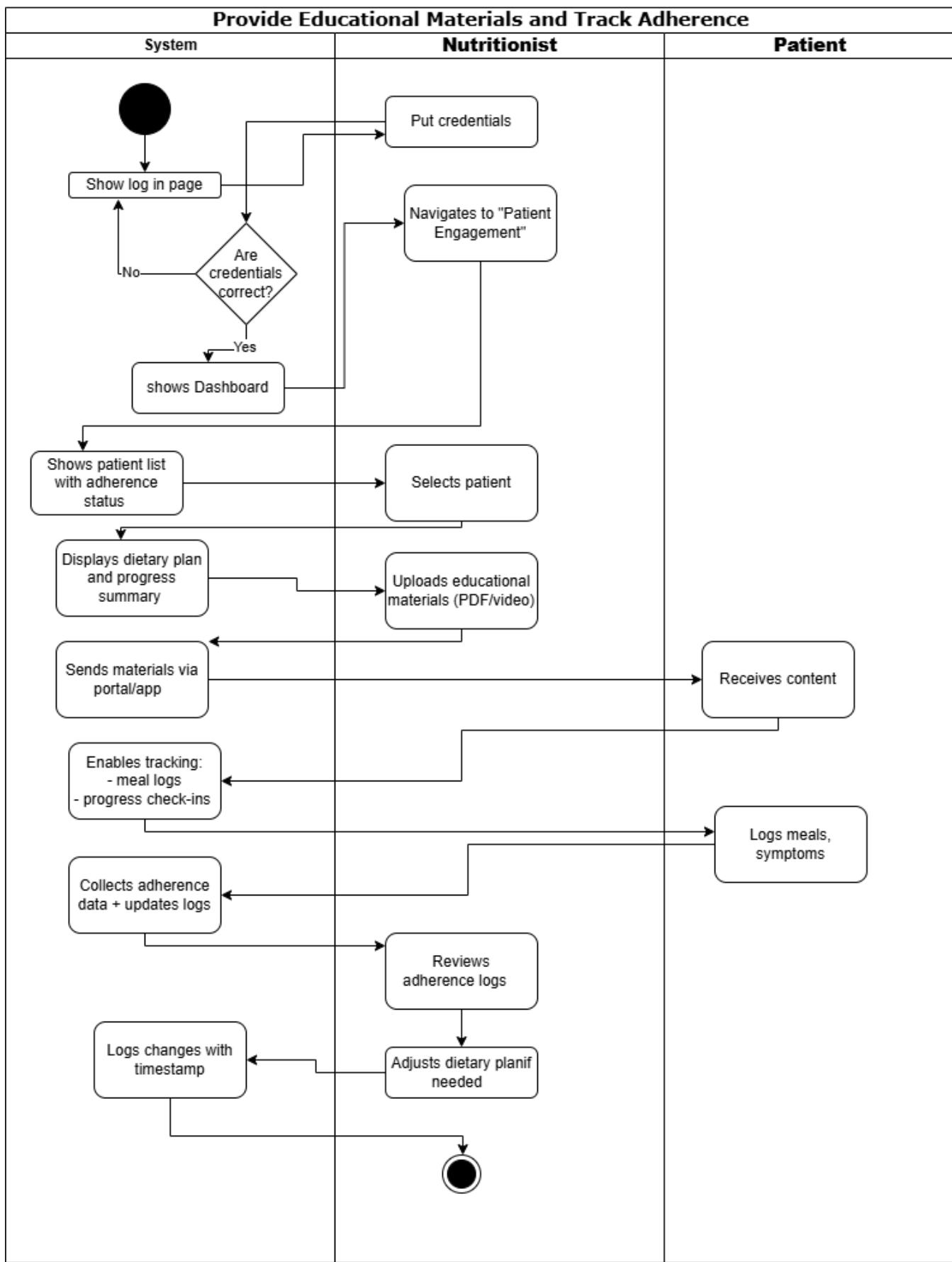
## Electronic Healthcare System Requirements Specification



## Electronic Healthcare System Requirements Specification

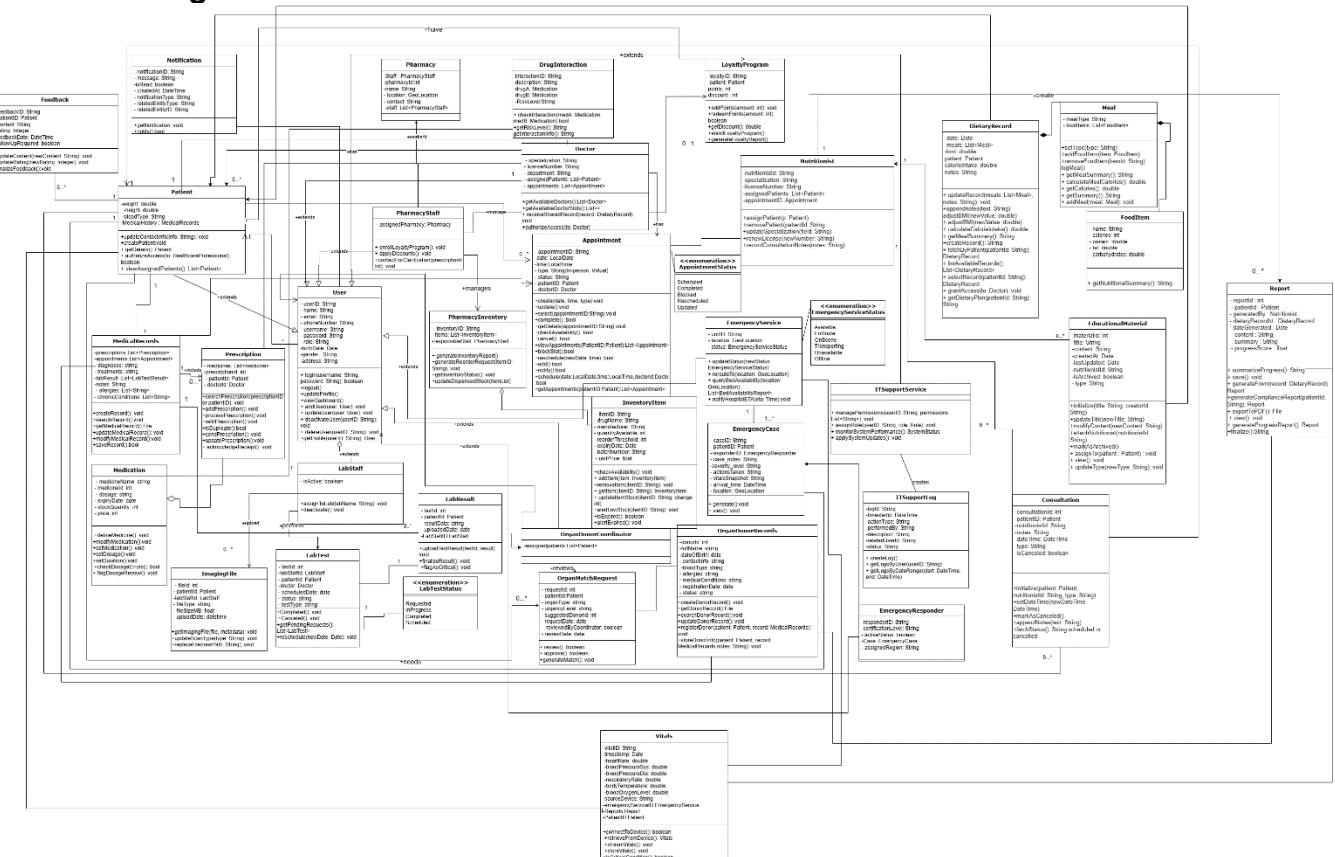


## Electronic Healthcare System Requirements Specification

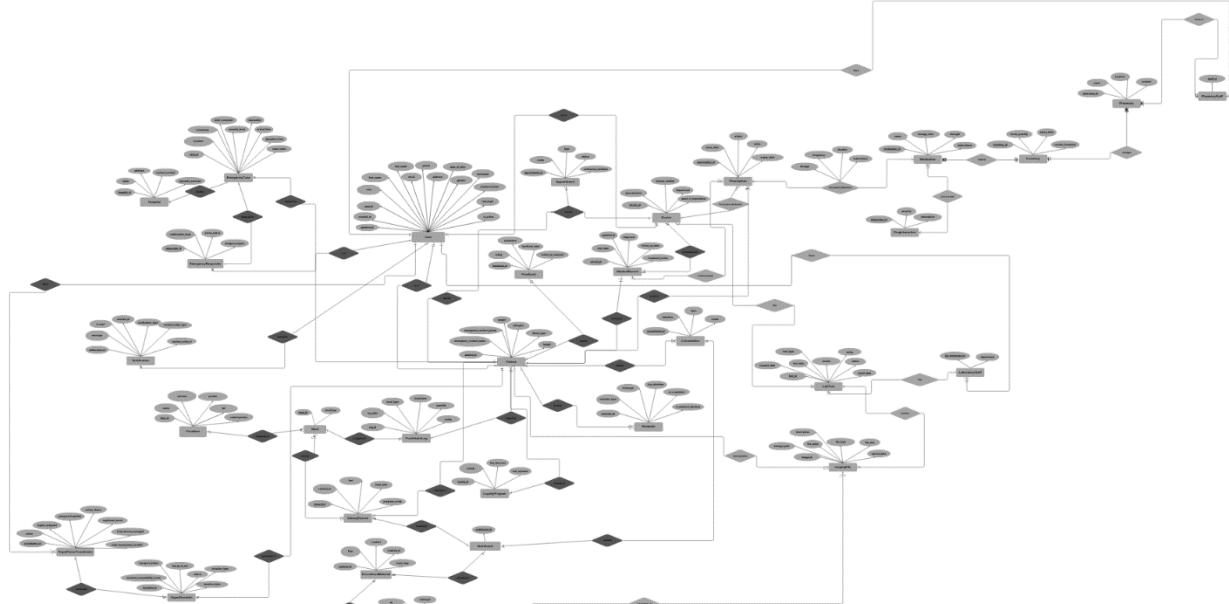


## Electronic Healthcare System Requirements Specification

### 5.3 Class Diagram -Flavia Koco & Antea Koxherri



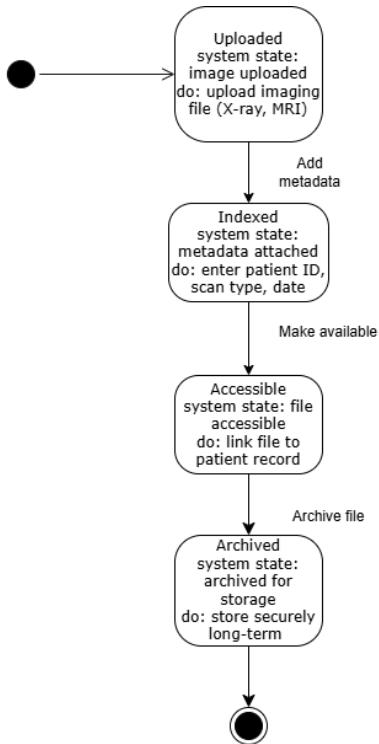
### 5.4 ERD- Belina Durmishi & Elisona Doku



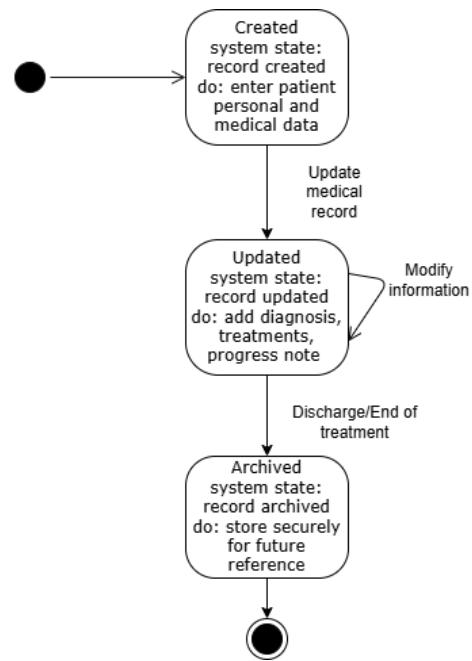
## 5.5 State Diagrams

-Evelina Gace:

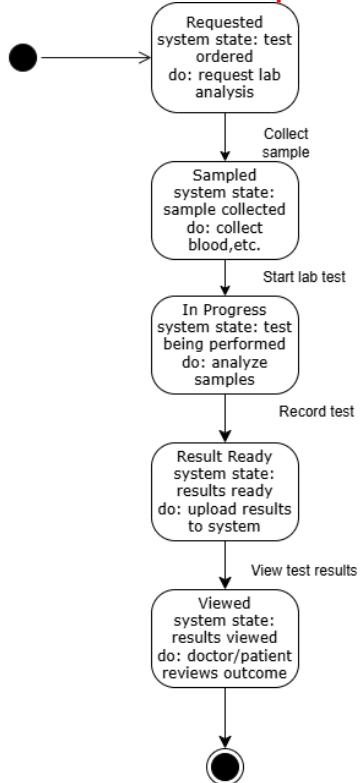
### Imaging File Upload:



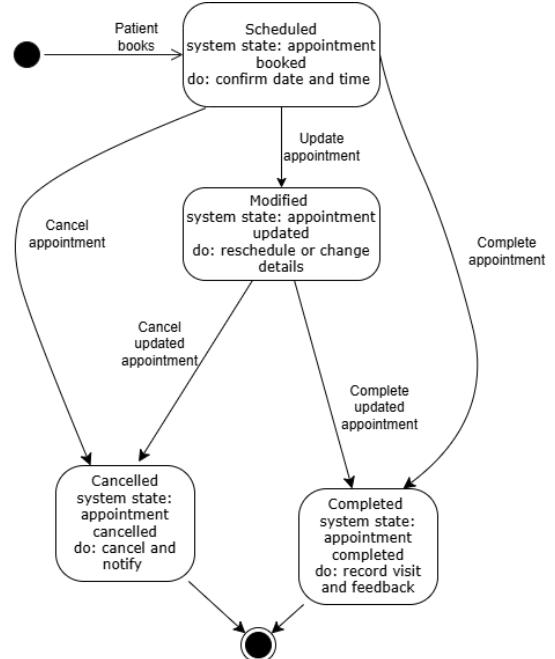
### Medical Record:



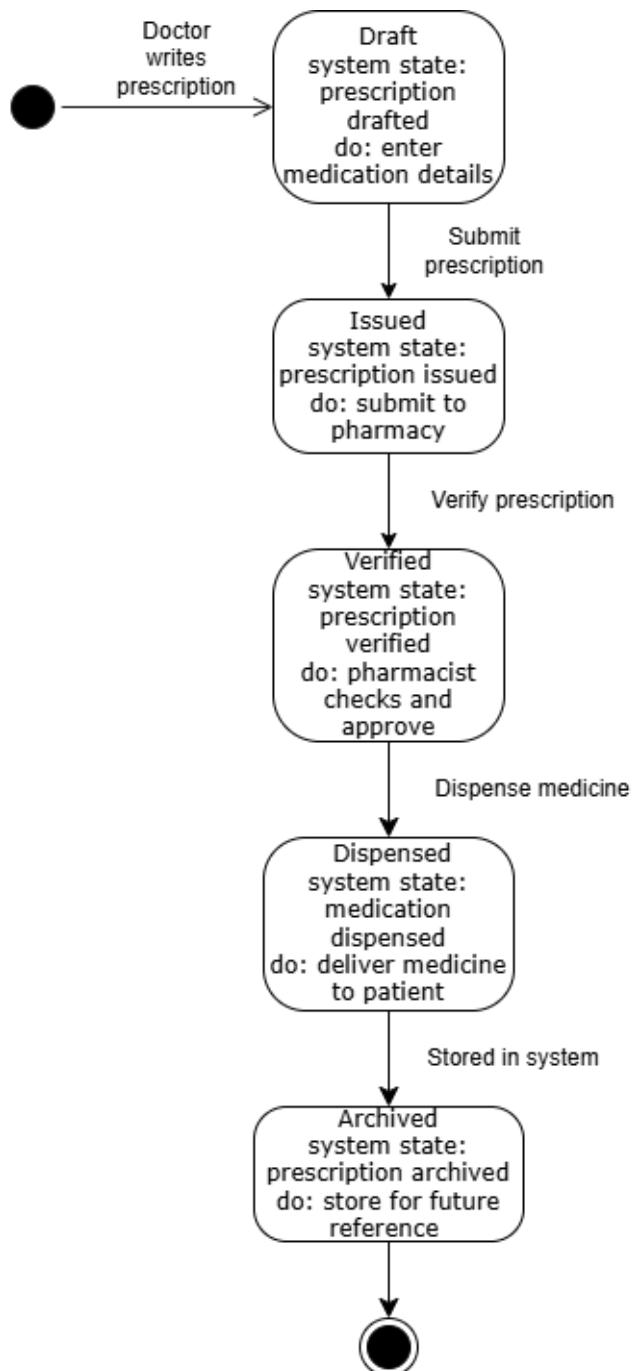
### Lab Test Request:



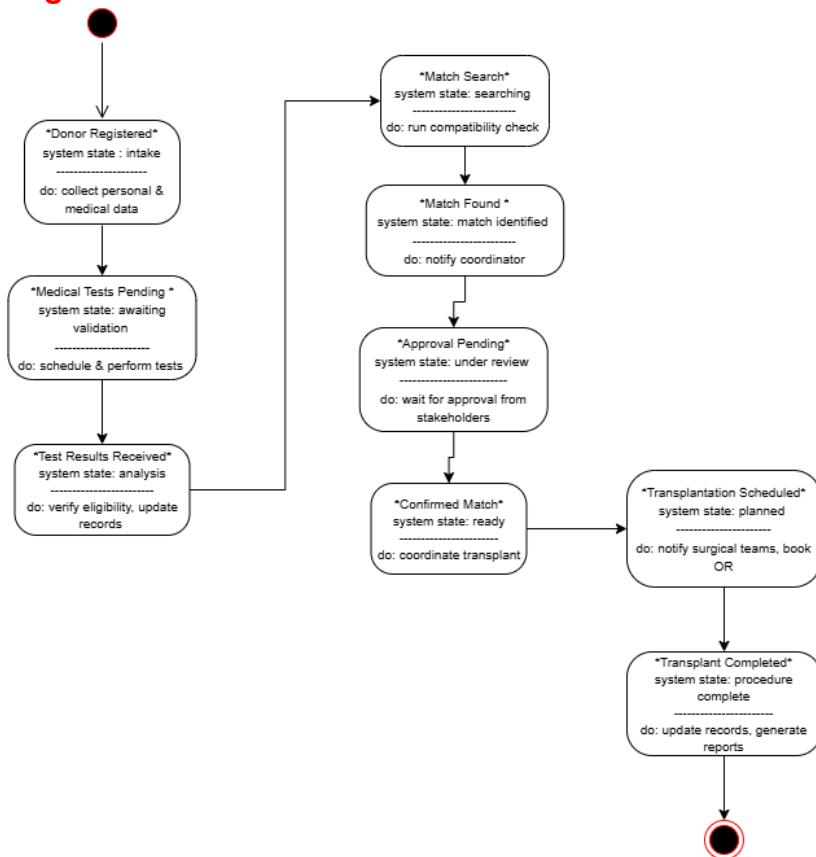
### Appointment:



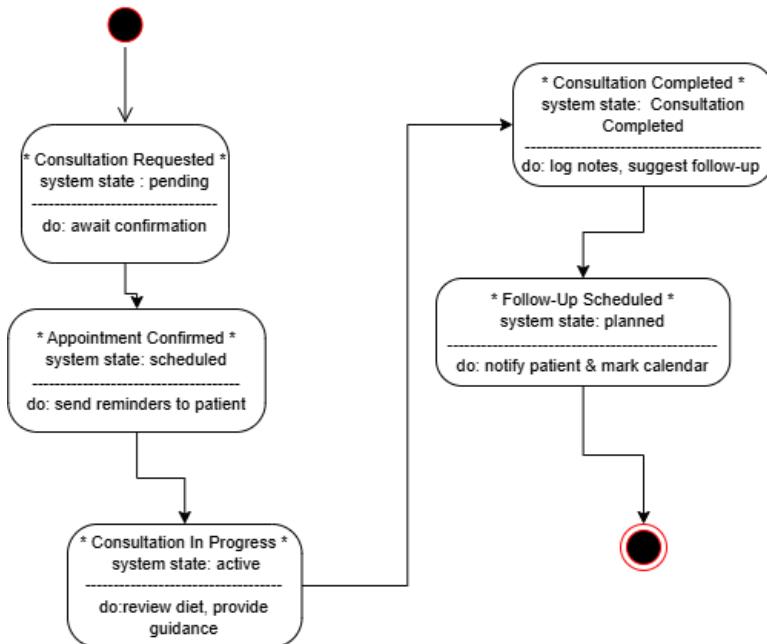
**Prescription:**



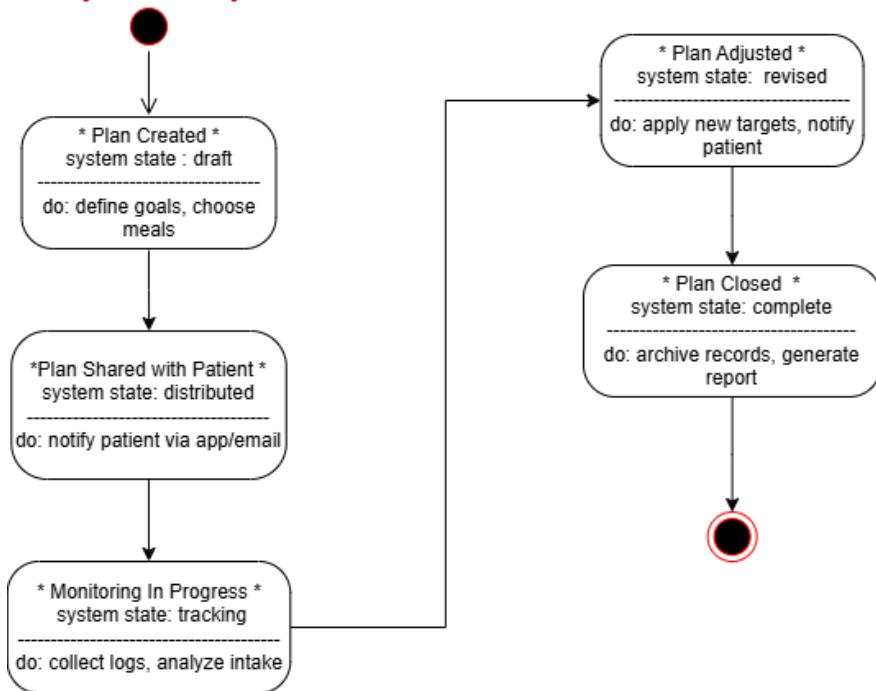
**Elkier Ago:**  
**Organ Donation Match Process**



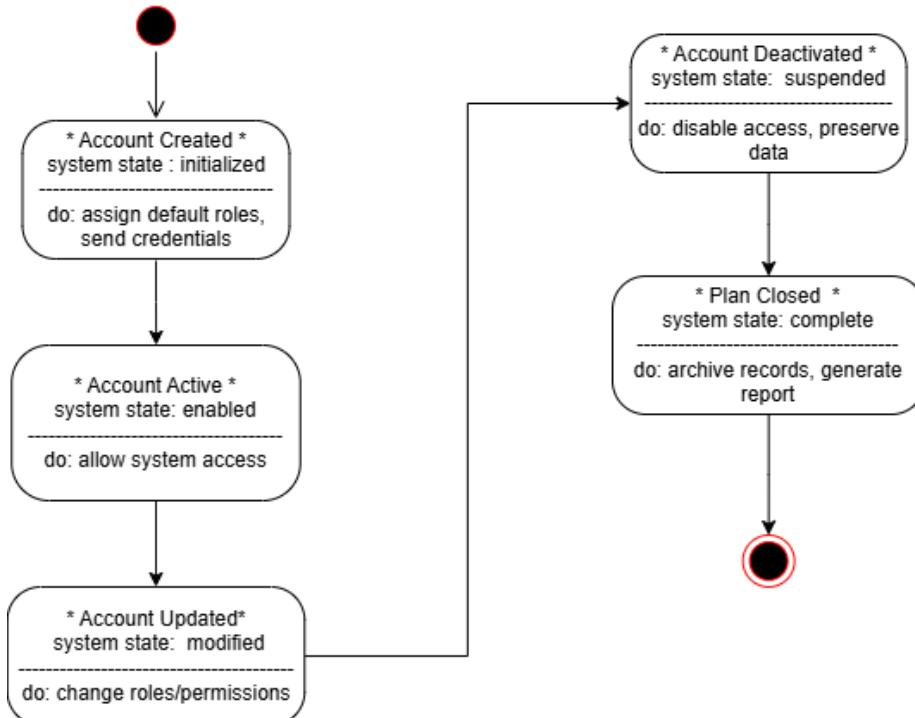
**Nutritionist Consultation Process**



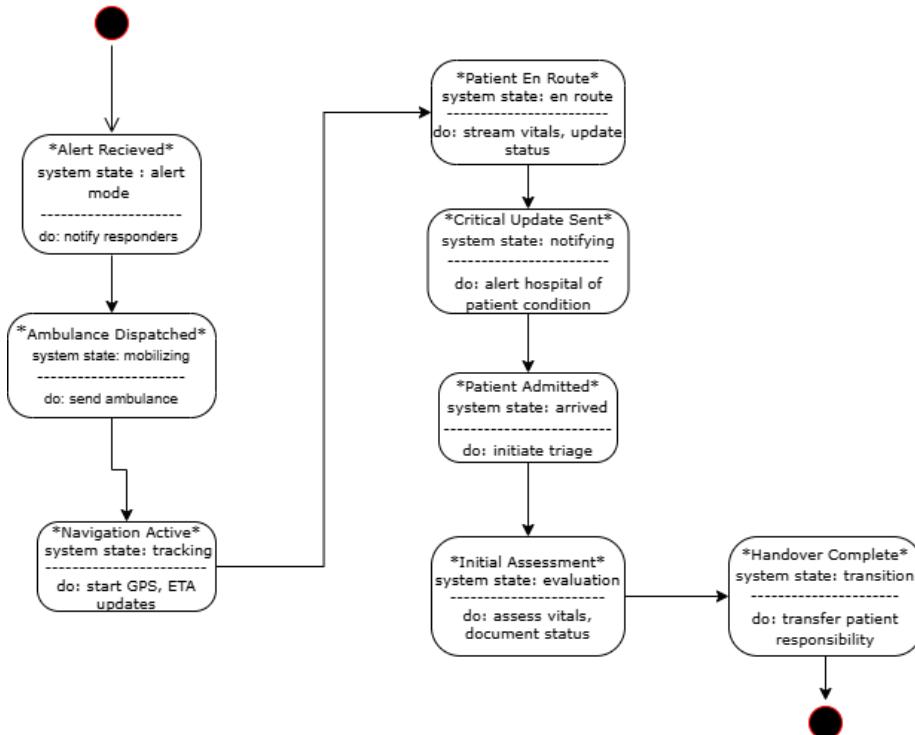
### Dietary Plan Lifecycle



### IT User Account Management



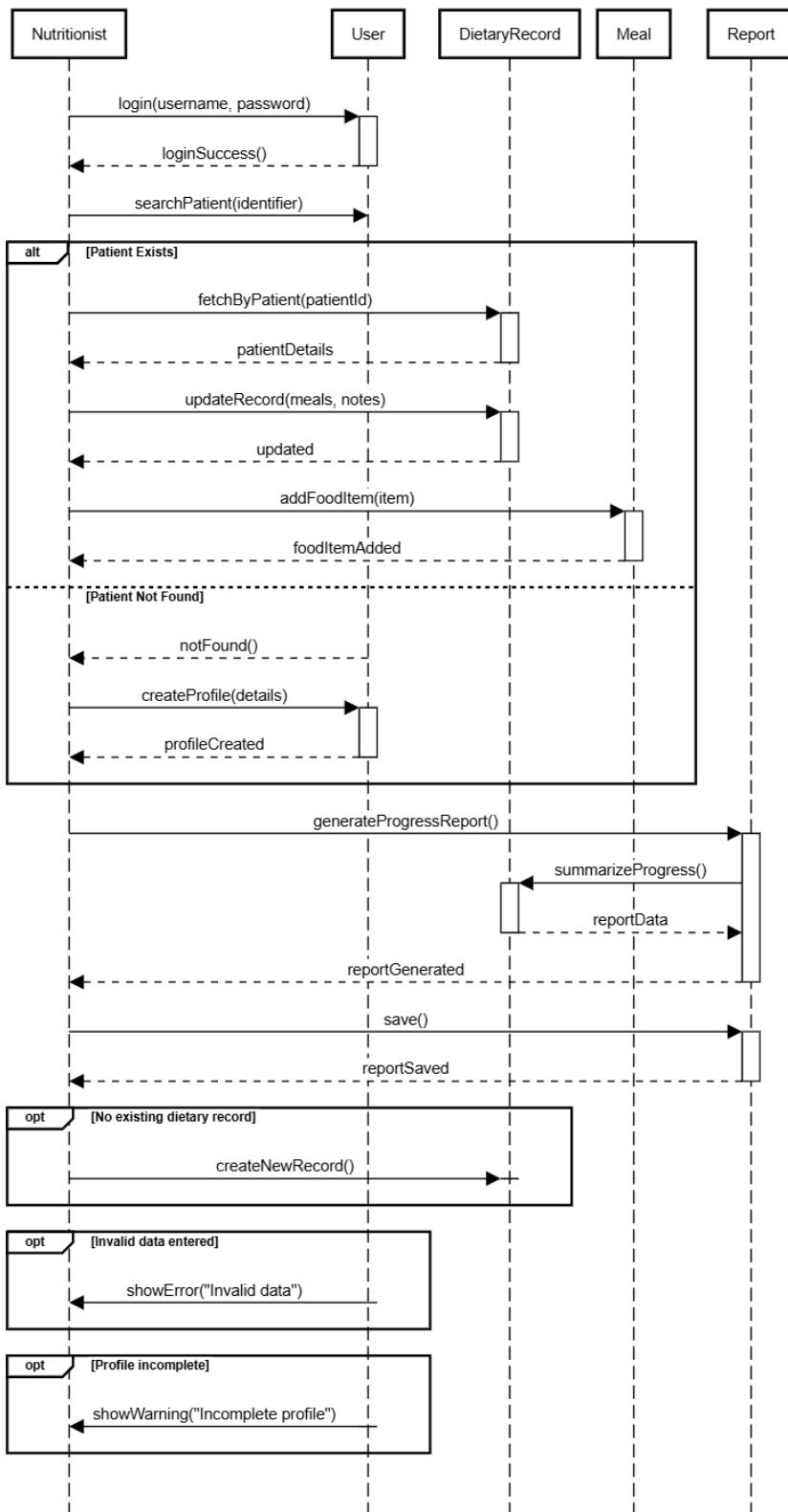
**Emergency Case Handling**



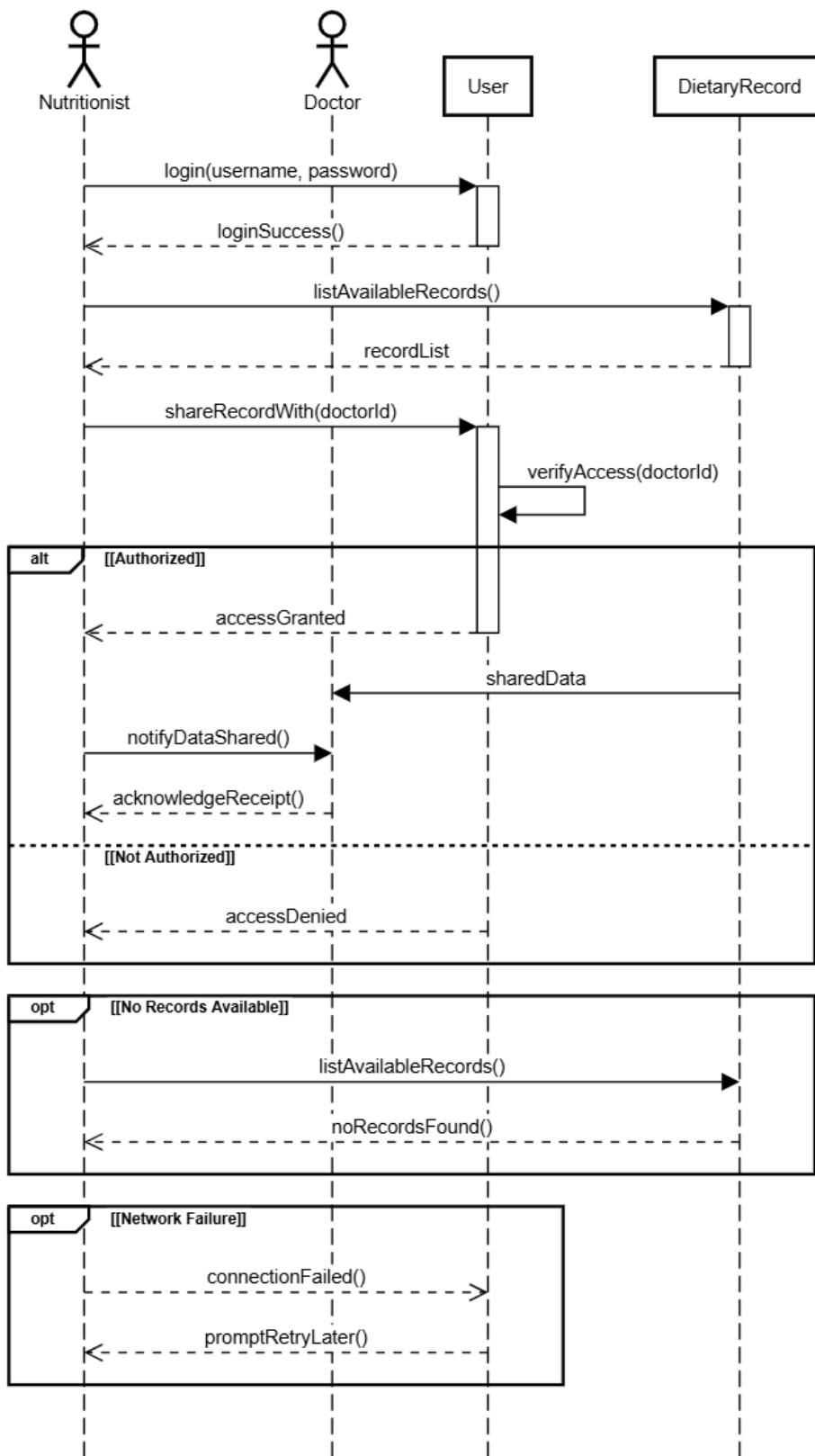
## 5.6 Sequence Diagrams:

### Nutritionist-Belina Durmishi

UC\_NUT\_01: Dietary Record Management & Monitoring

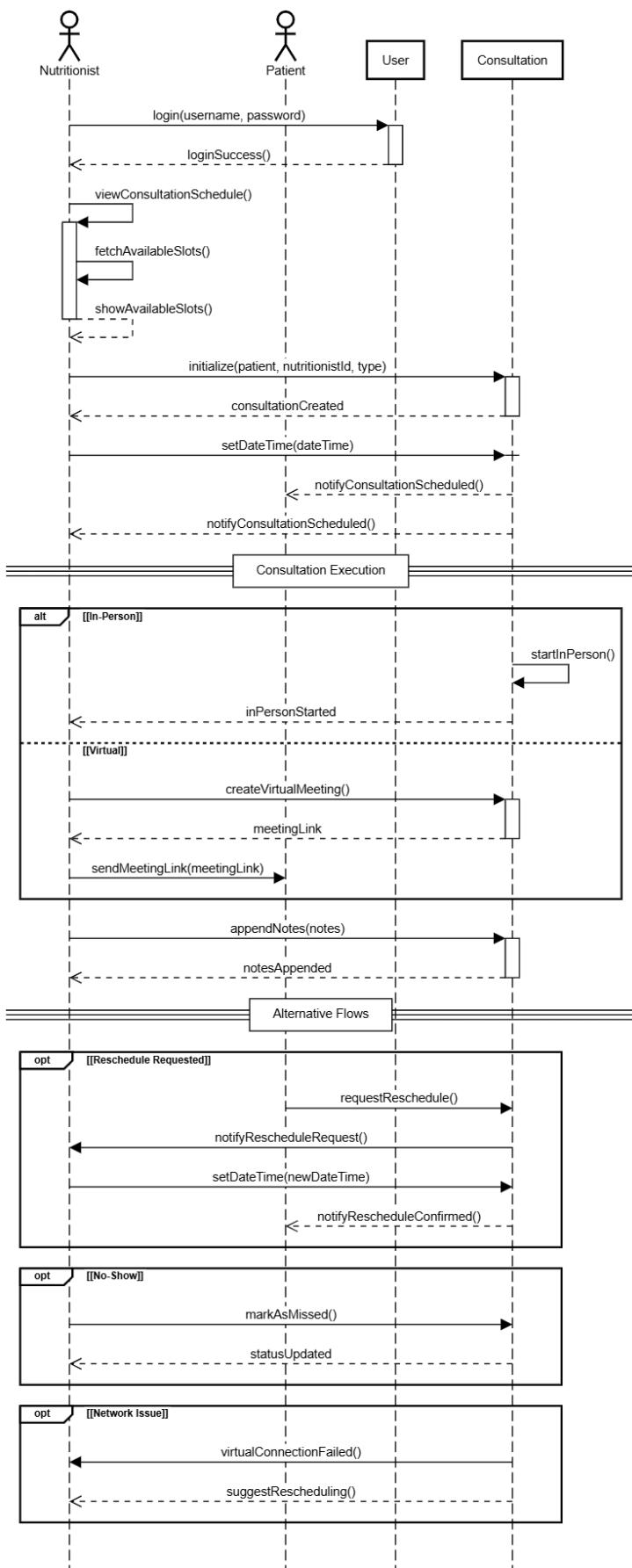


**UC\_NUT\_02: Healthcare Collaboration**



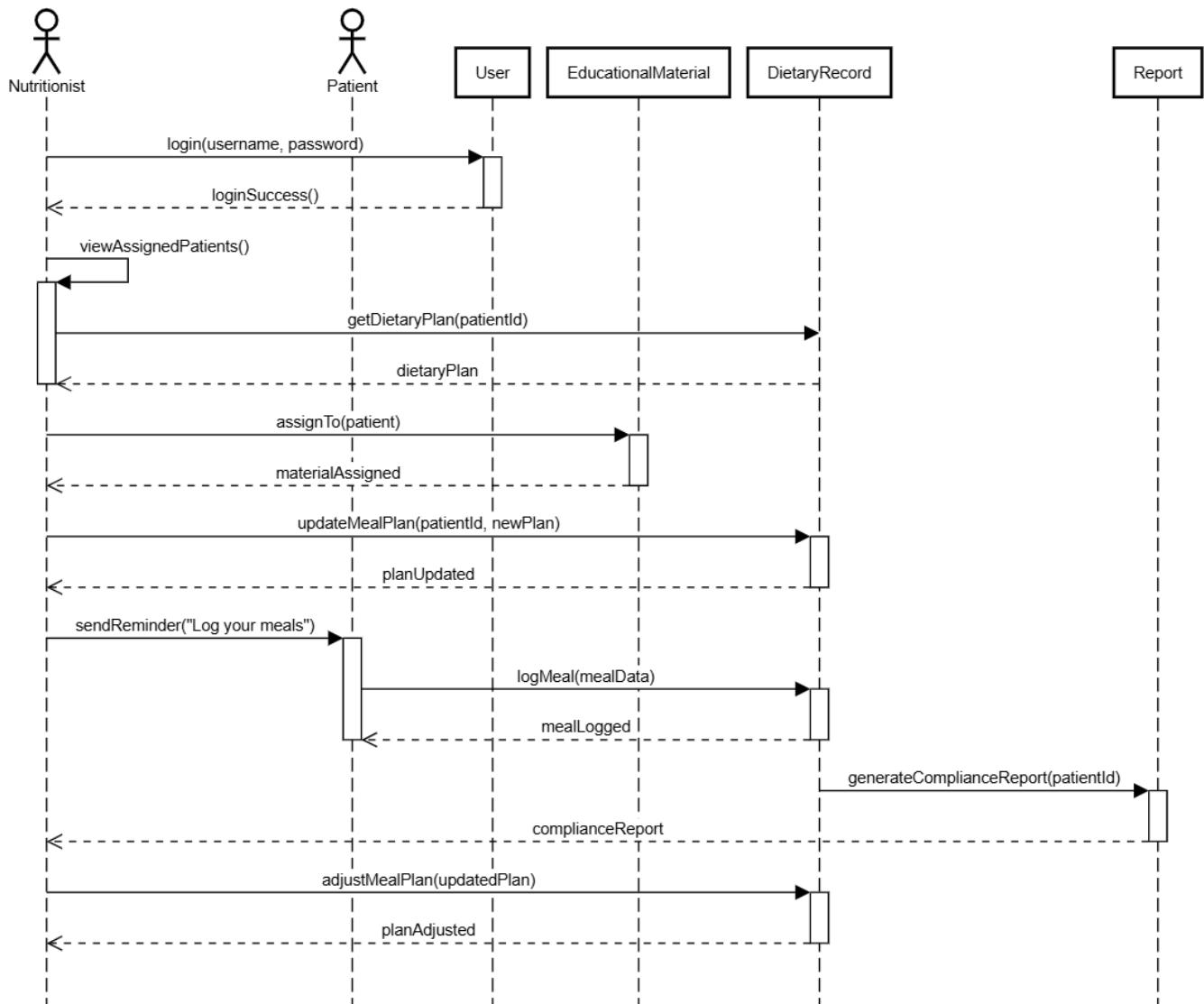
## Electronic Healthcare System Requirements Specification

UC\_NUT\_03: Consultation Management



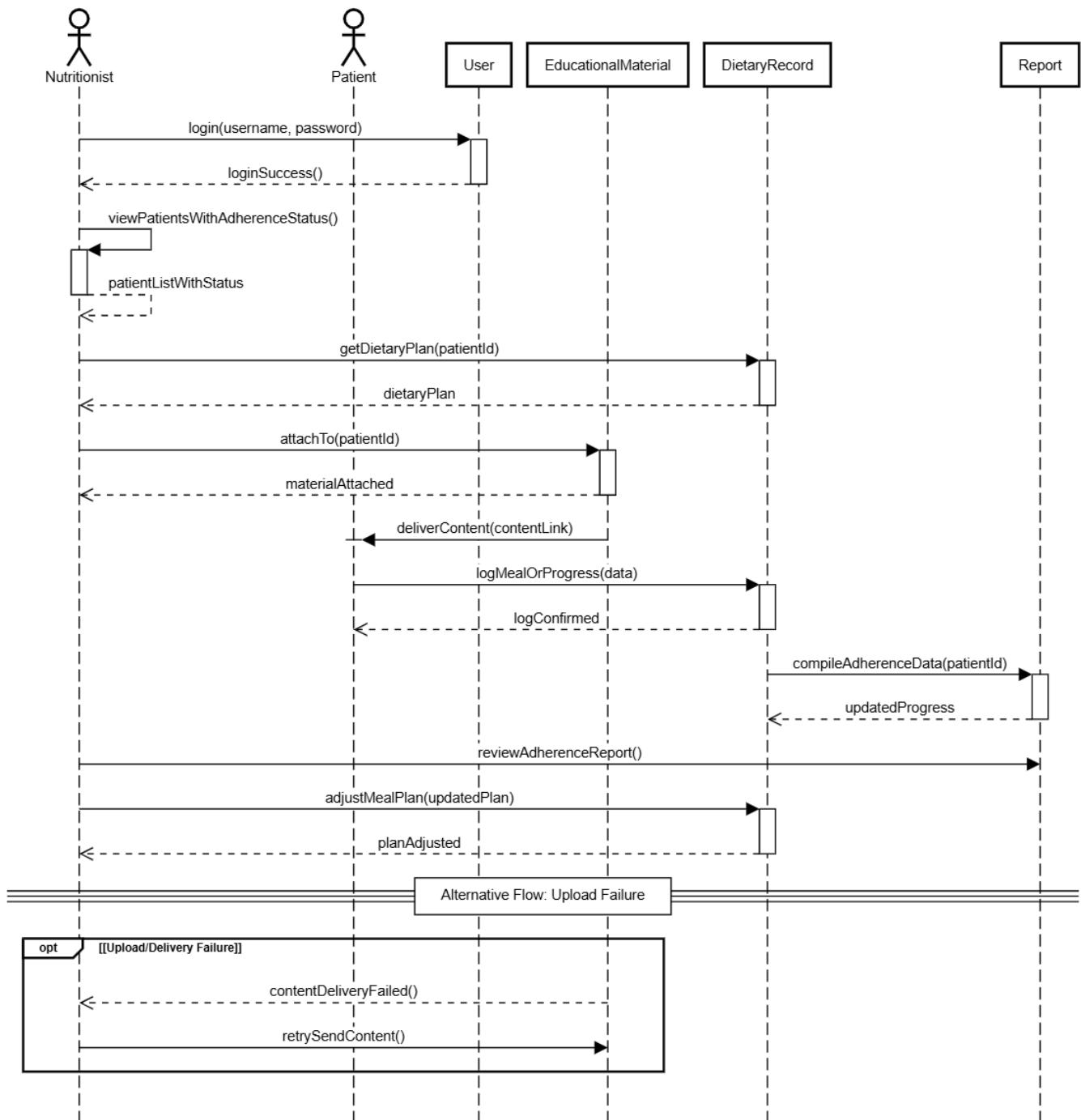
## Electronic Healthcare System Requirements Specification

### UC\_NUT\_04: Patient Engagement & Compliance

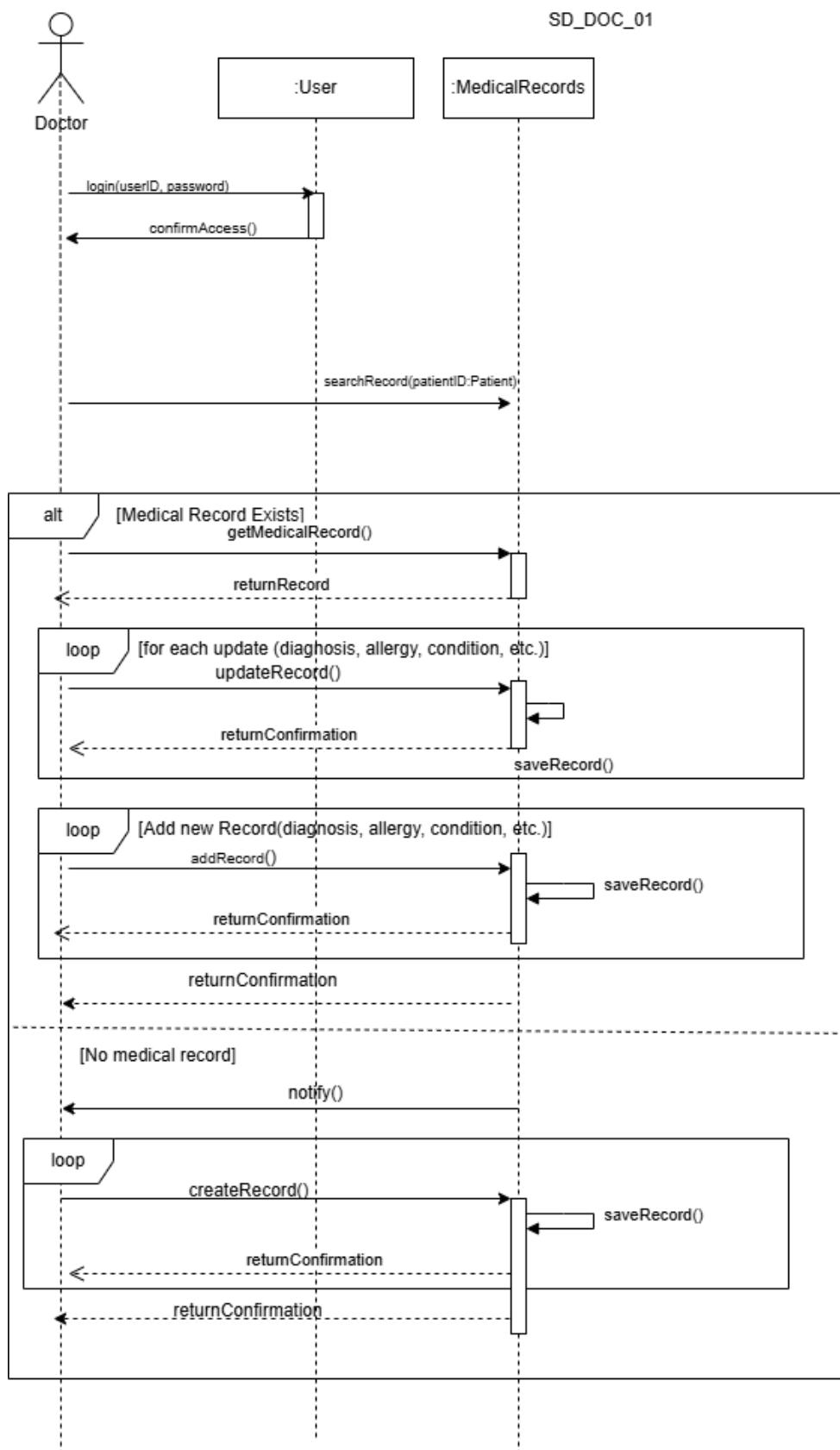


## Electronic Healthcare System Requirements Specification

UC\_NUT\_05: Provide Educational Materials and Track Adherence

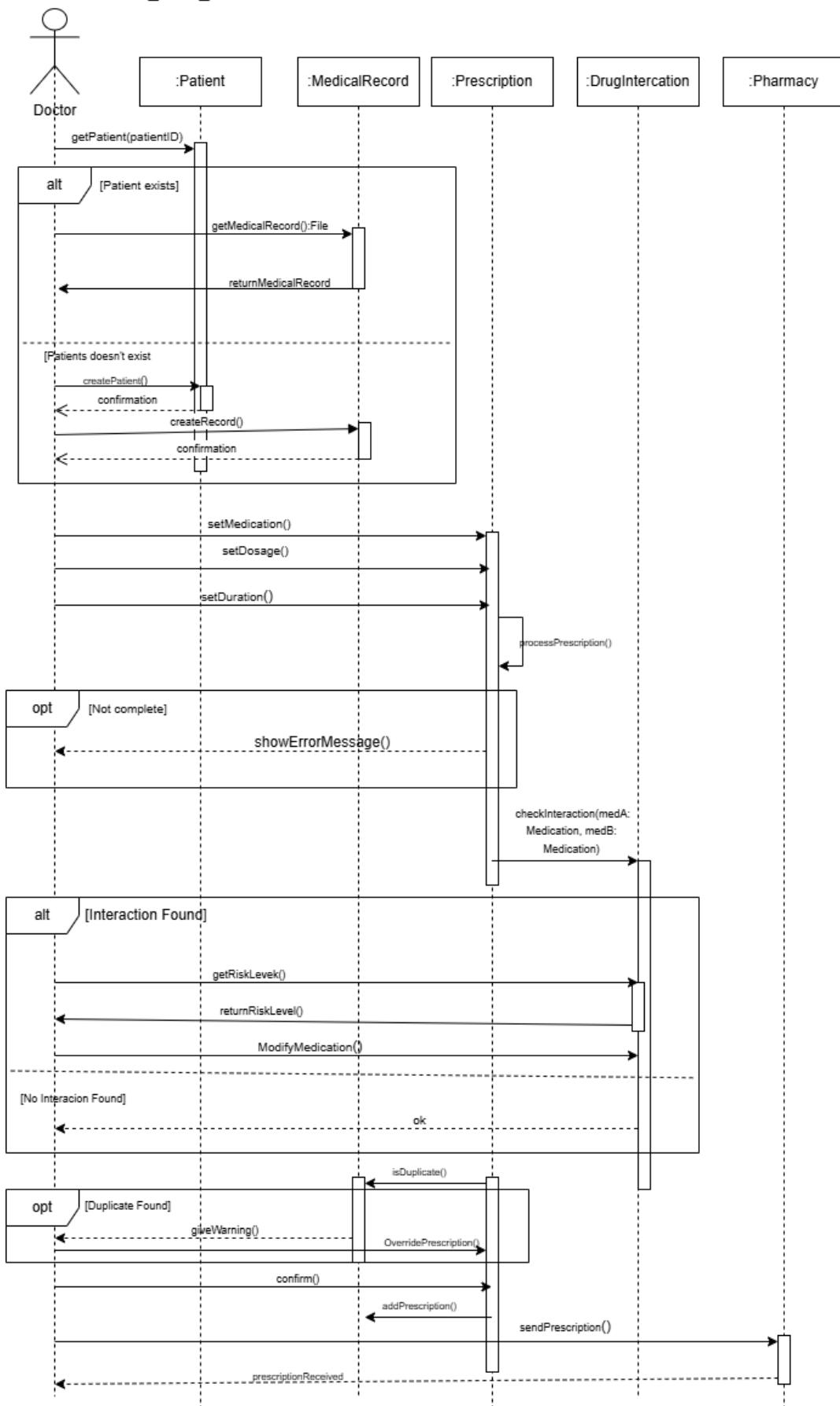


**Antea Koxherri:**



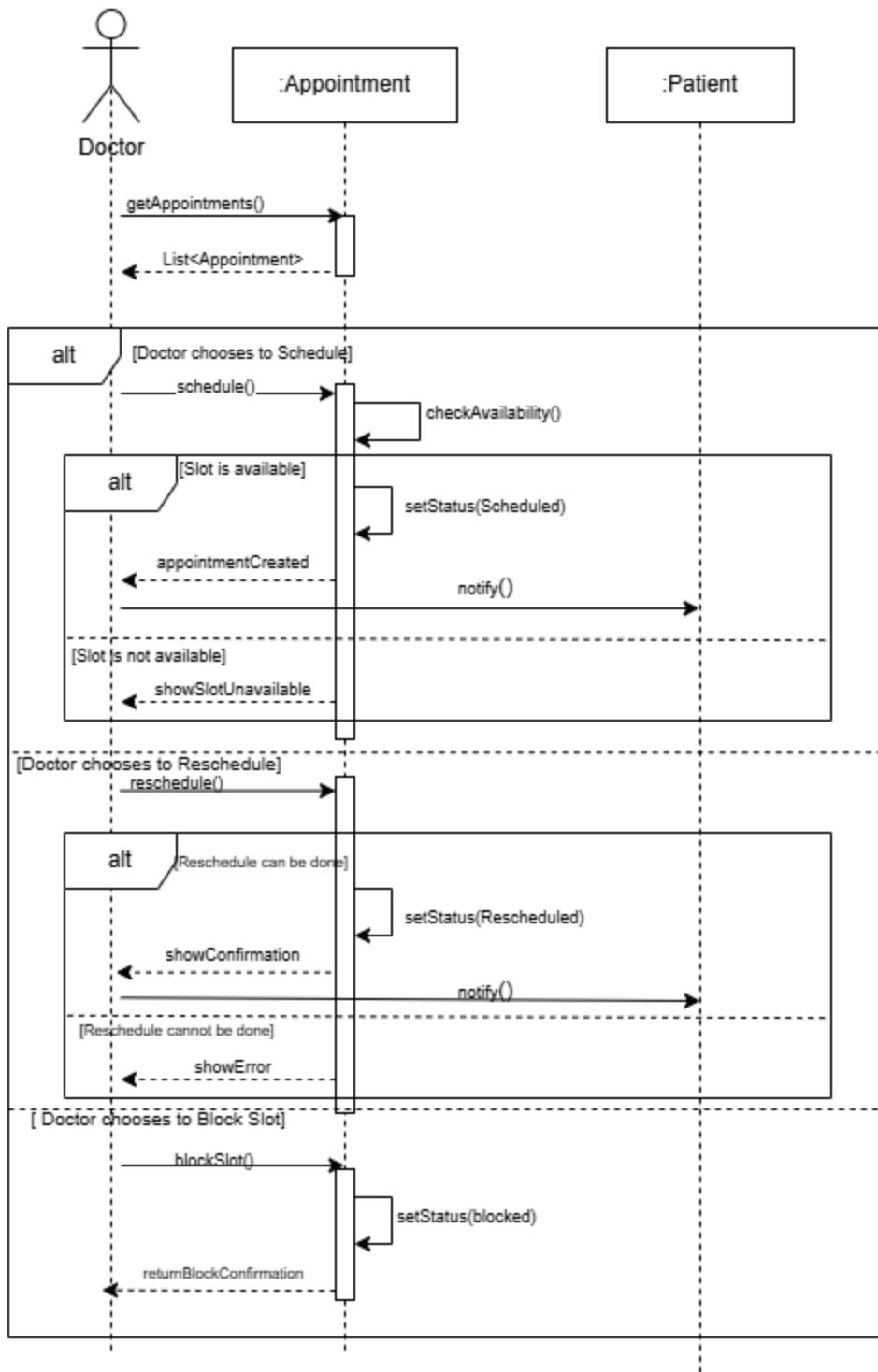
## Electronic Healthcare System Requirements Specification

SD\_DOC\_02

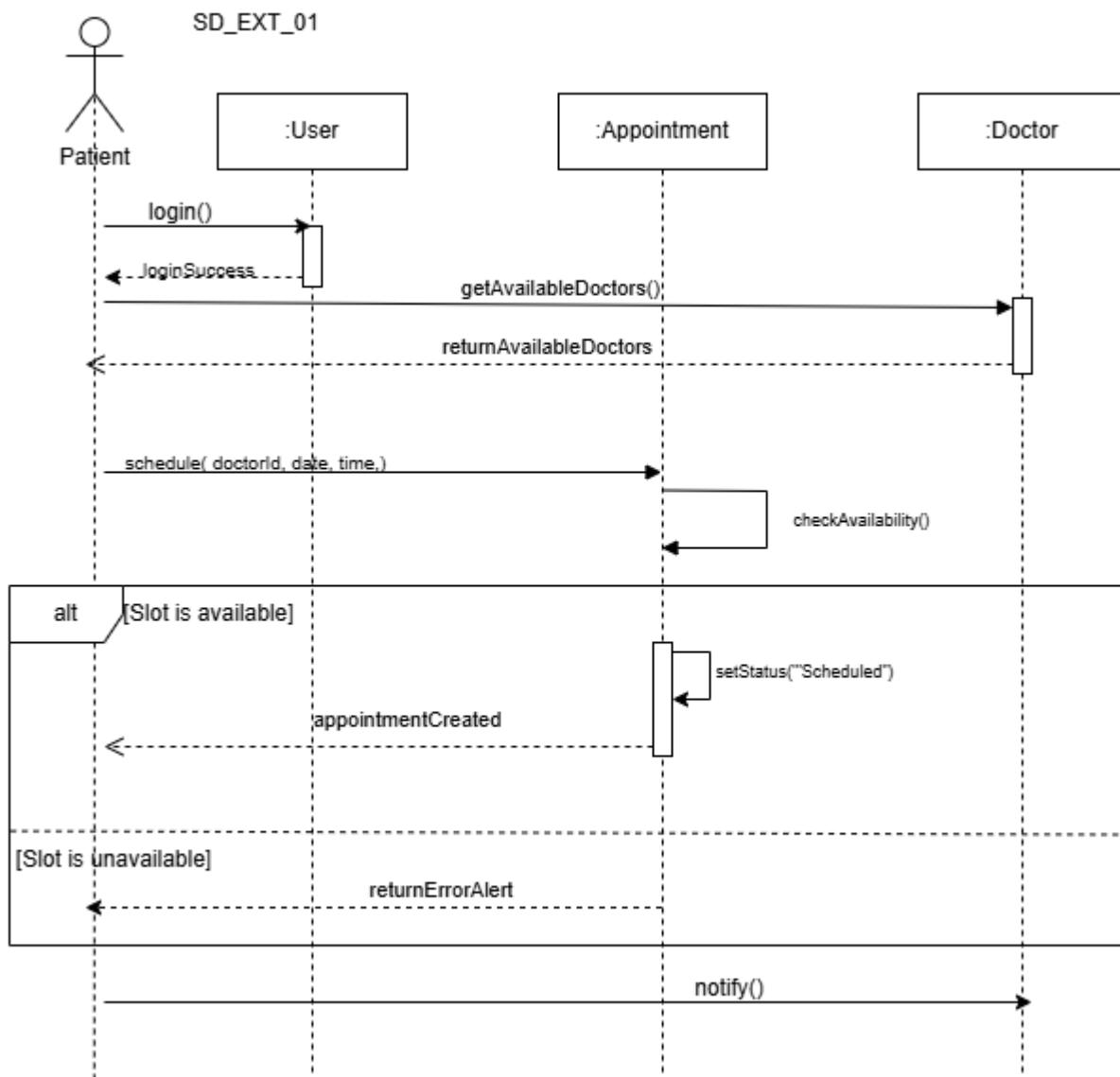


## Electronic Healthcare System Requirements Specification

SD\_DOC\_03

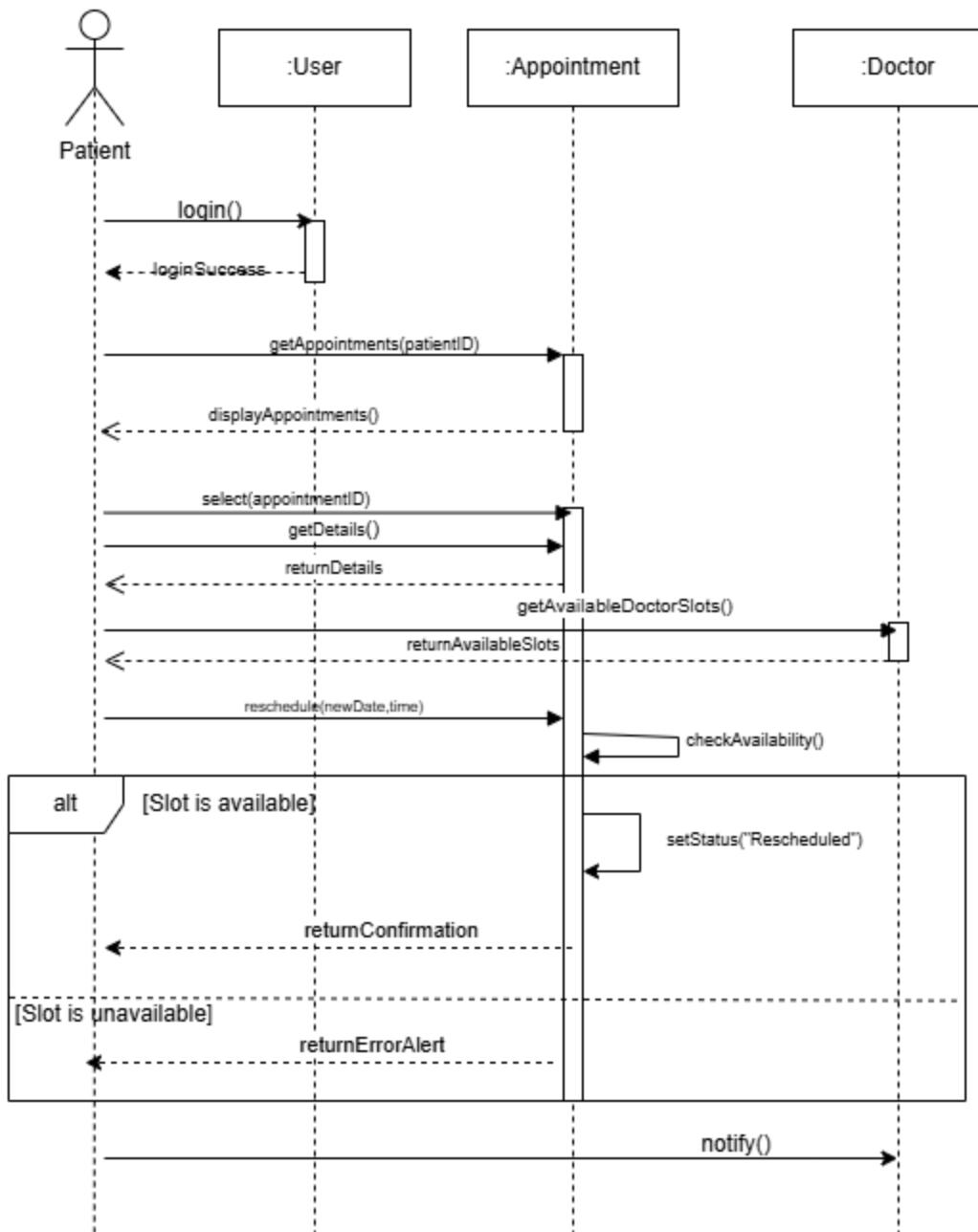


## Electronic Healthcare System Requirements Specification



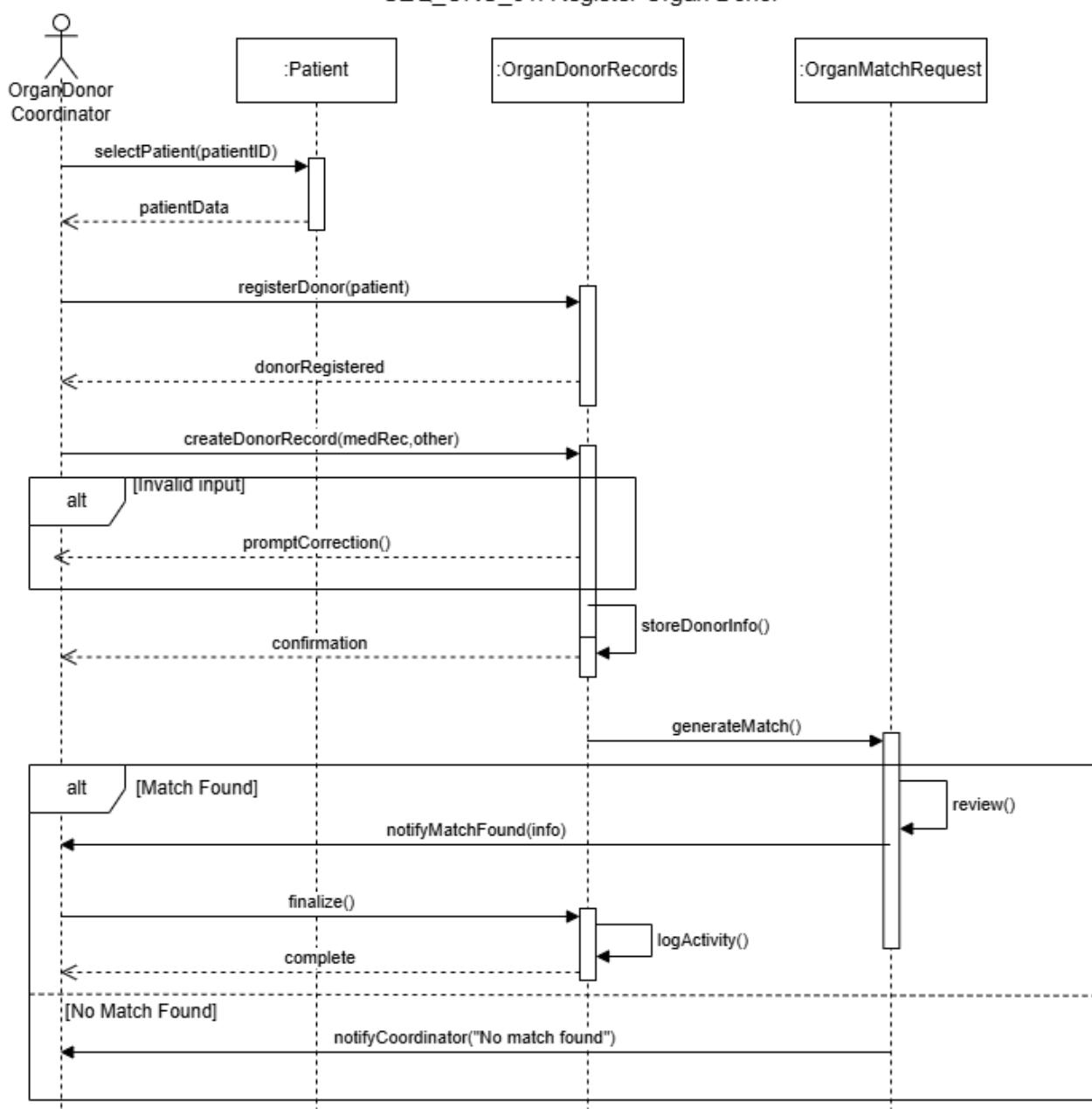
## Electronic Healthcare System Requirements Specification

SD\_EXT\_02



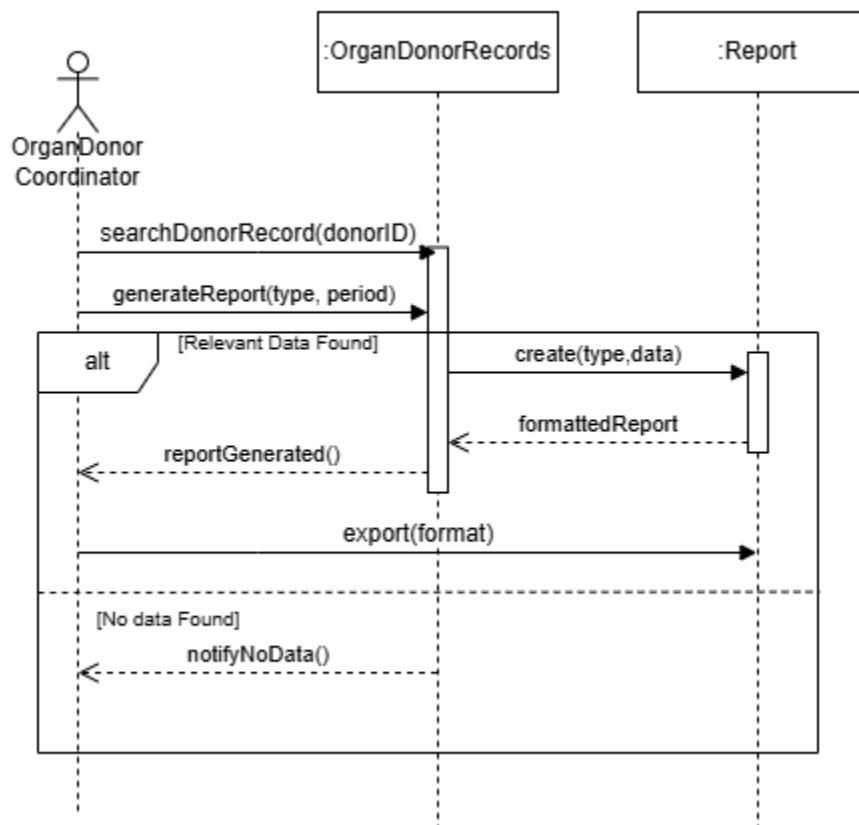
**Flavia Koco**

**SEQ\_ORG\_01: Register Organ Donor**



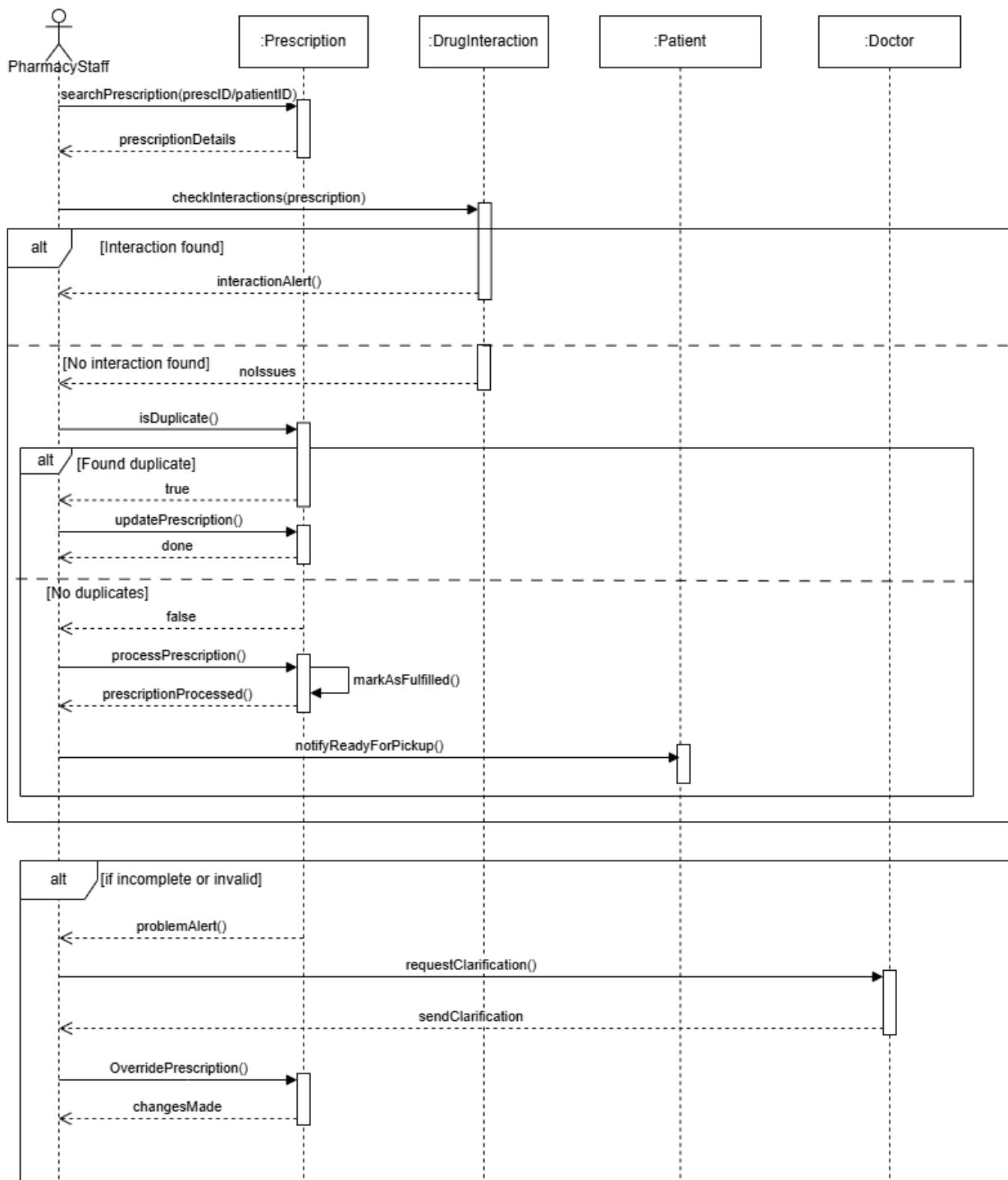
## *Electronic Healthcare System Requirements Specification*

### SEQ\_ORG\_02: Generate Reports



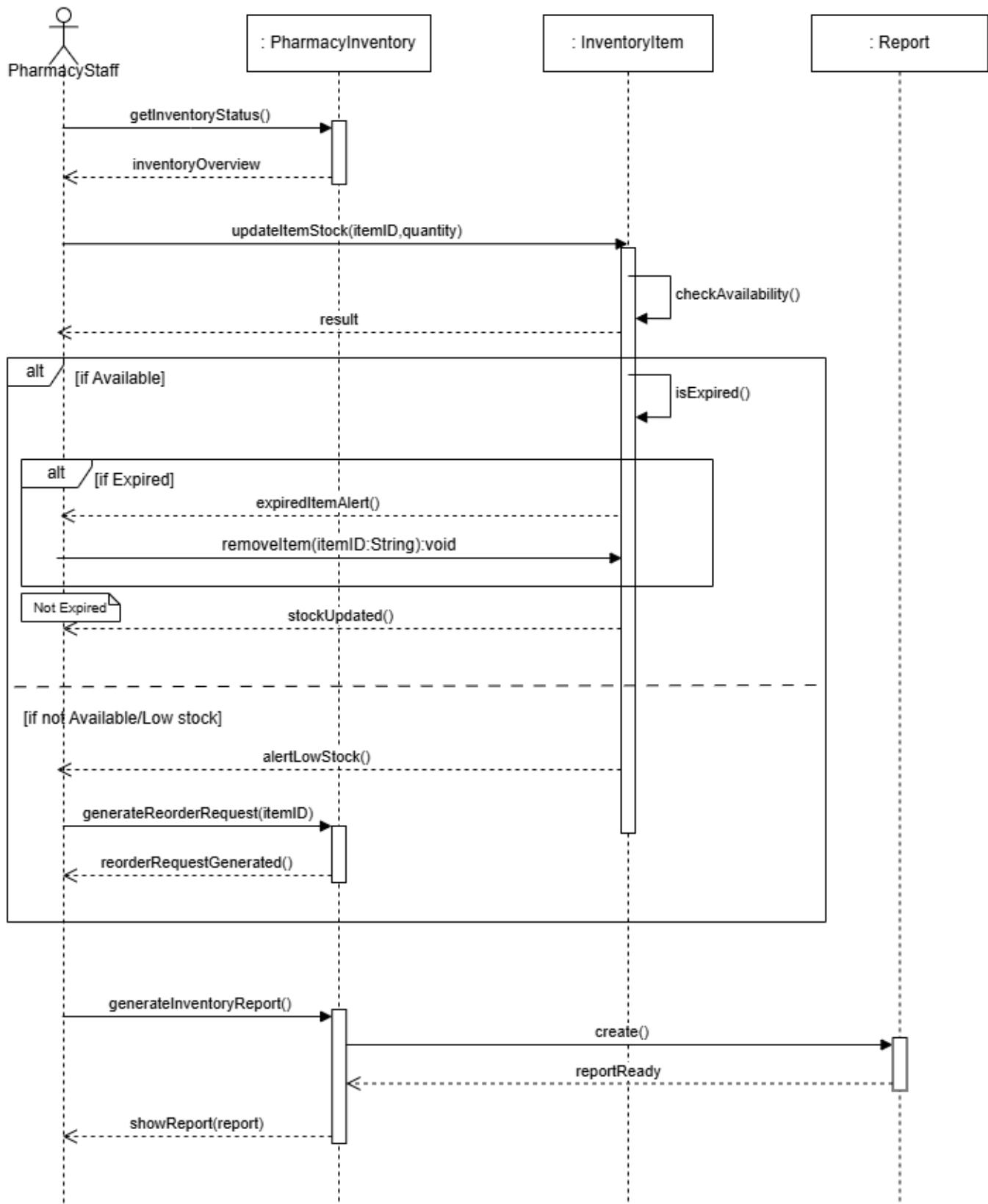
## Electronic Healthcare System Requirements Specification

### SEQ\_PH\_01: Prescription Management



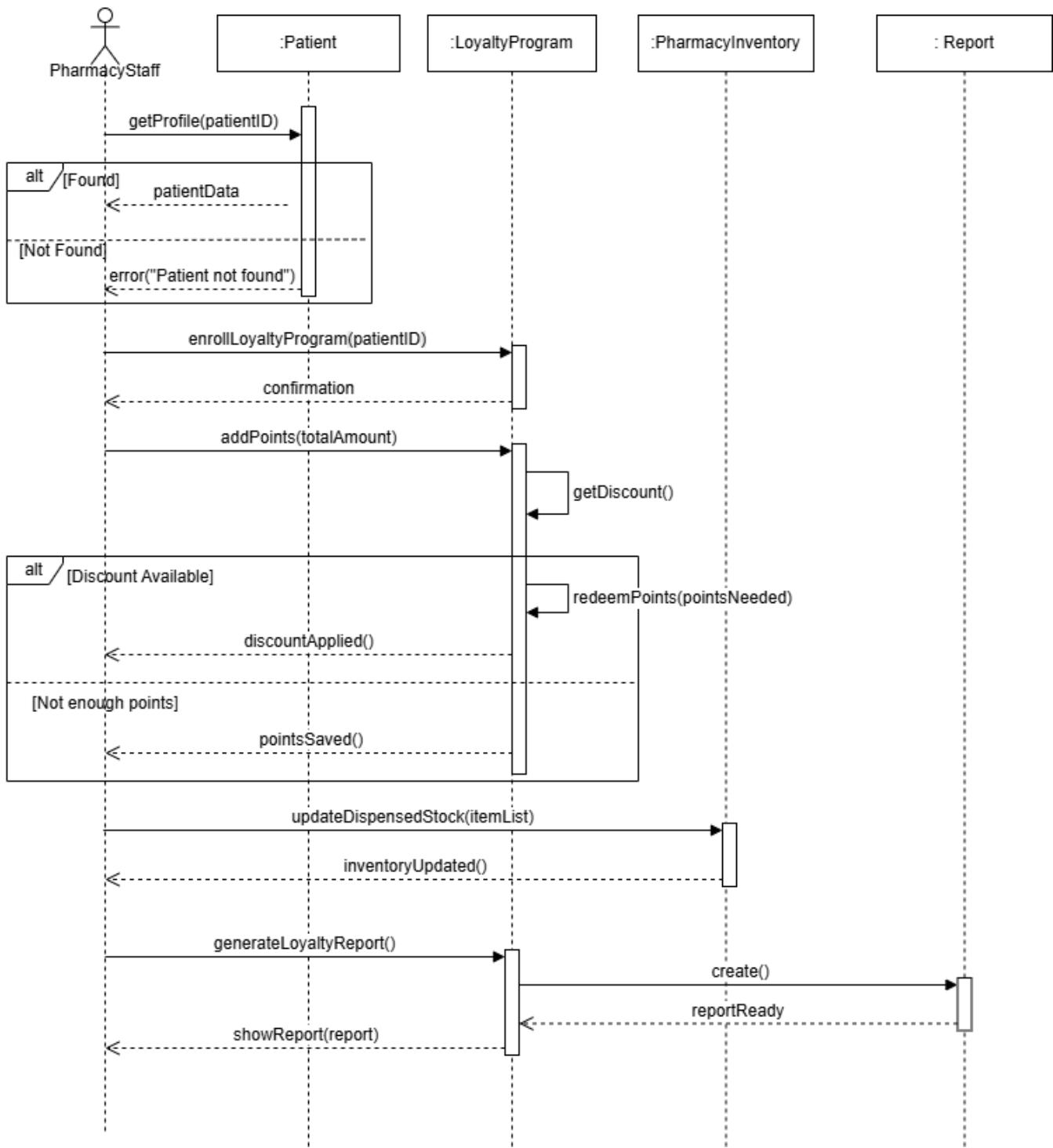
# Electronic Healthcare System Requirements Specification

## SEQ\_PH\_02: Pharmacy Inventory Management

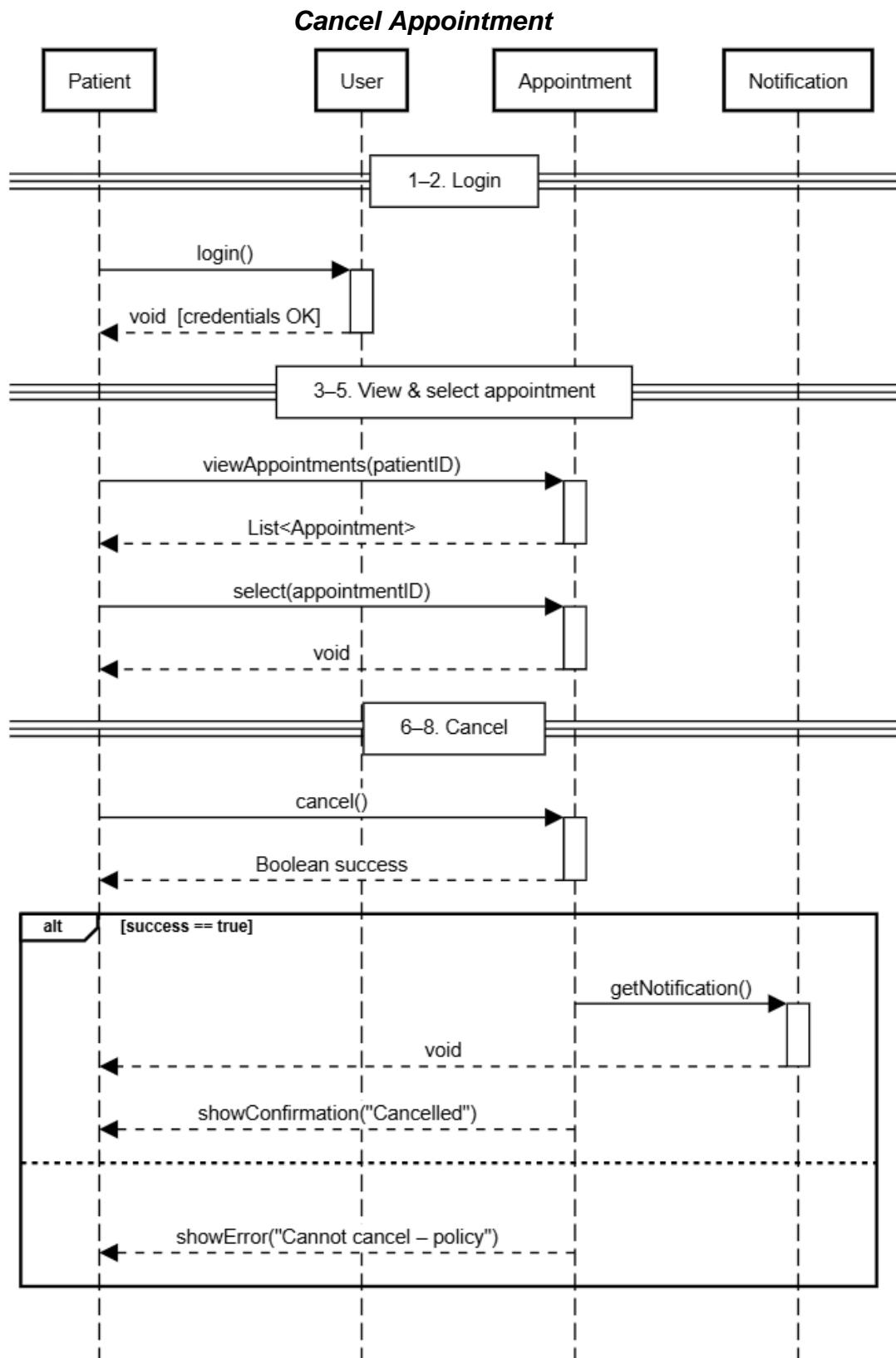


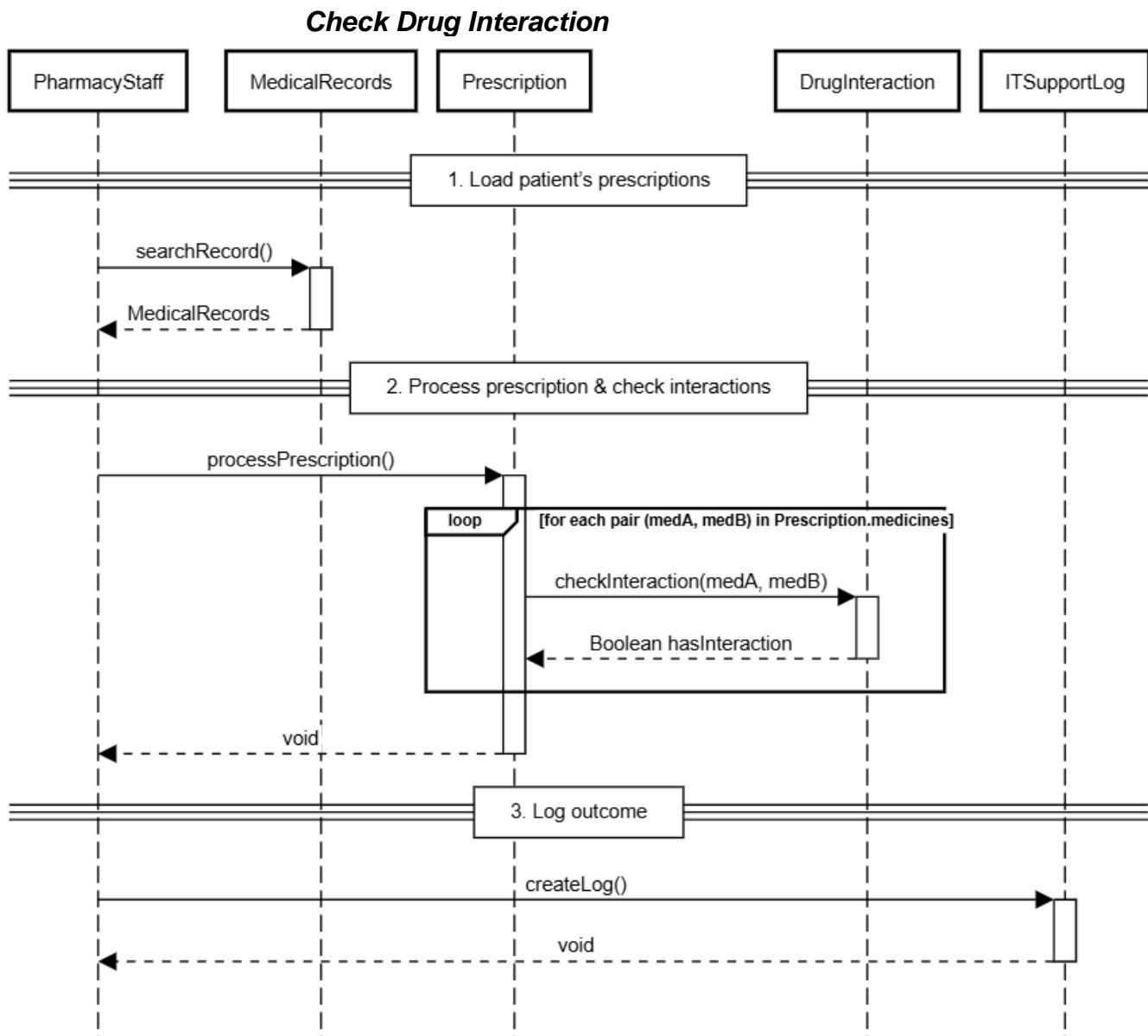
## Electronic Healthcare System Requirements Specification

### SEQ\_PH\_03: Customer Loyalty & Discount Program

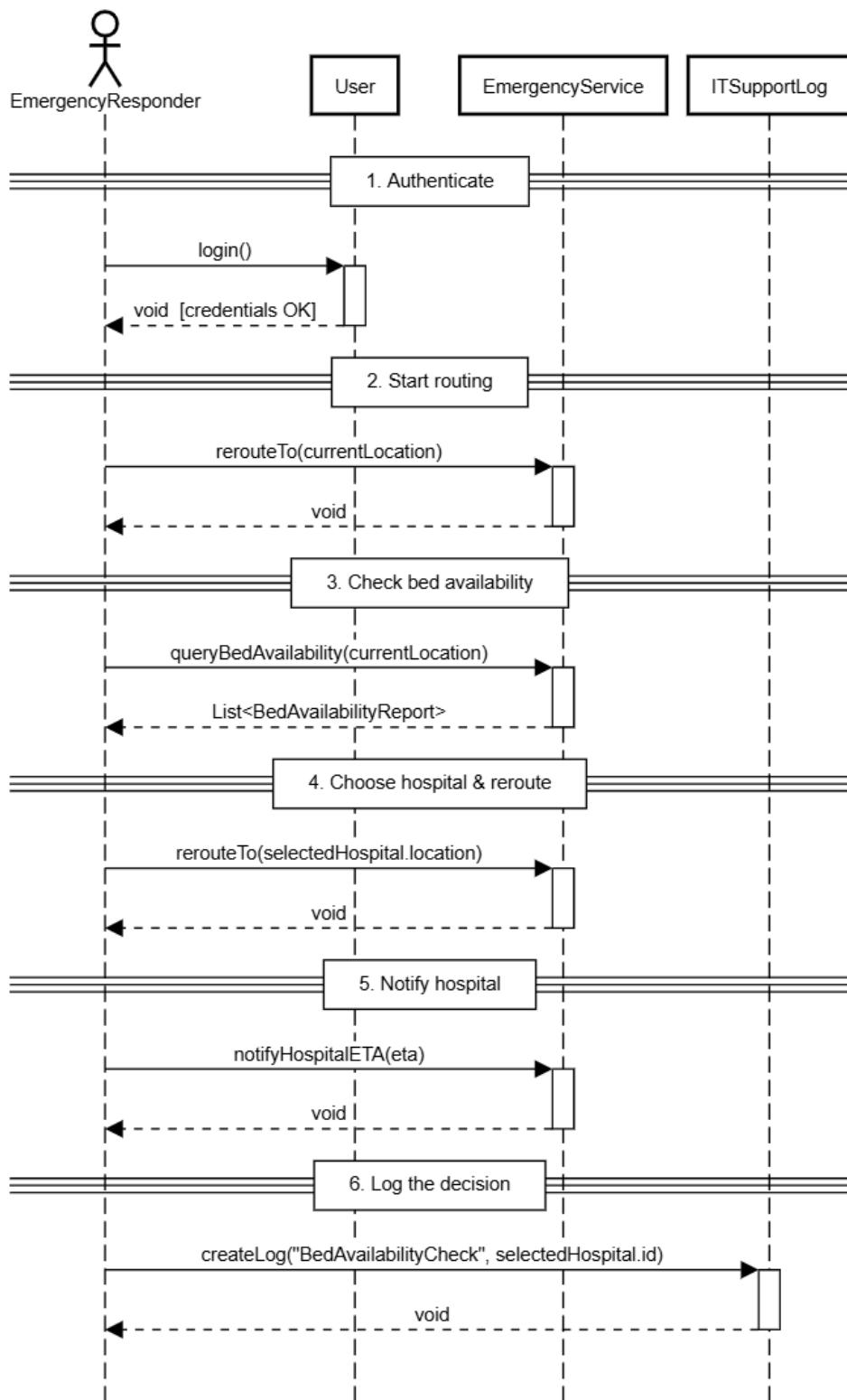


**Elisona Doku:**



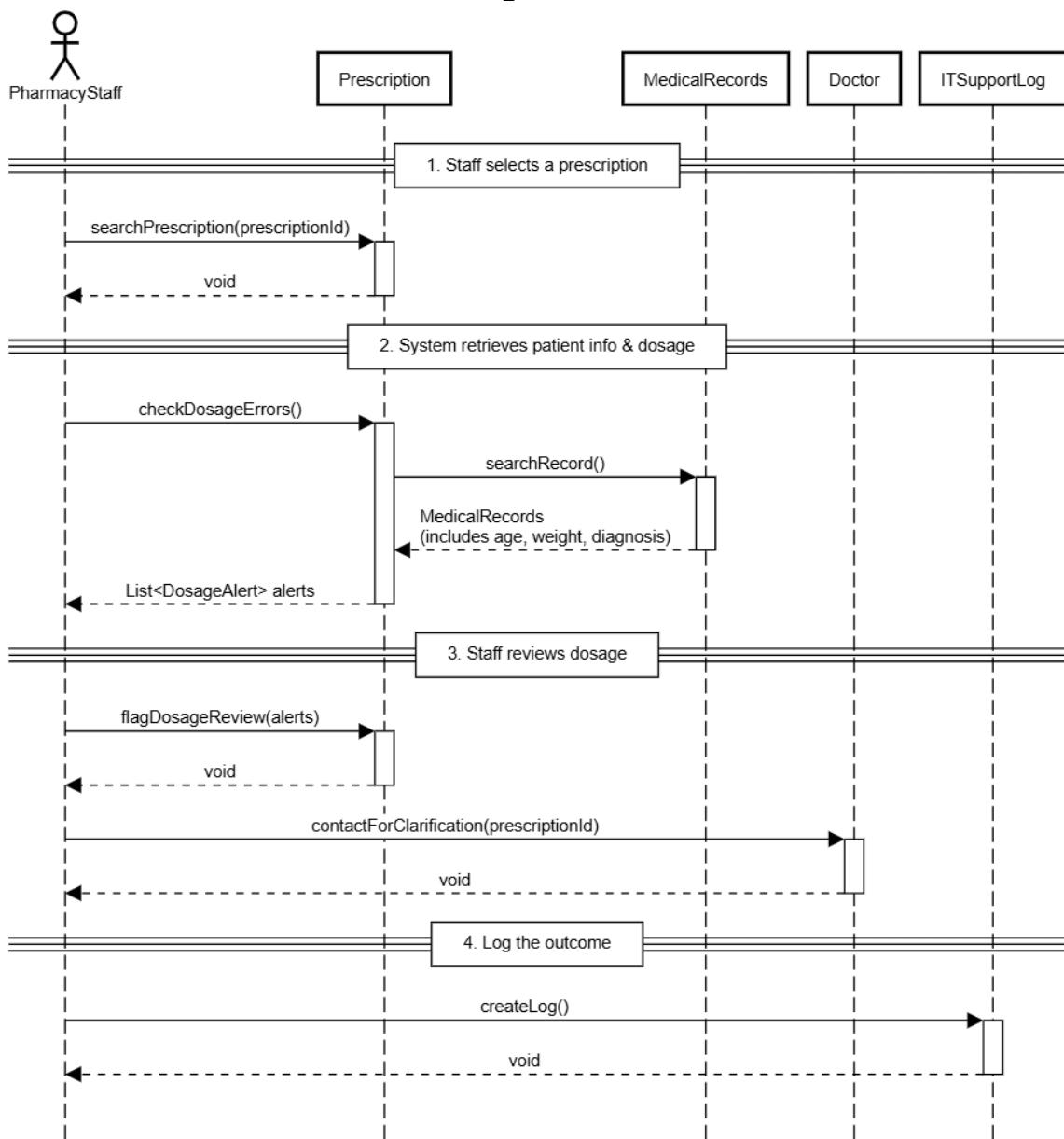


### **Check Hospital Bed Availability**

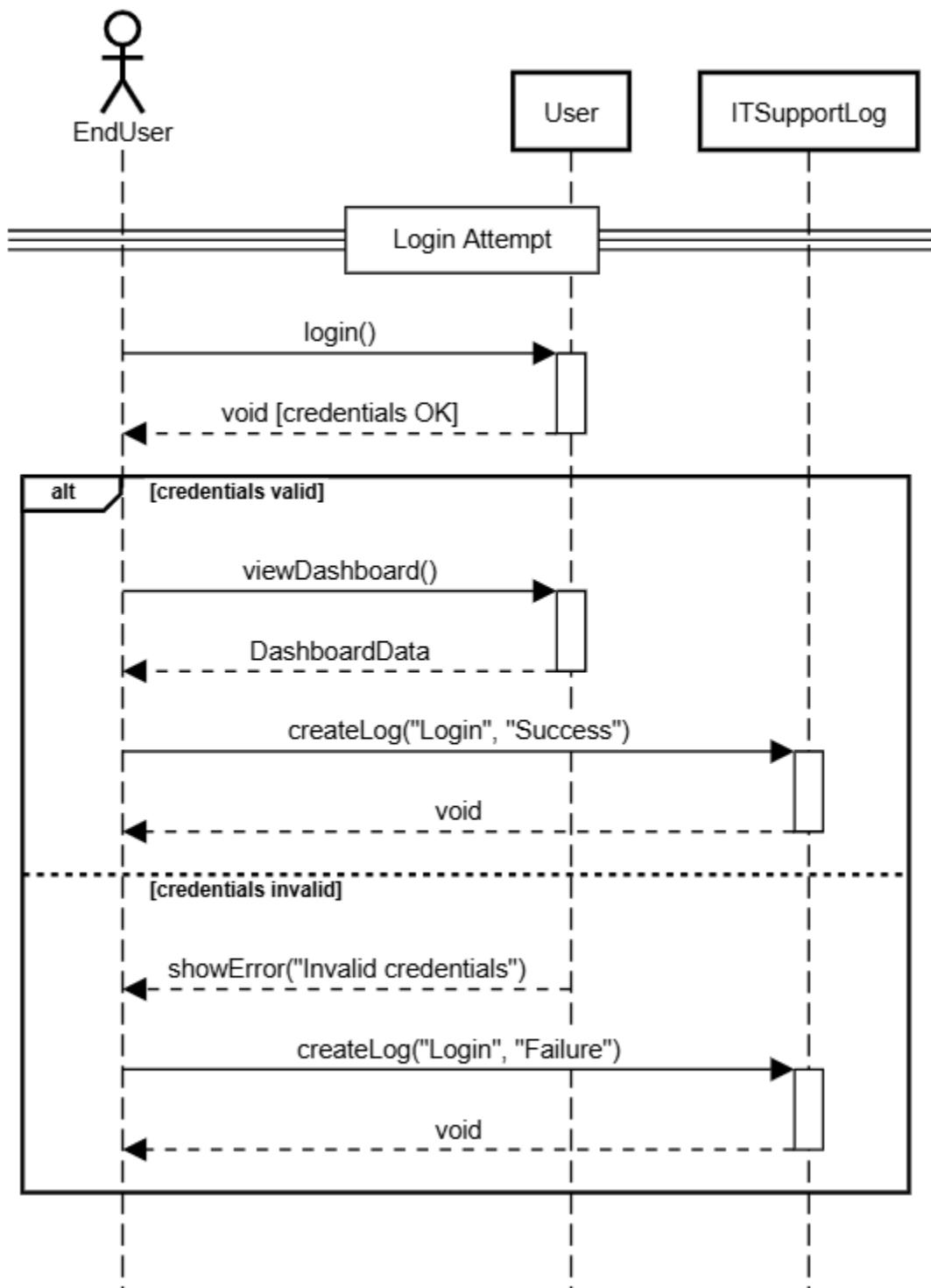


## Electronic Healthcare System Requirements Specification

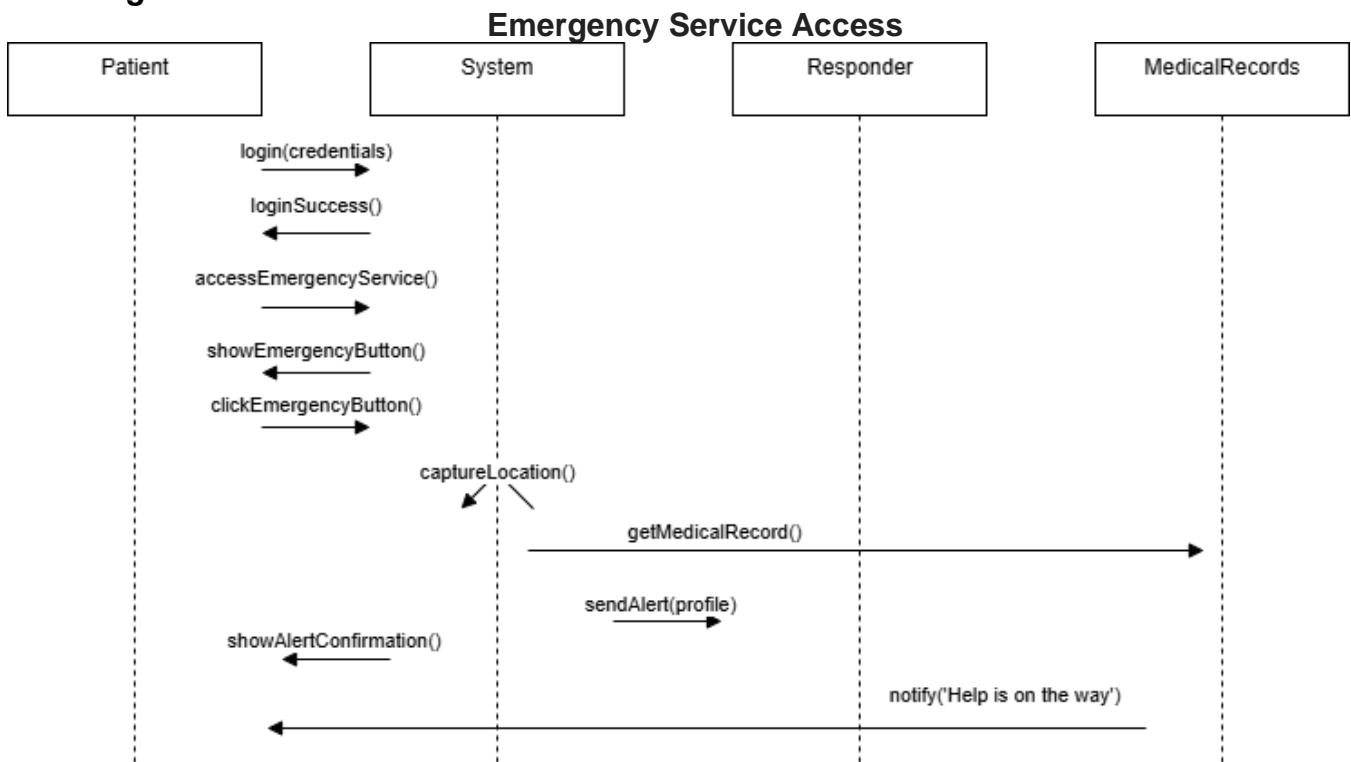
### Check Dosage Errors



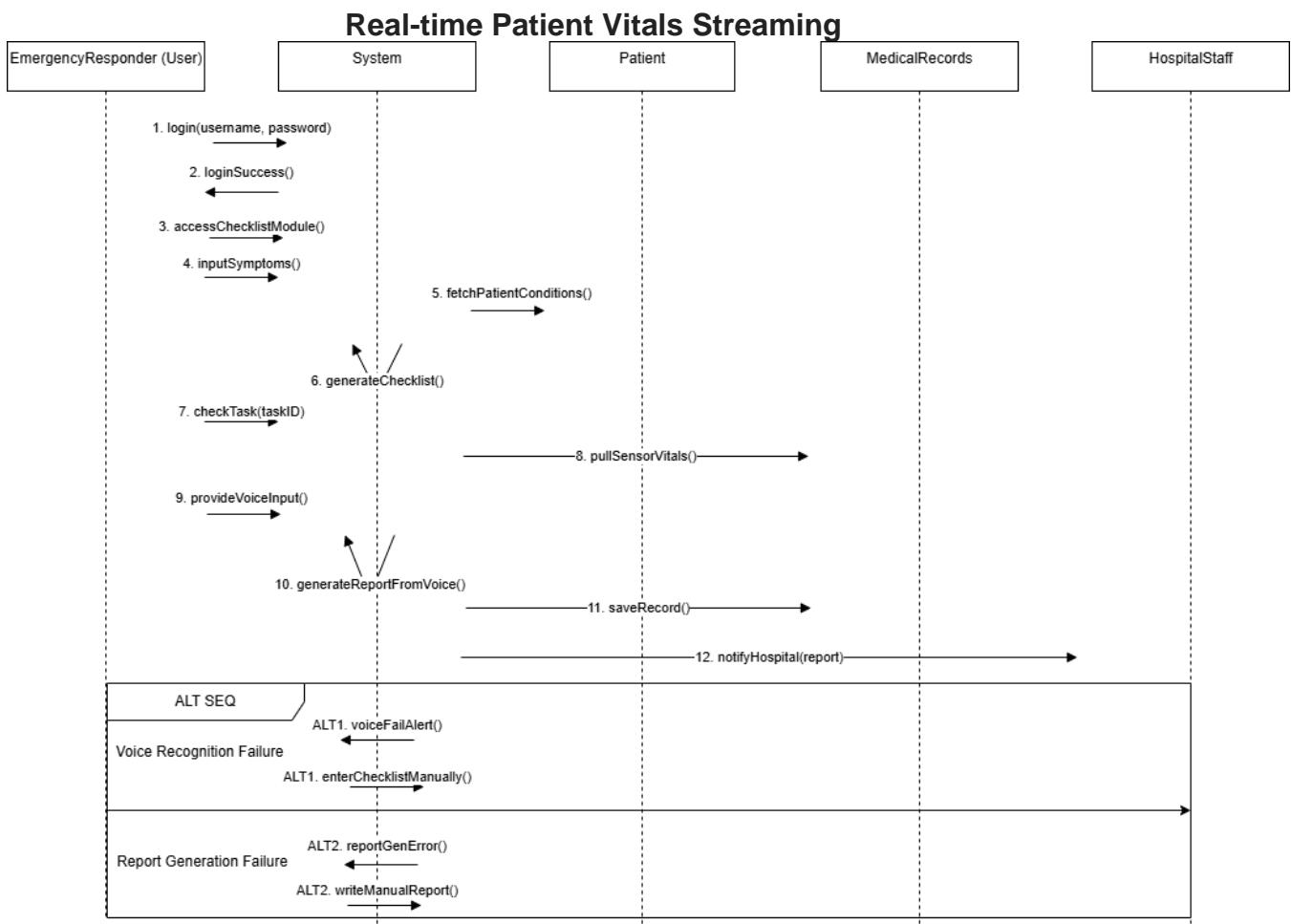
### User Login



**Elkier Ago:**

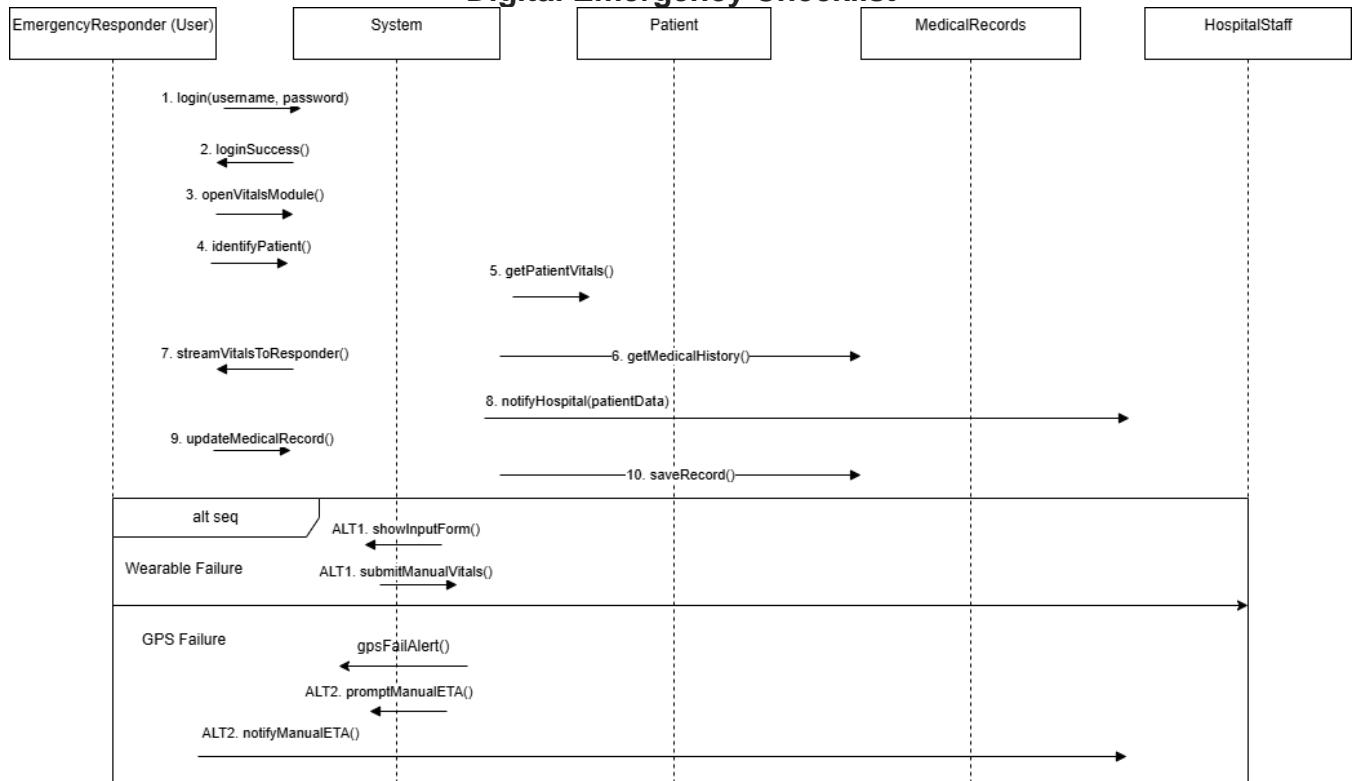


## Electronic Healthcare System Requirements Specification



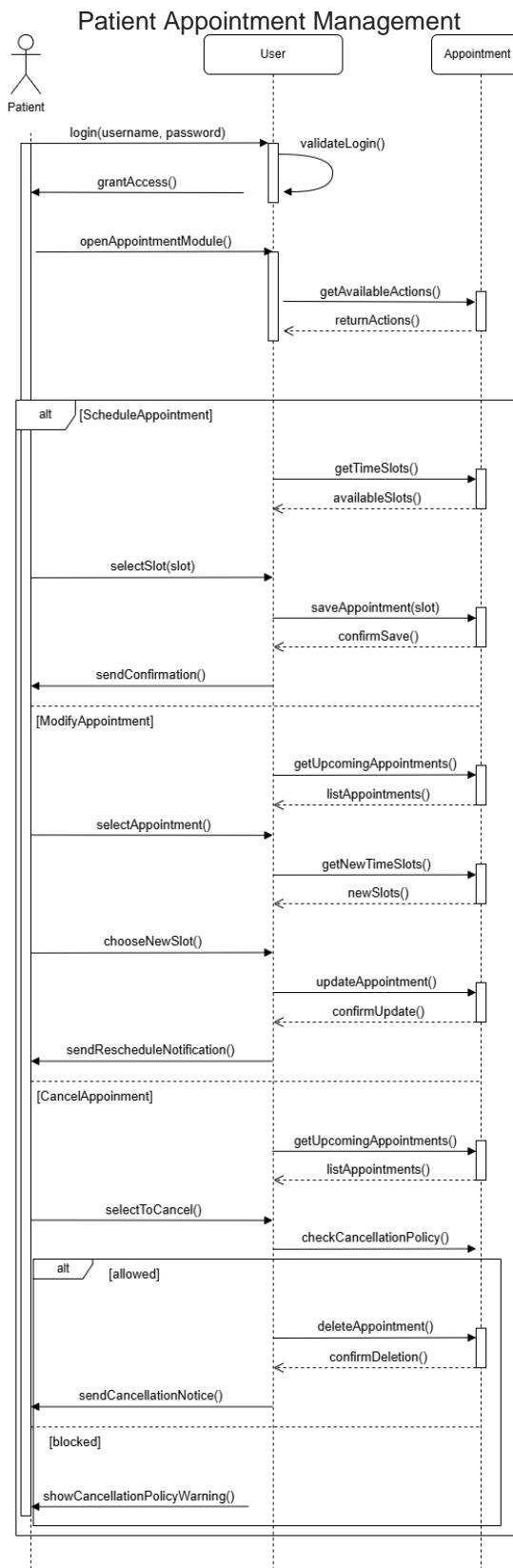
## Electronic Healthcare System Requirements Specification

### Digital Emergency Checklist

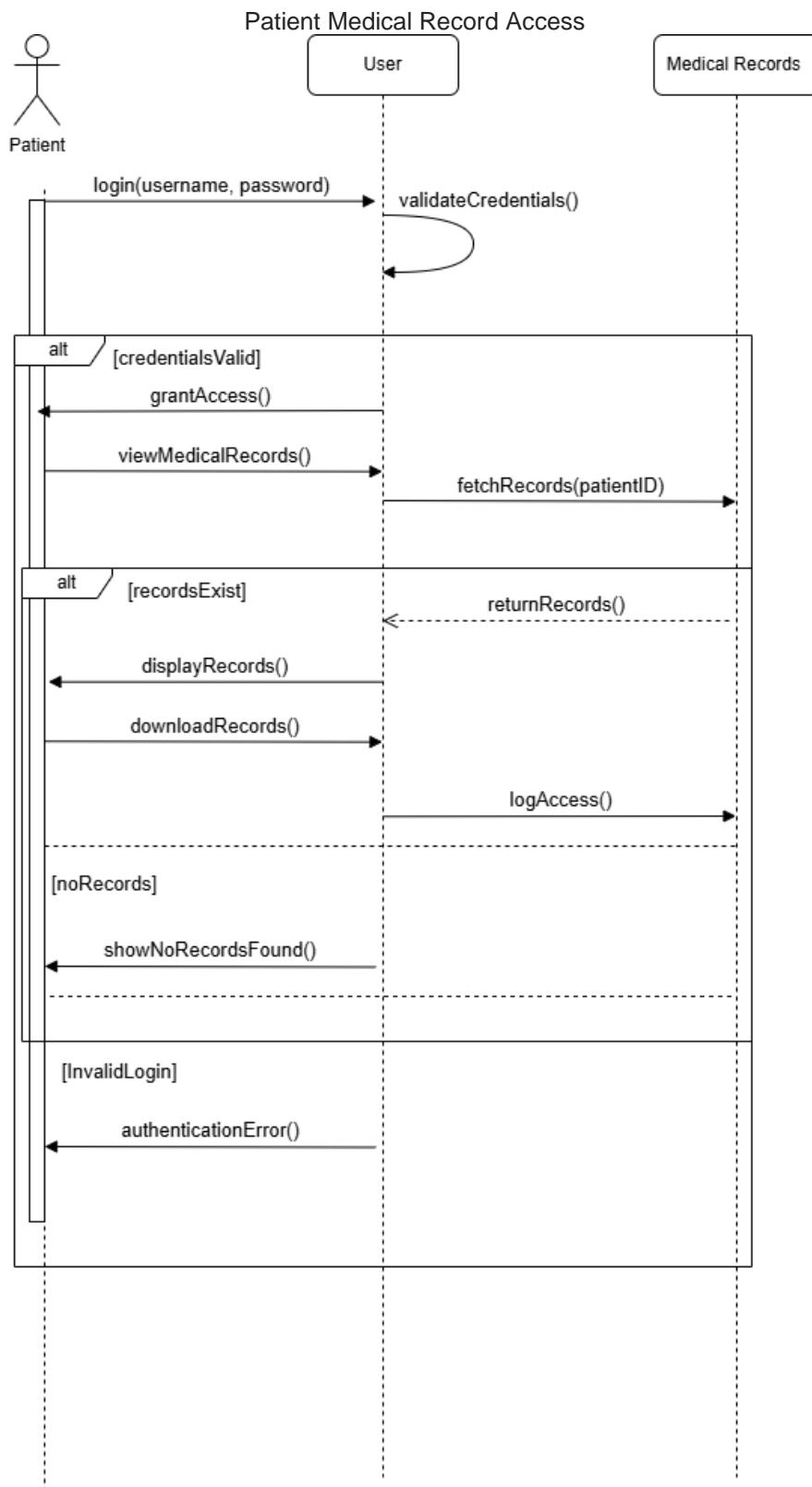


# Electronic Healthcare System Requirements Specification

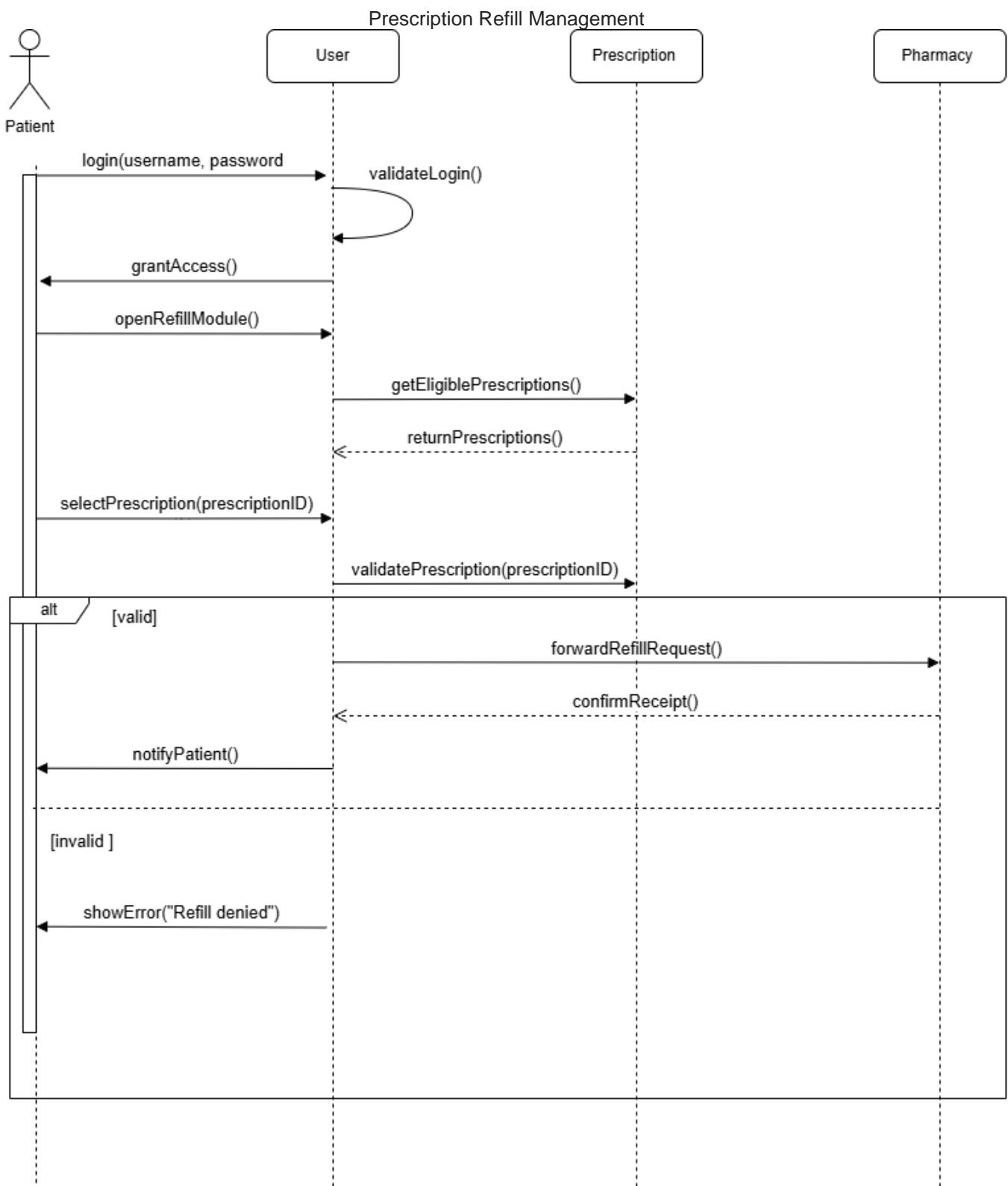
Evelina Gace



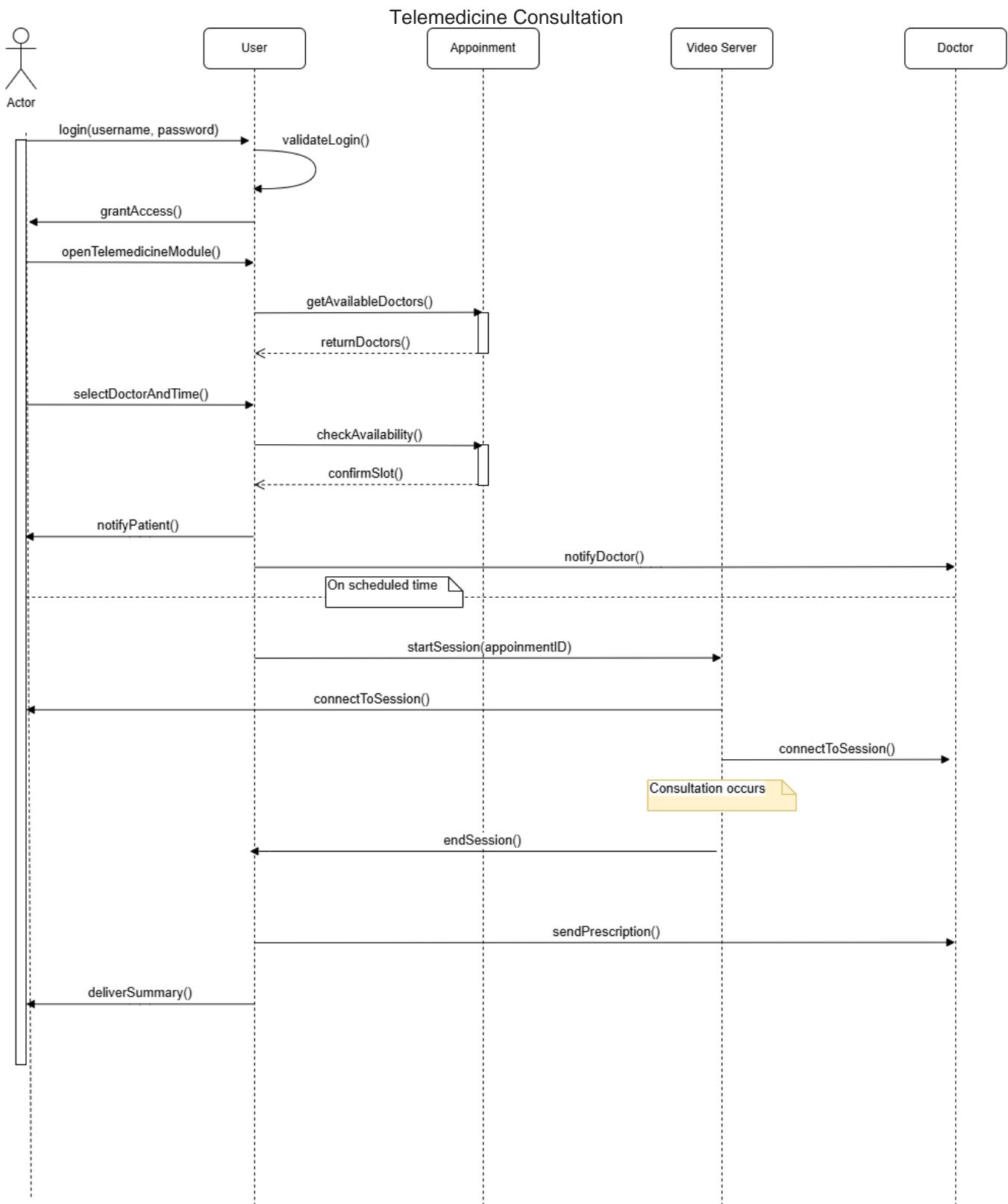
## Electronic Healthcare System Requirements Specification



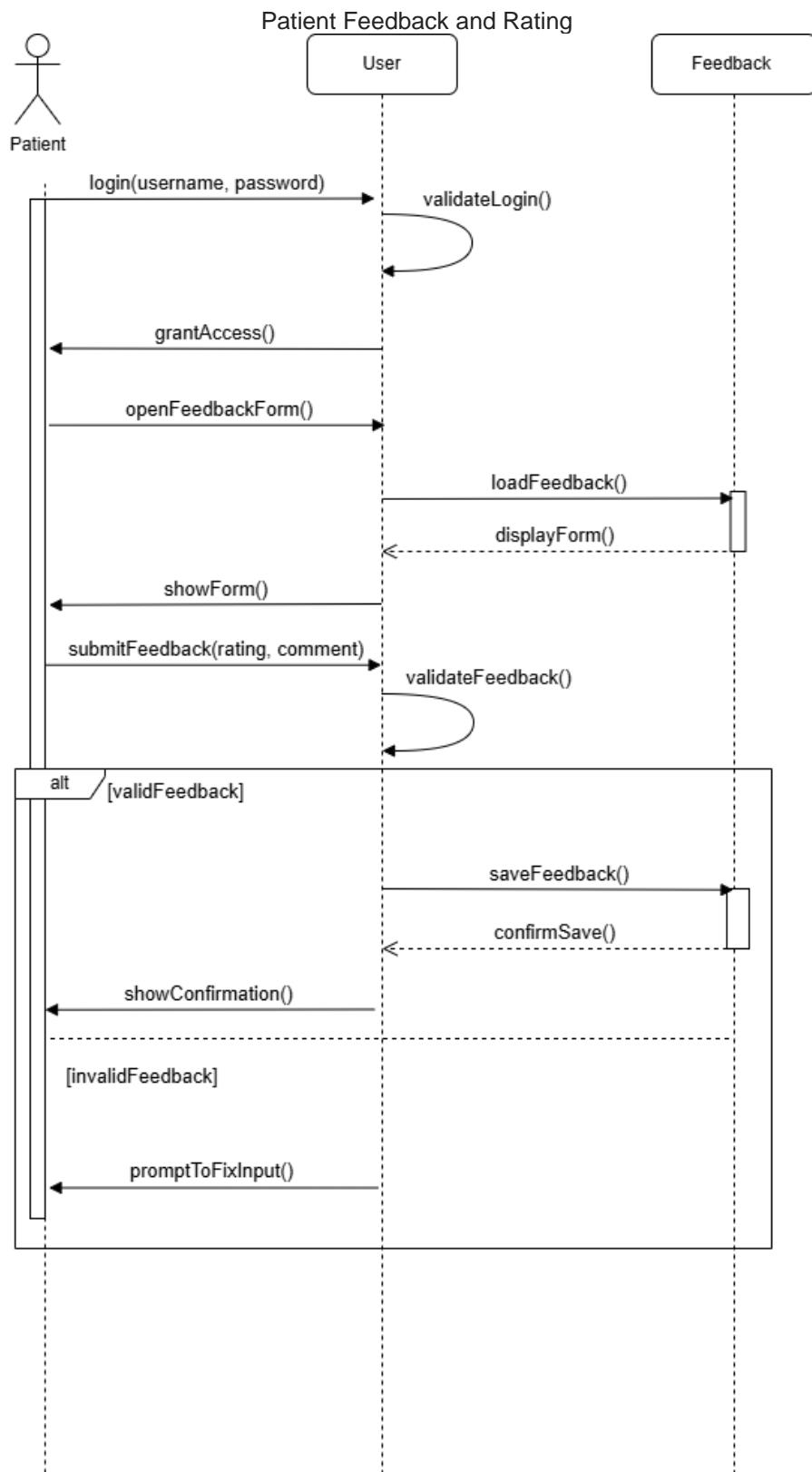
## Electronic Healthcare System Requirements Specification



## Electronic Healthcare System Requirements Specification



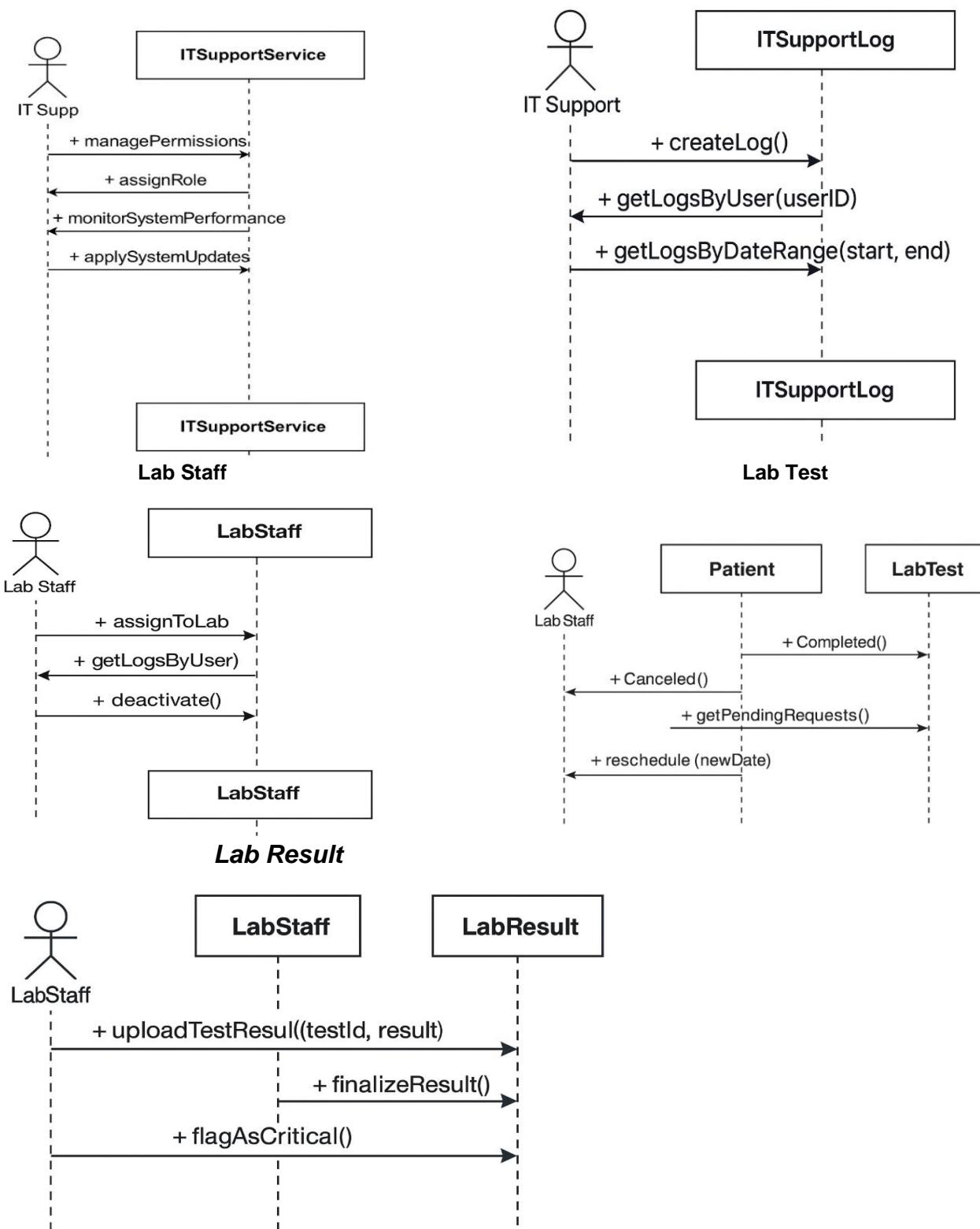
## Electronic Healthcare System Requirements Specification



**Liza Koliqi**

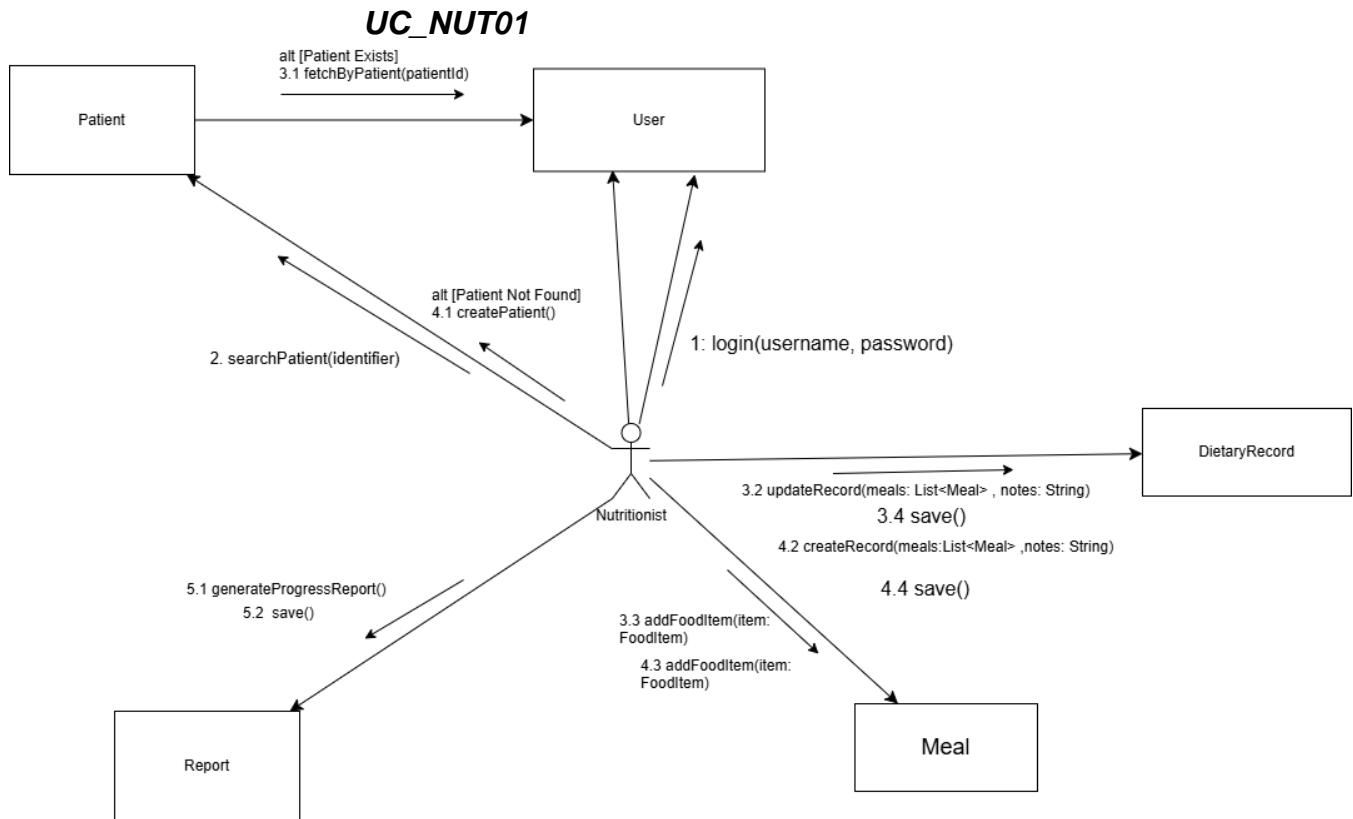
**IT Support Service**

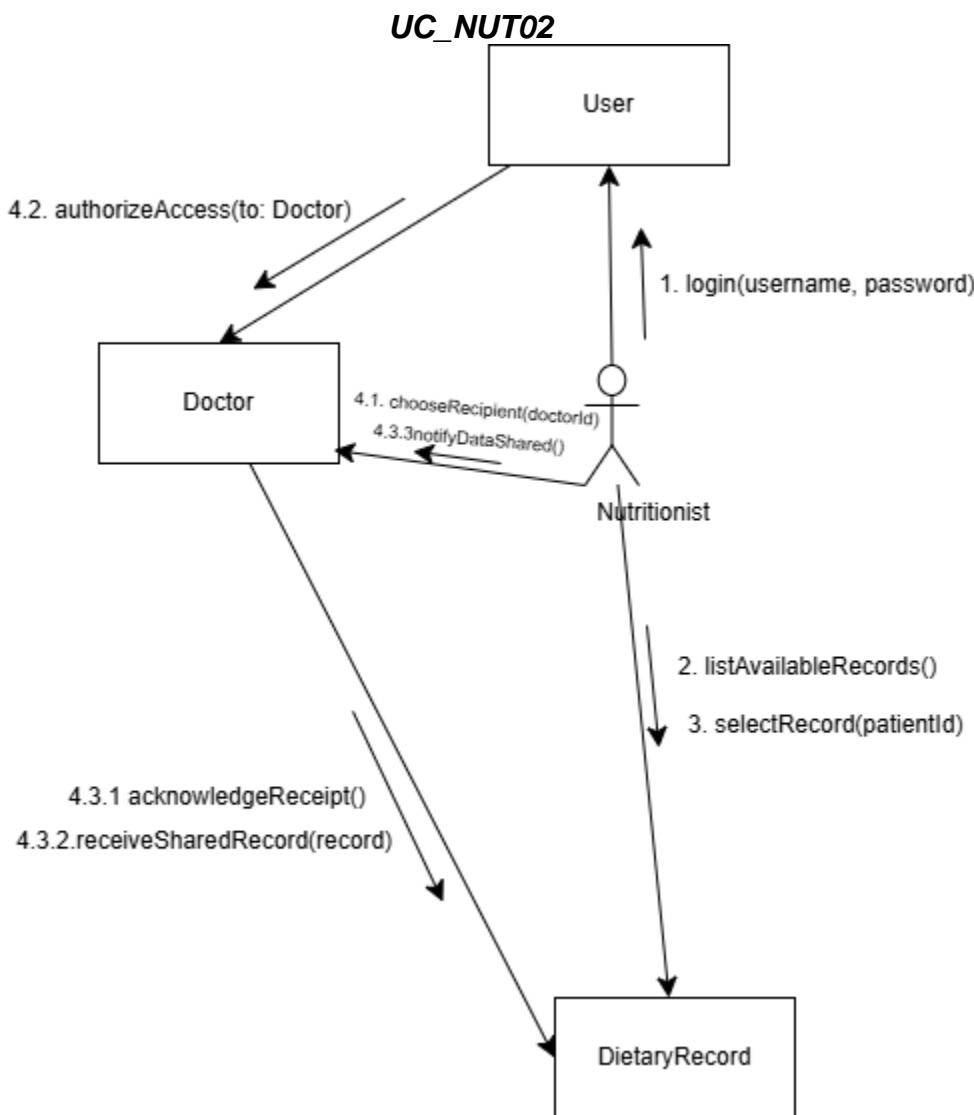
**IT Support**



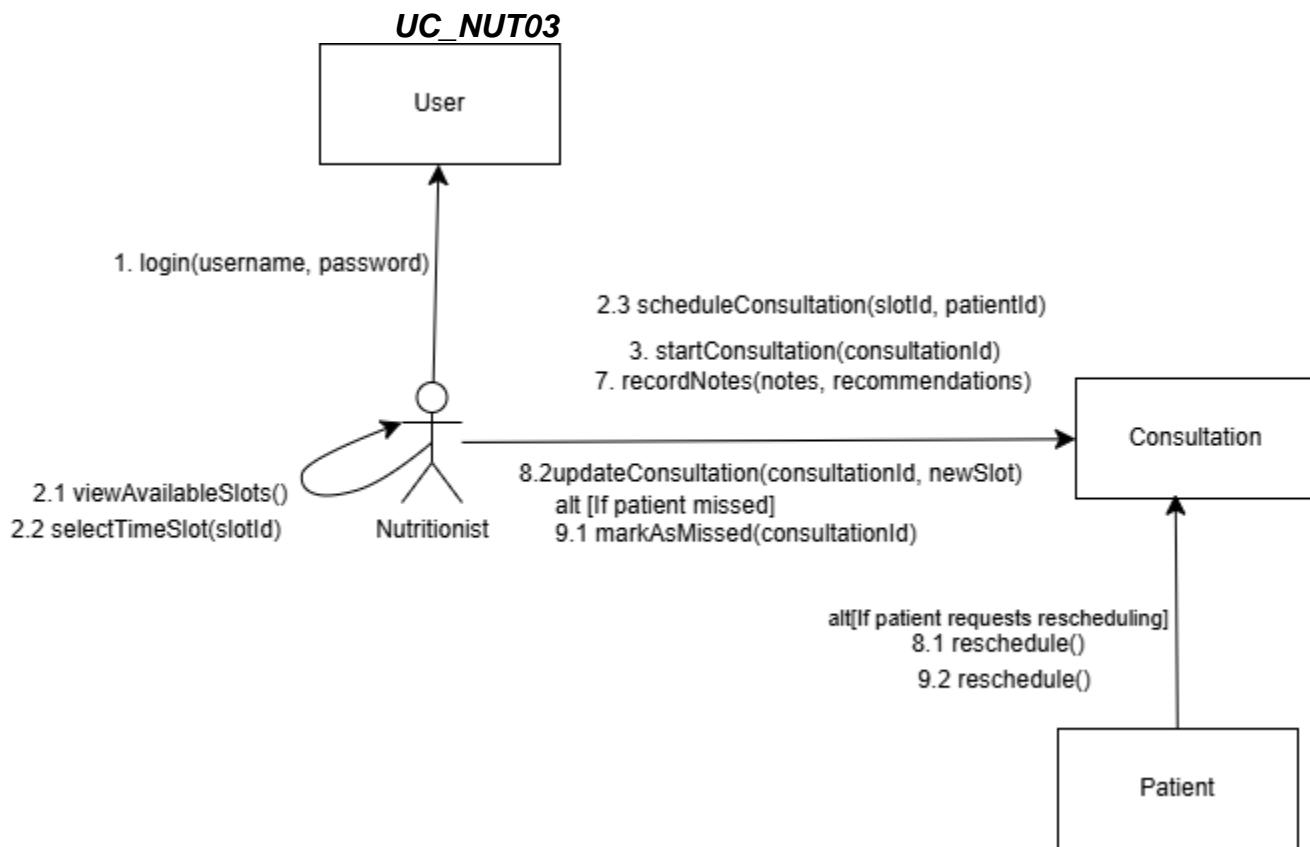
## 5.7 Collaboration Diagrams

**Belina Durmishi:**



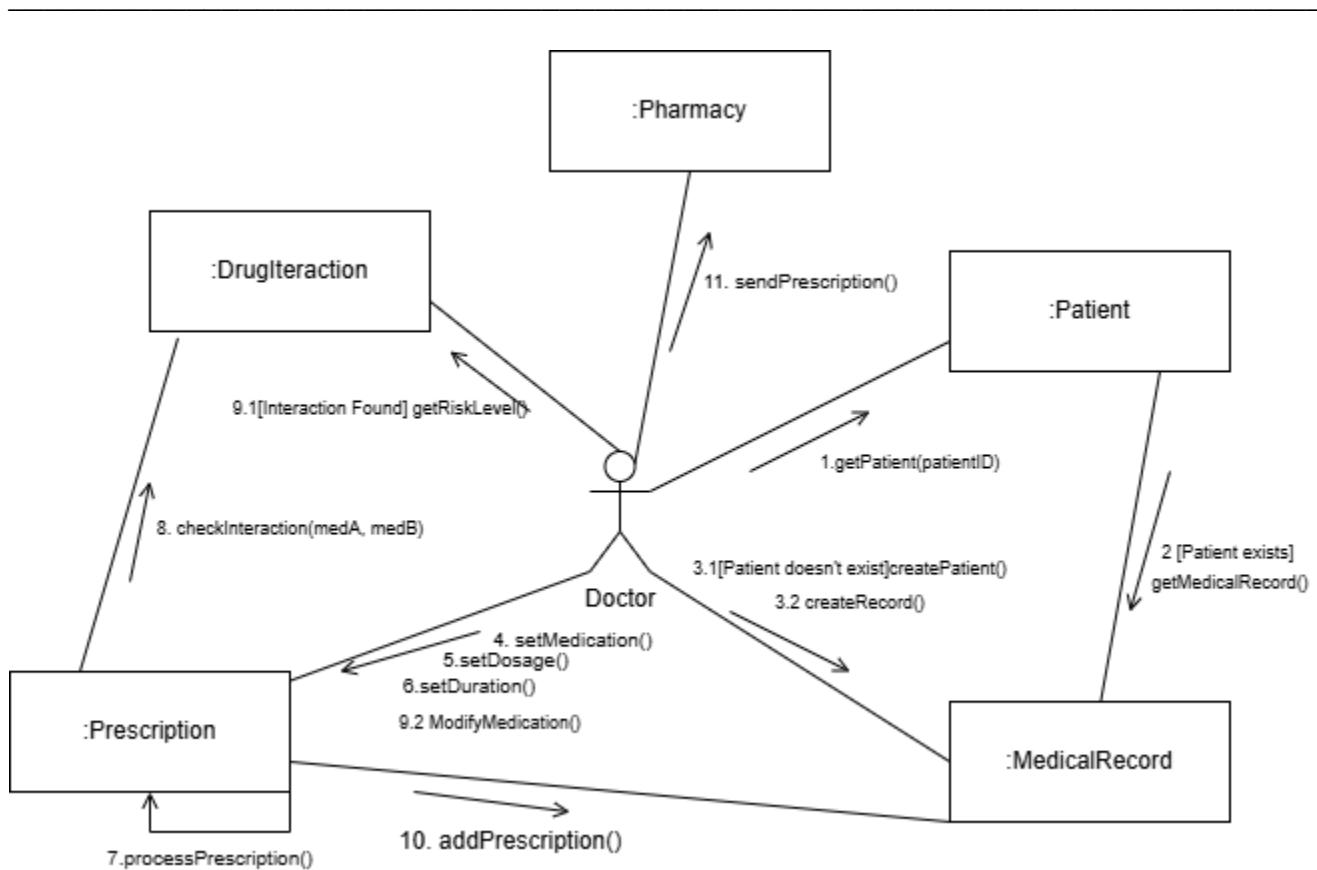
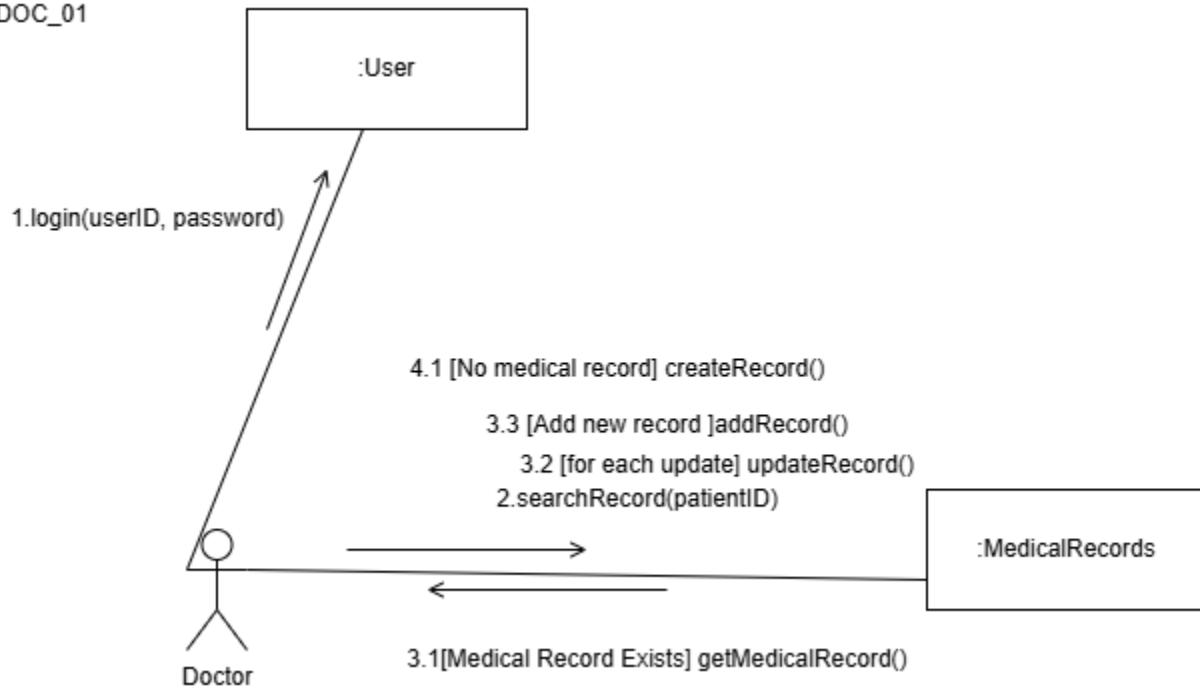


**Electronic Healthcare System Requirements Specification**



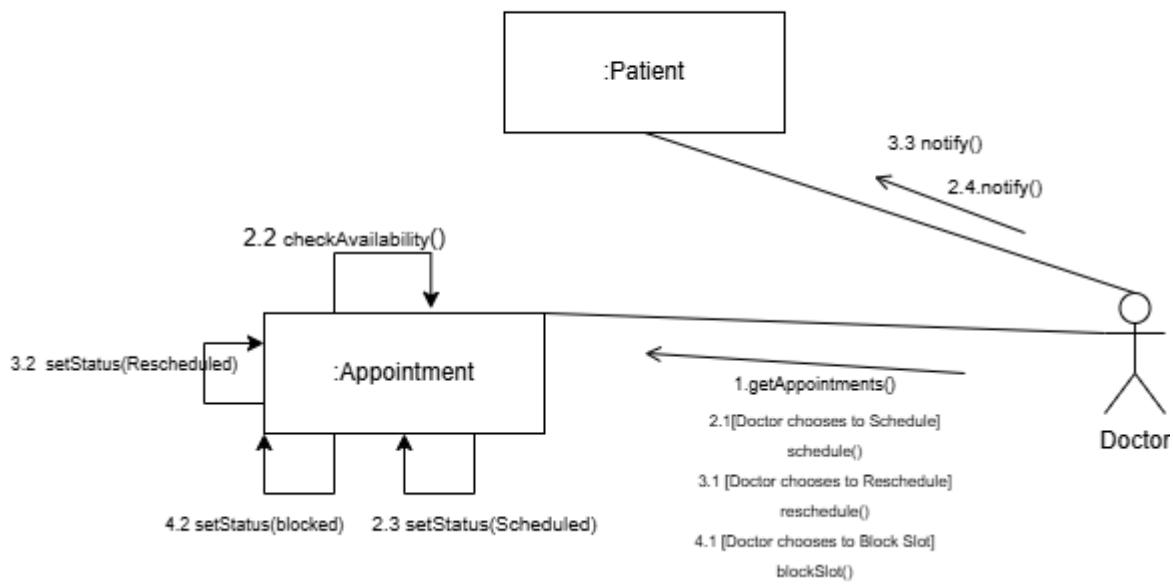
**Anteakoxherri:**

CD\_DOC\_01

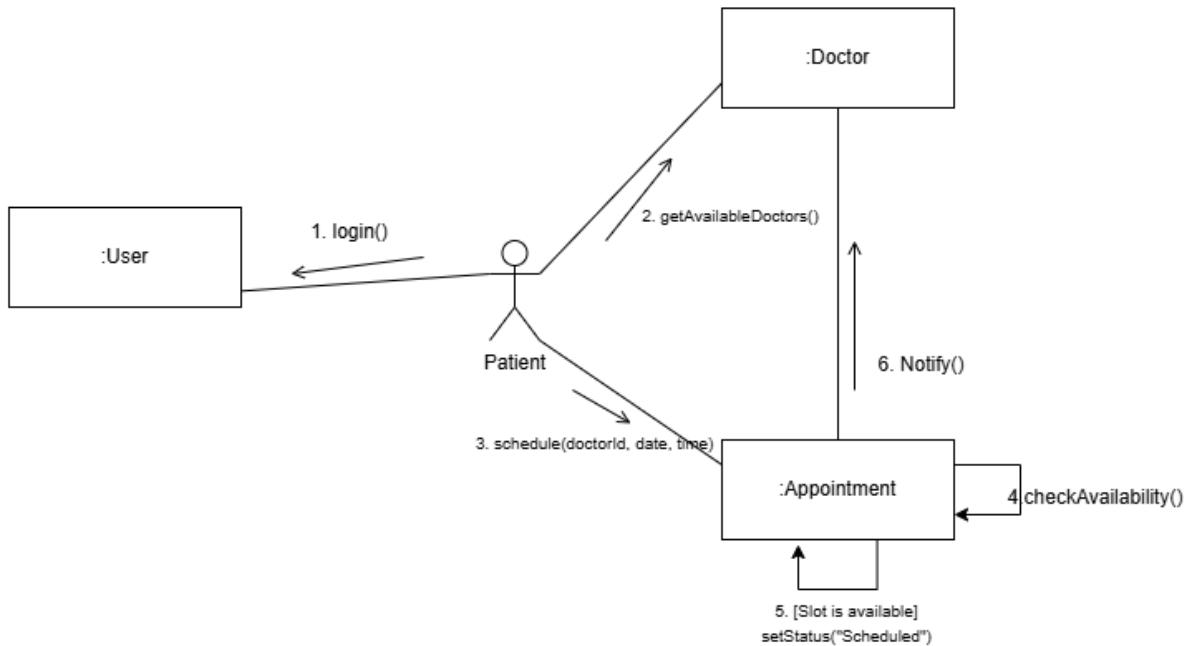


## Electronic Healthcare System Requirements Specification

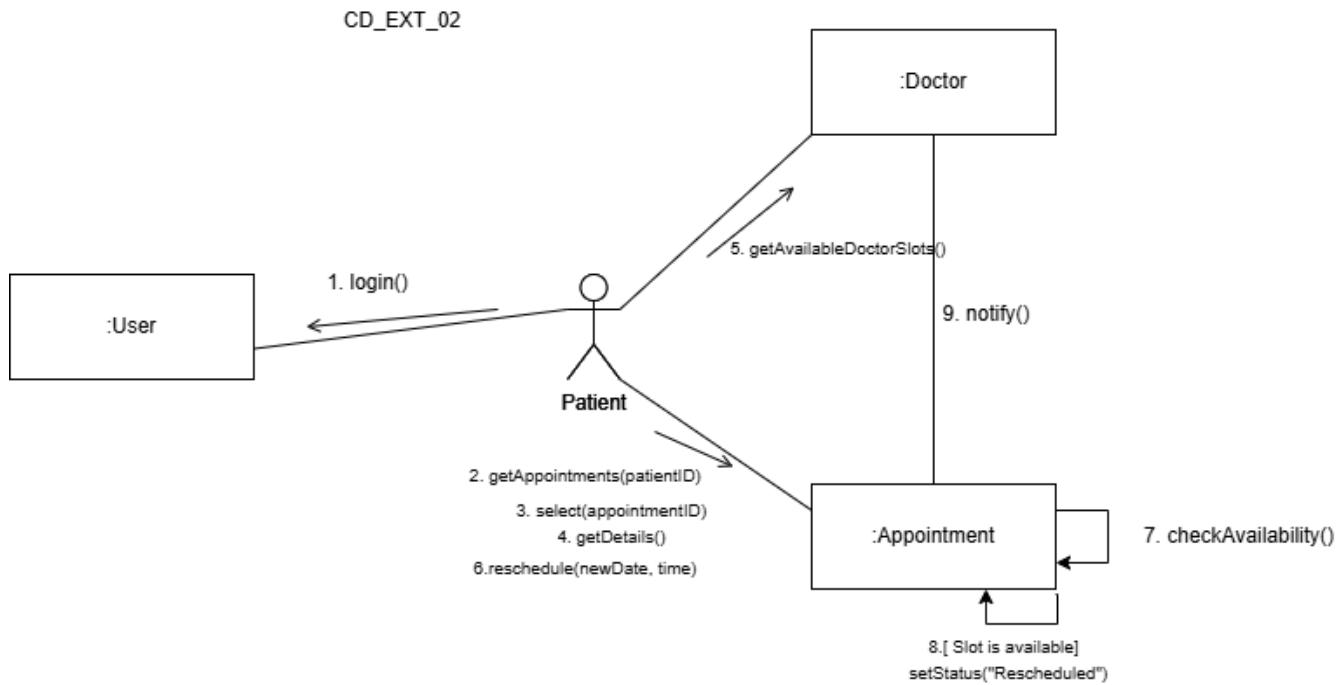
CD\_DOC\_03



CD\_EXT\_01

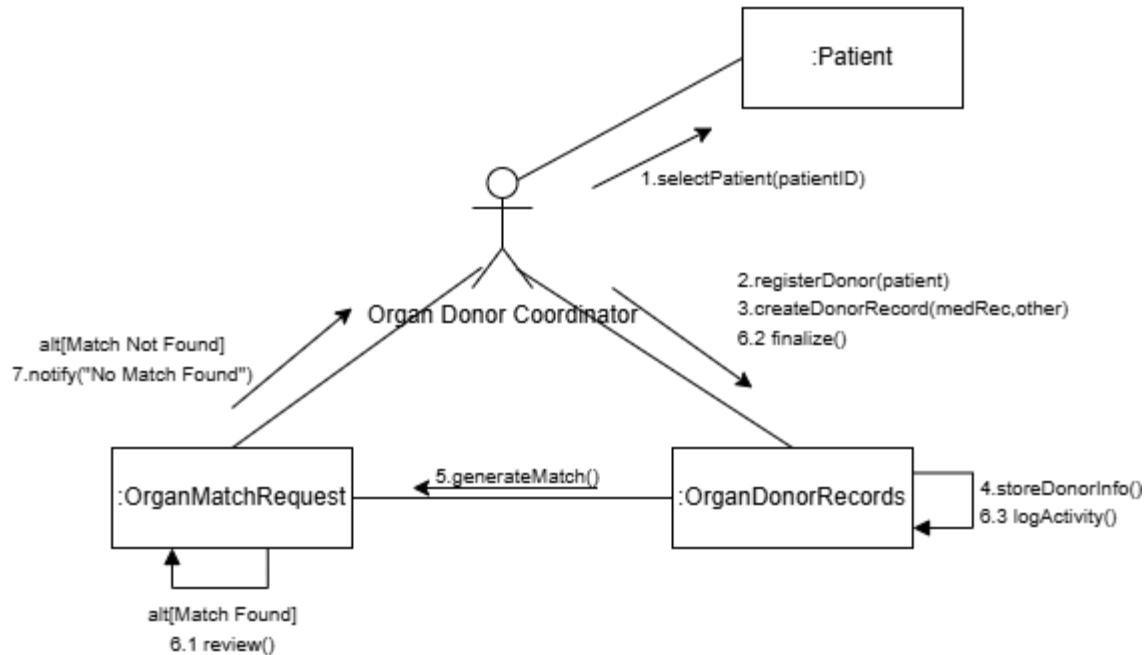


## Electronic Healthcare System Requirements Specification



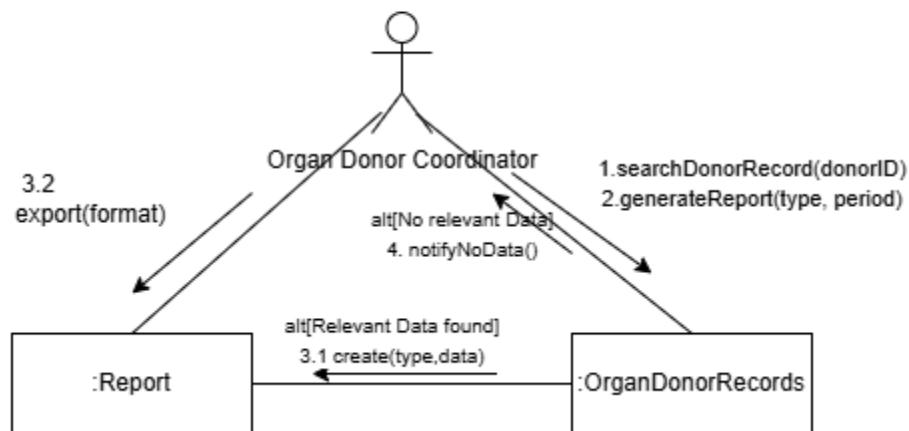
**Flavia Koco**

CD\_ORG\_01: Register Organ Donor

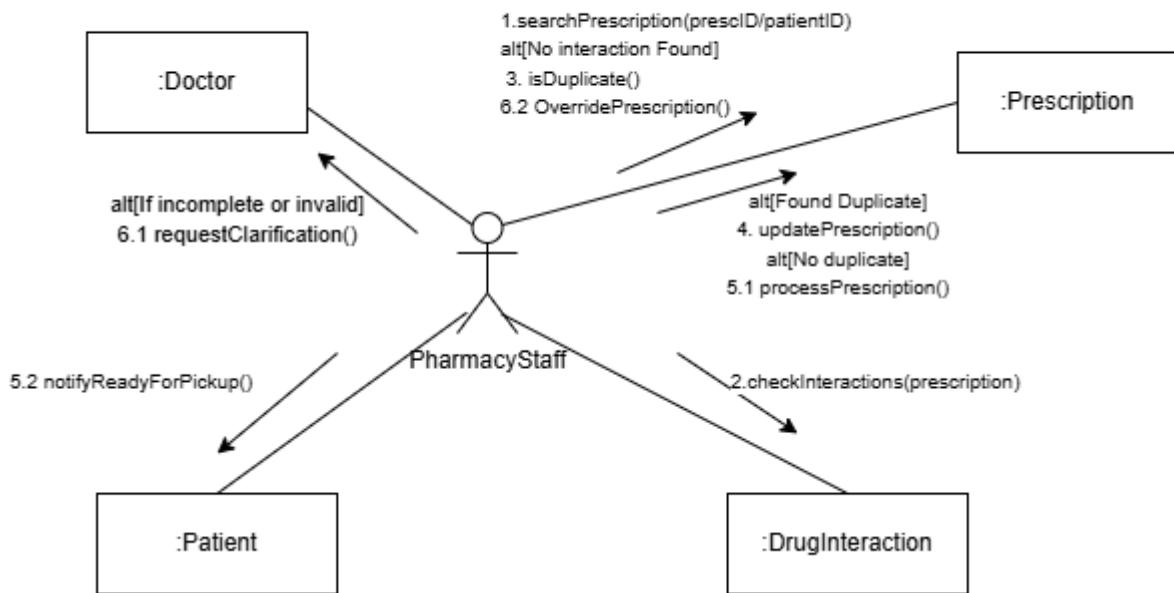


## Electronic Healthcare System Requirements Specification

### CD\_ORG\_02: Generate Reports

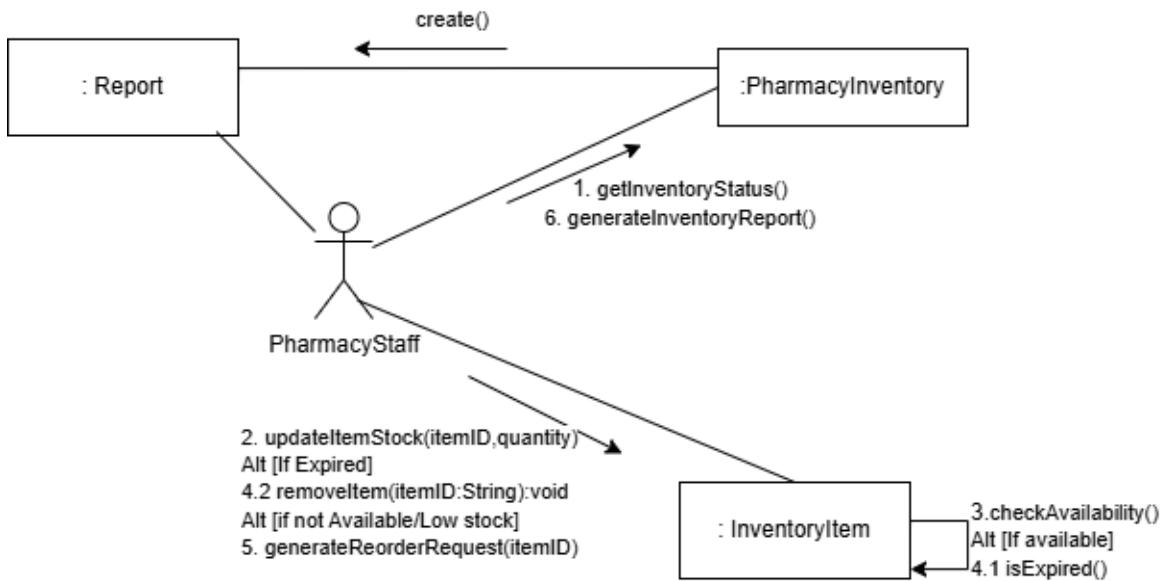


### CD\_PH\_01: Prescription Management

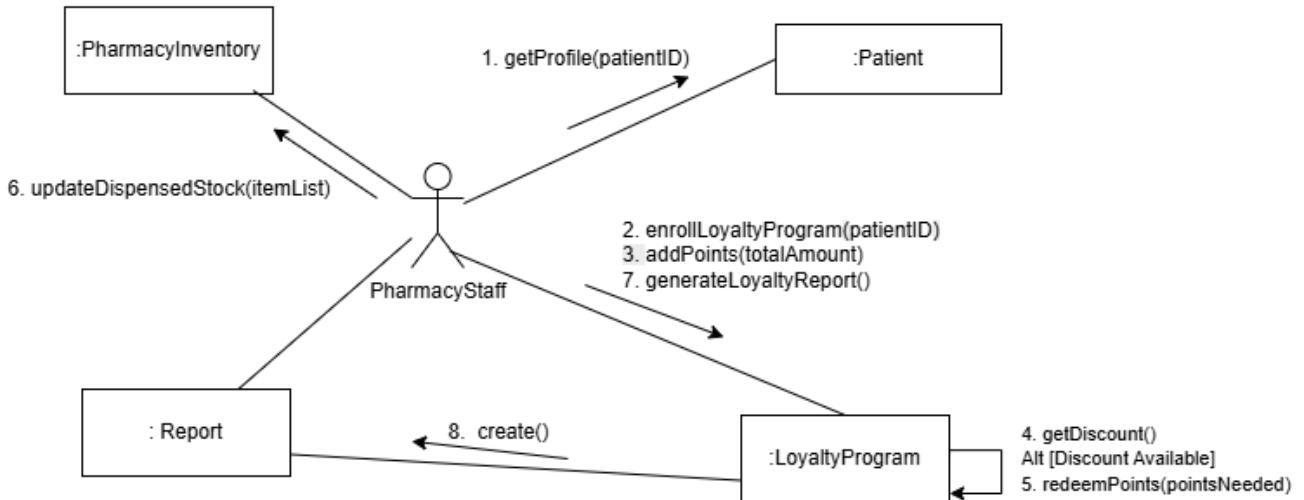


## Electronic Healthcare System Requirements Specification

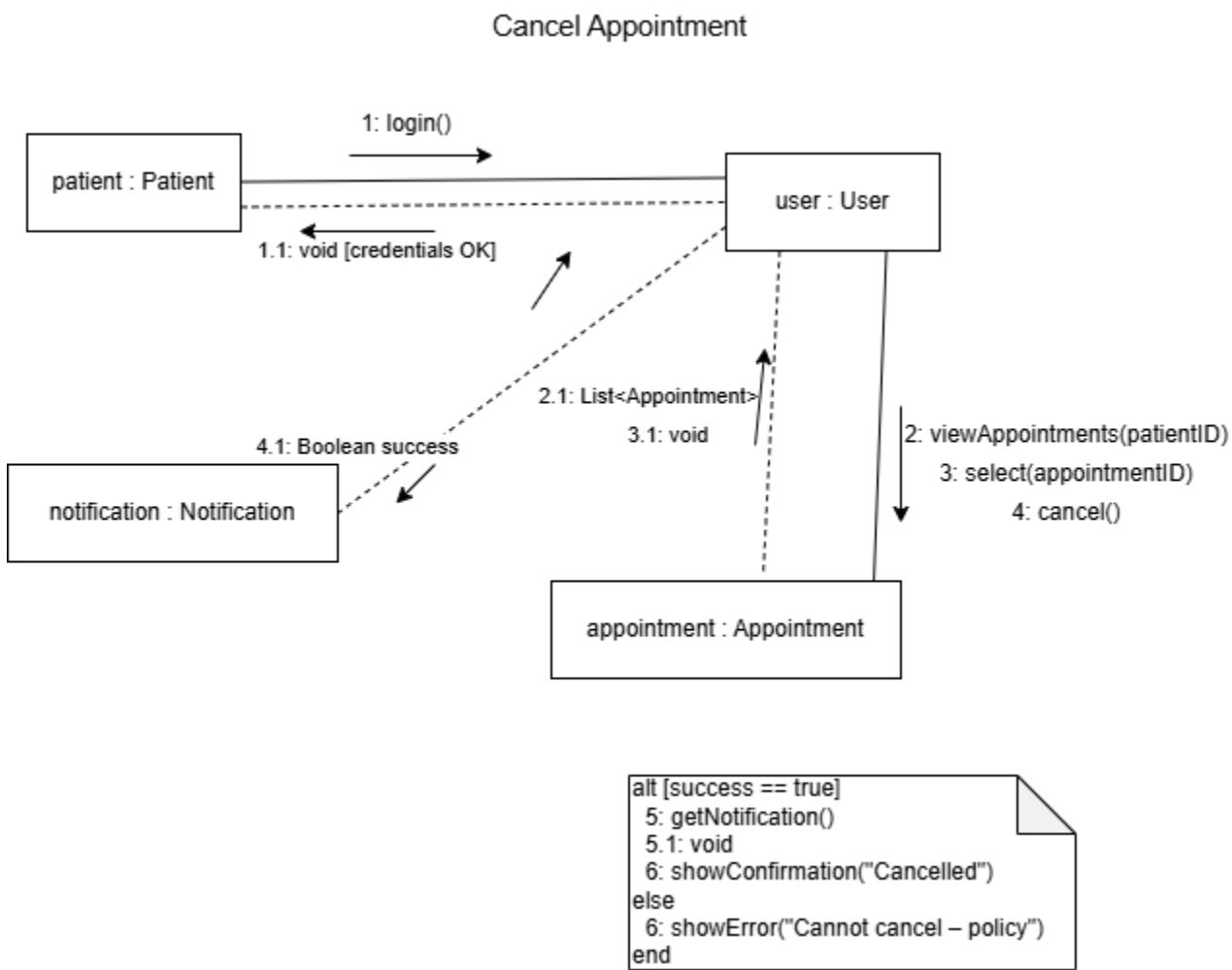
### CD\_PH\_02: Pharmacy Inventory Management



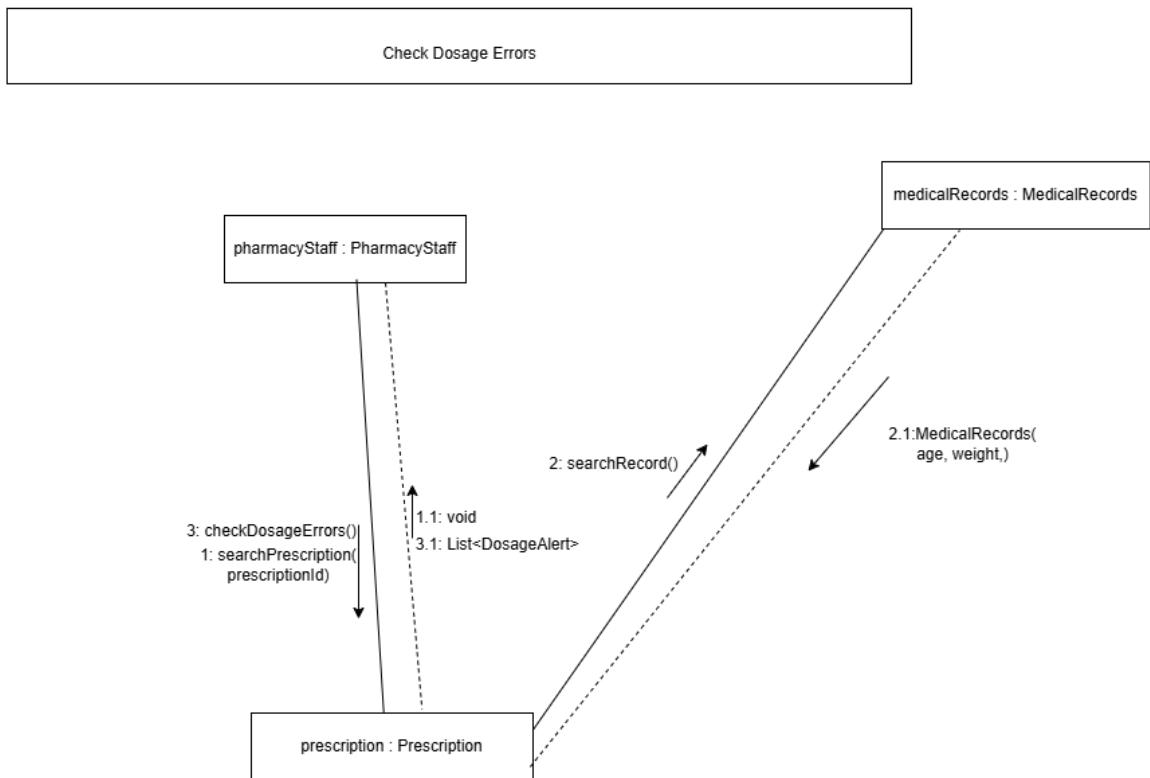
### CD\_PH\_03: Customer Loyalty & Discount Program



**Elisona Doku:**

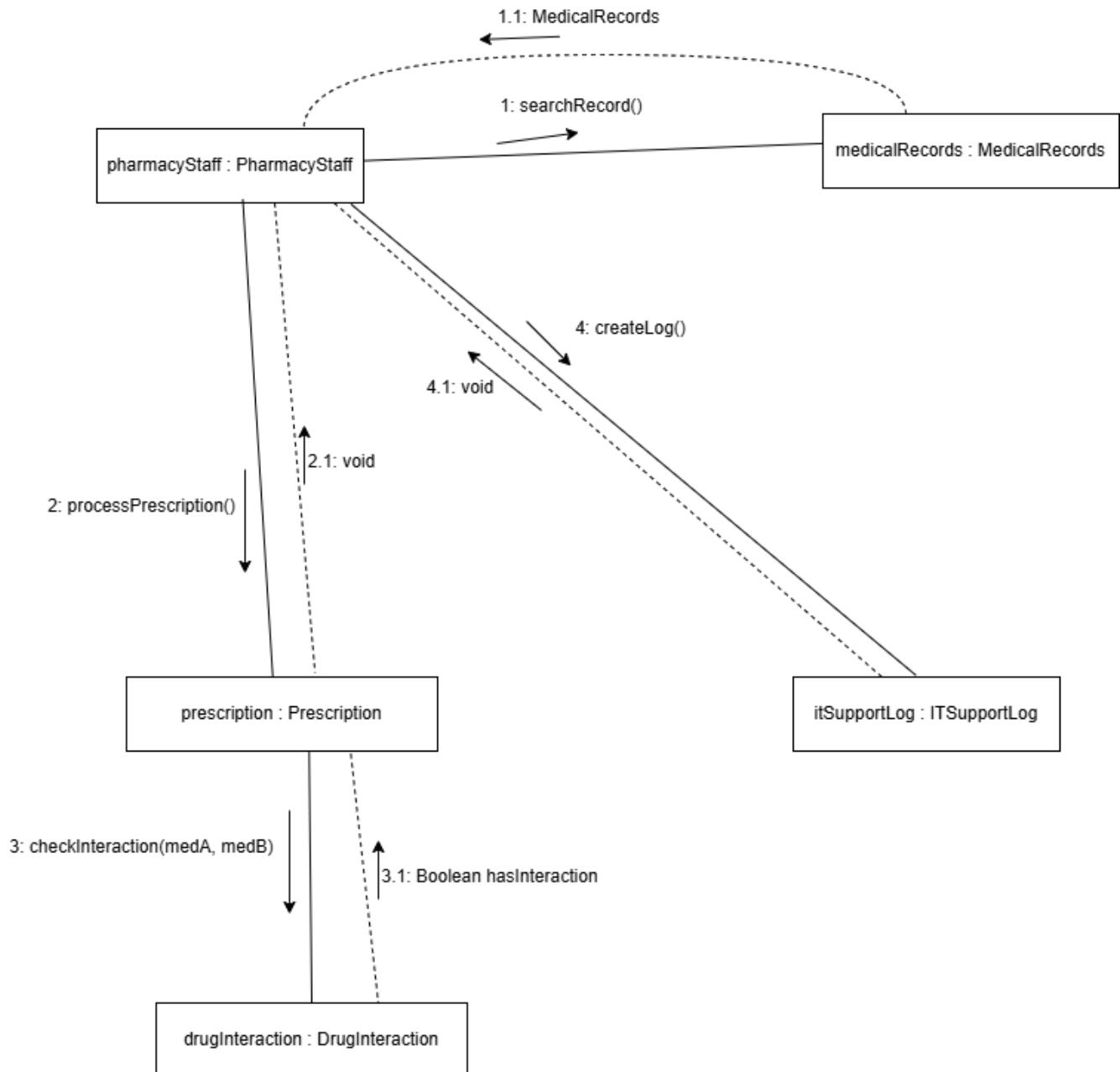


## **Electronic Healthcare System Requirements Specification**

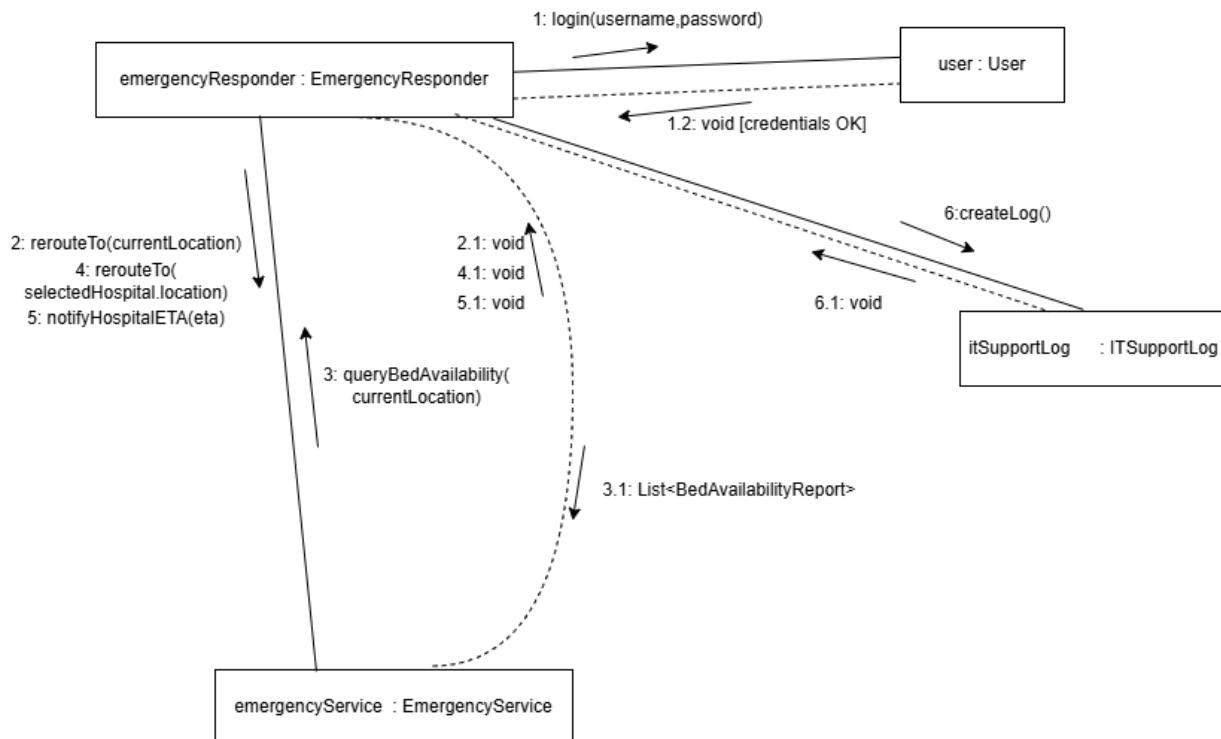


## Electronic Healthcare System Requirements Specification

### Check Drug Interactions



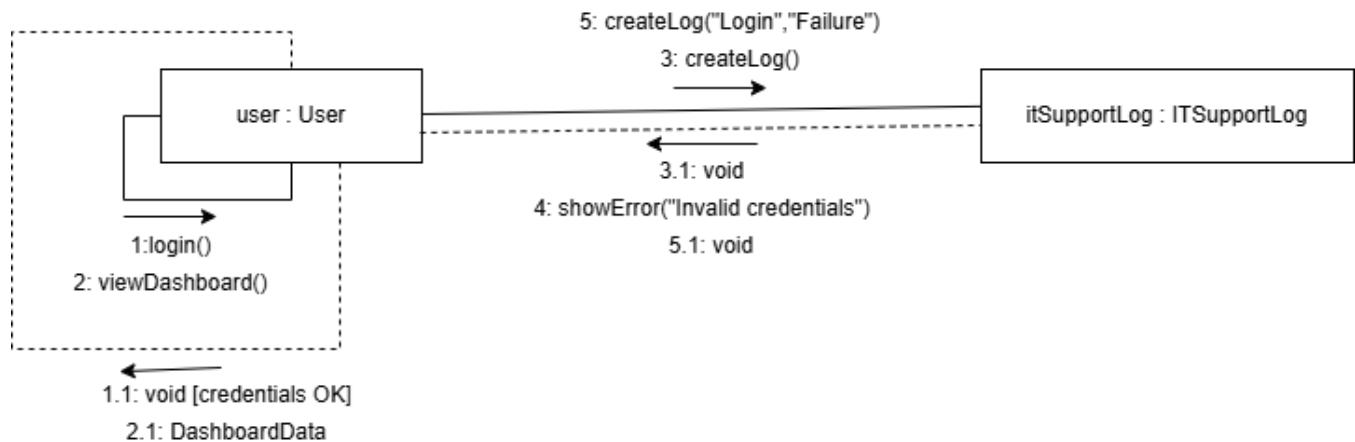
## Electronic Healthcare System Requirements Specification



Check Hospital Bed Availability

## *Electronic Healthcare System Requirements Specification*

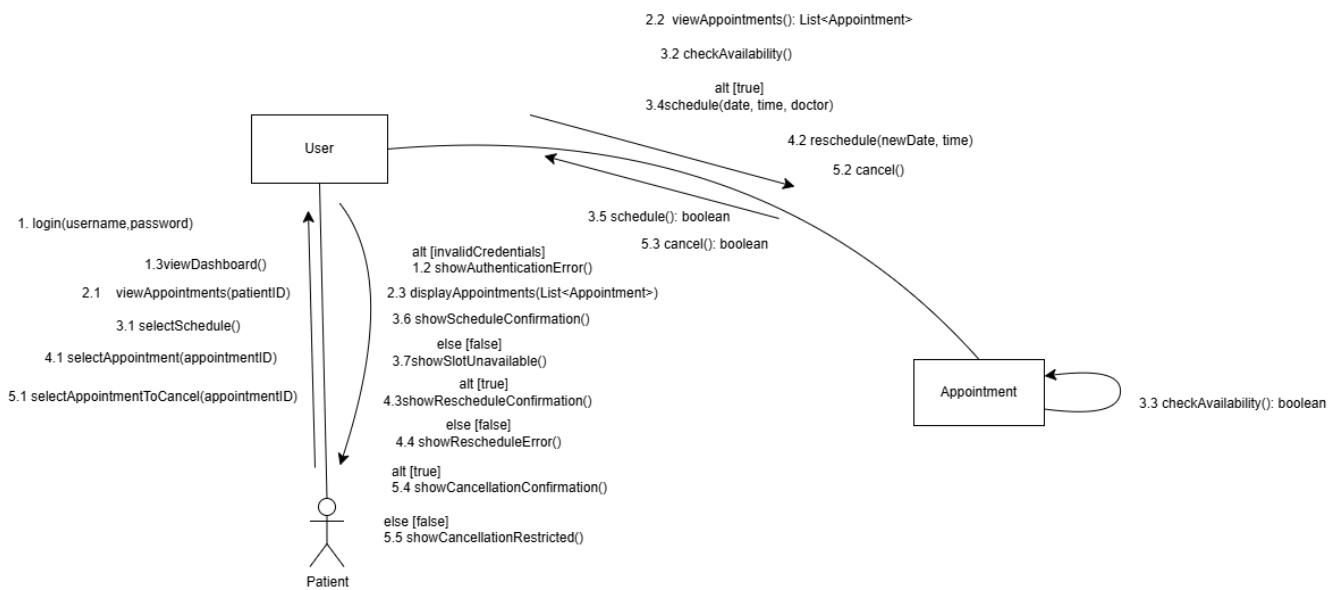
### User LogIn



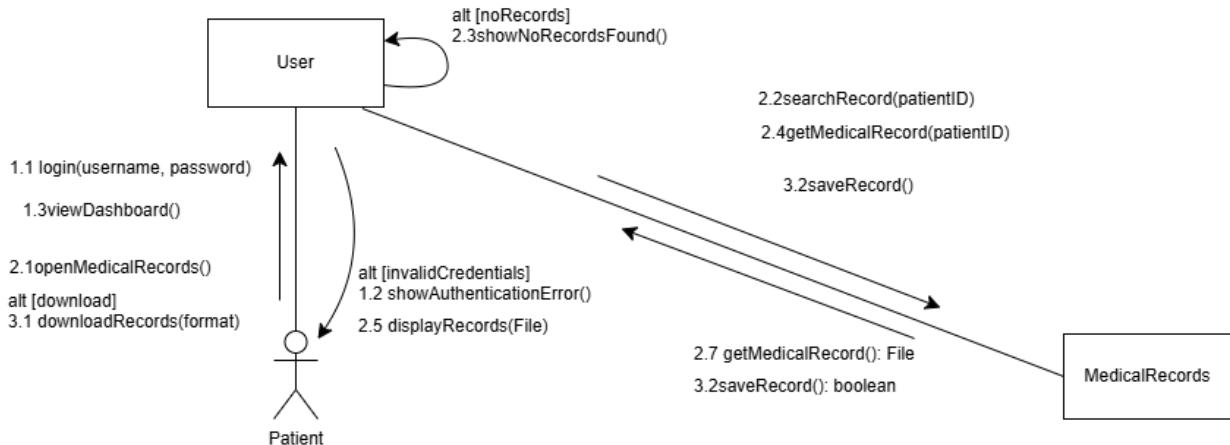
# Electronic Healthcare System Requirements Specification

Evelina Gace

## Patient Appointment Management

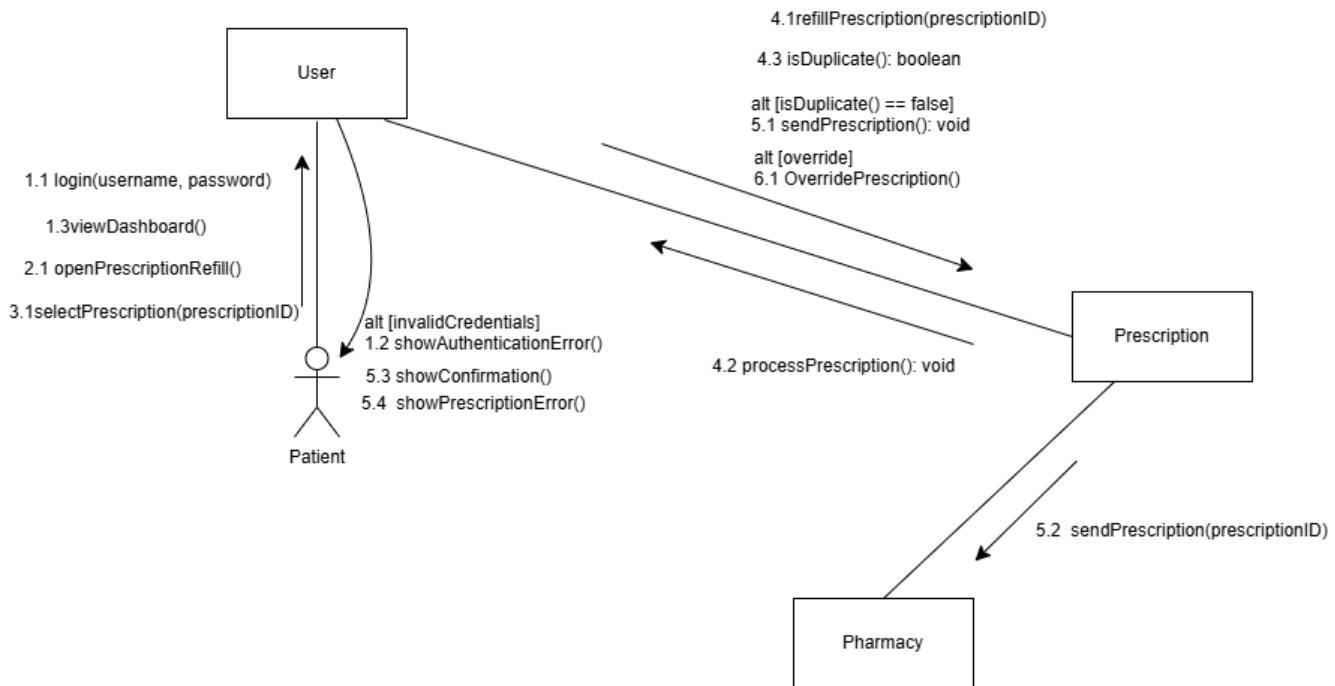


## Patient Medical Record Access

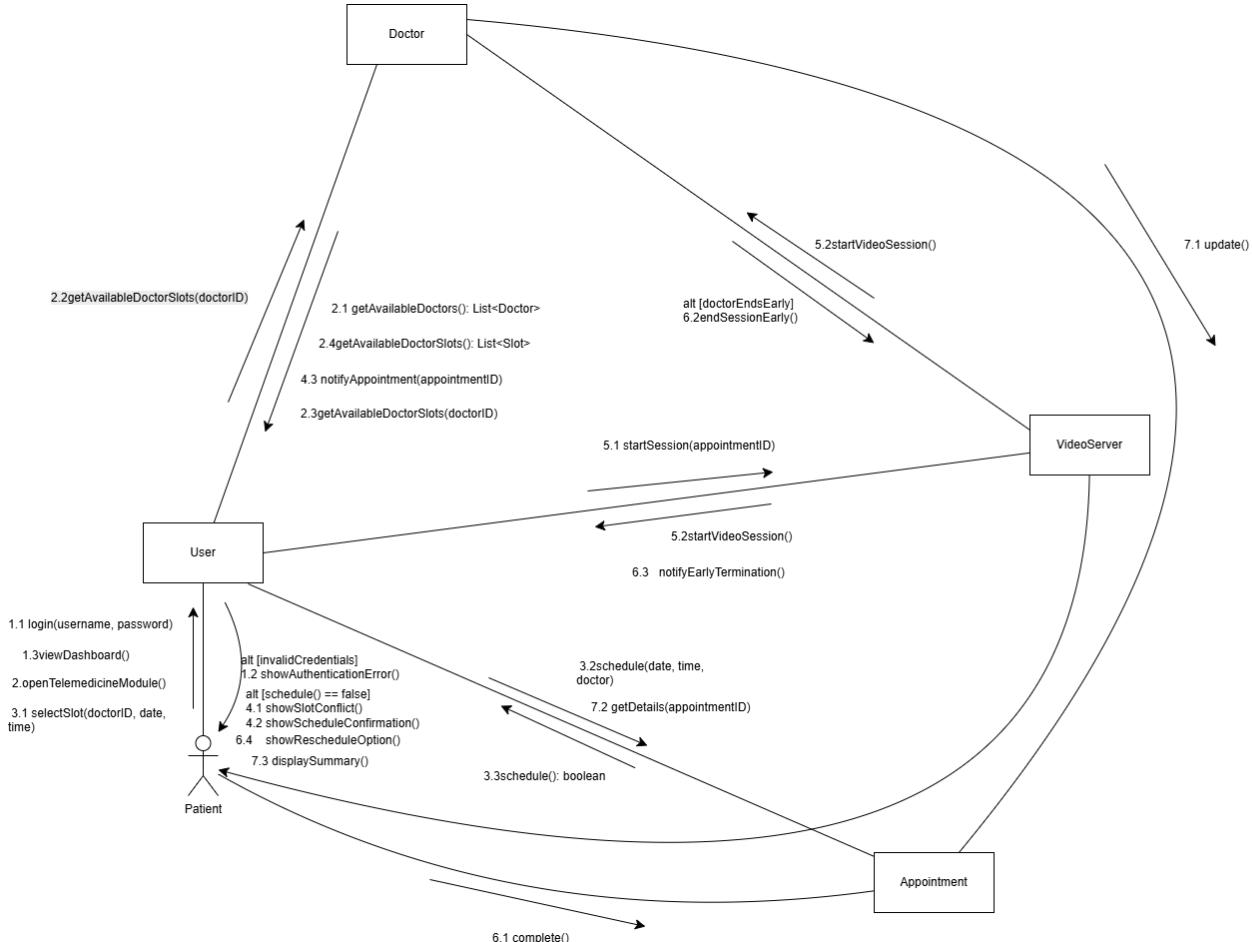


# Electronic Healthcare System Requirements Specification

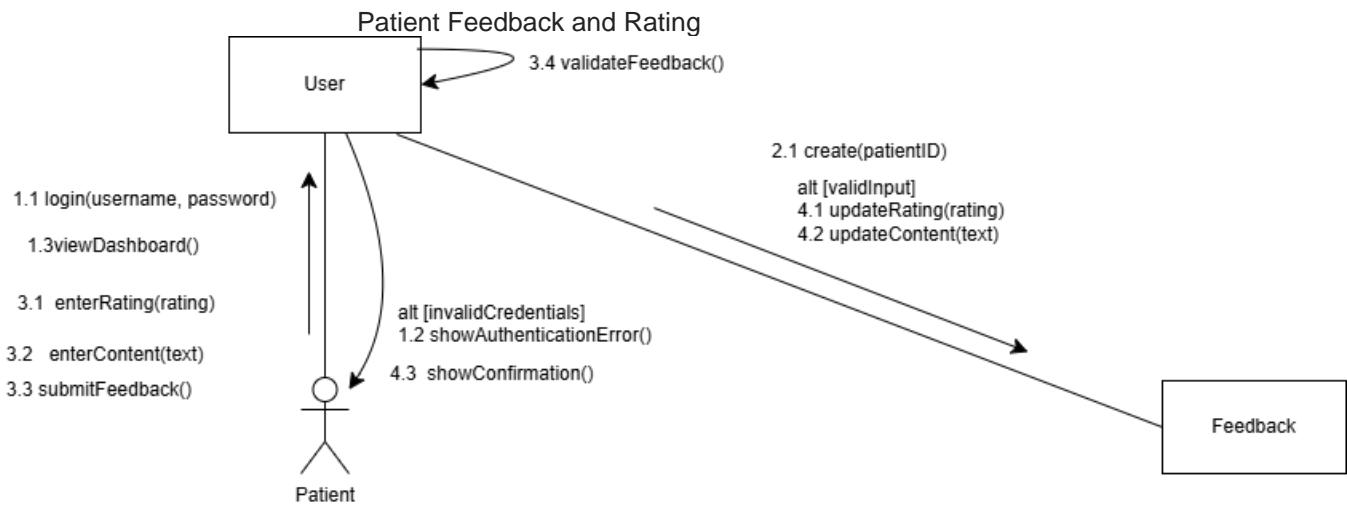
## Prescription Refill Management



## Telemedicine Consultation

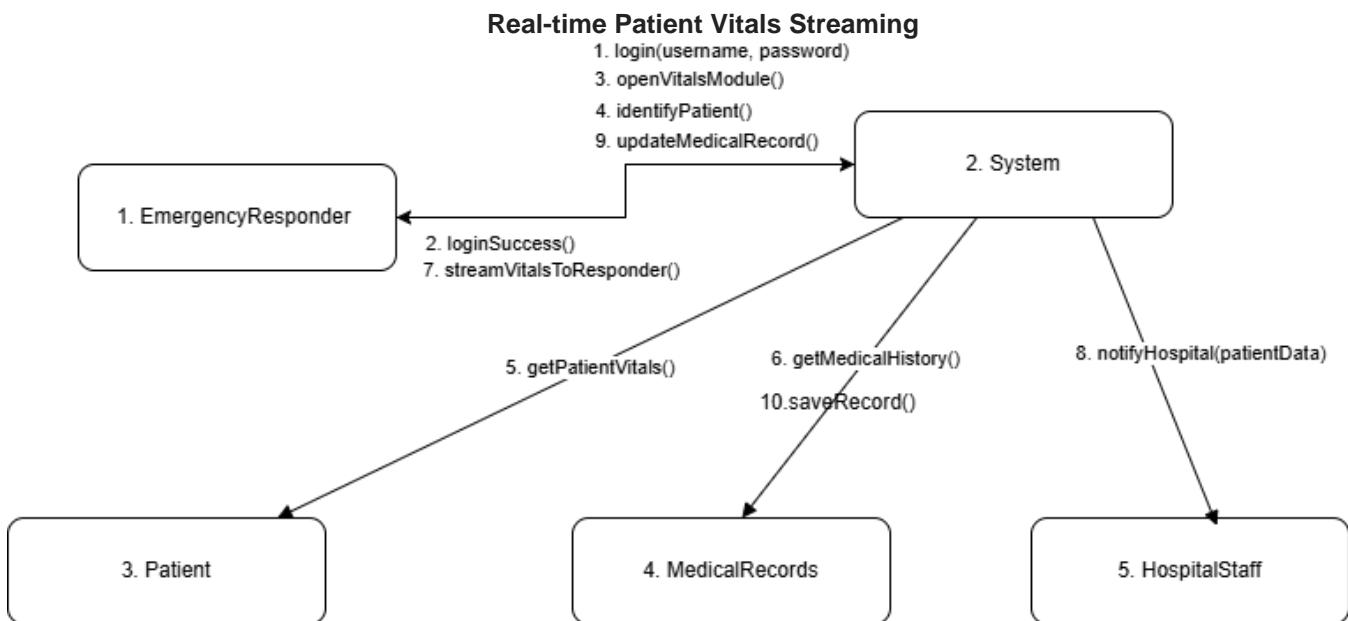
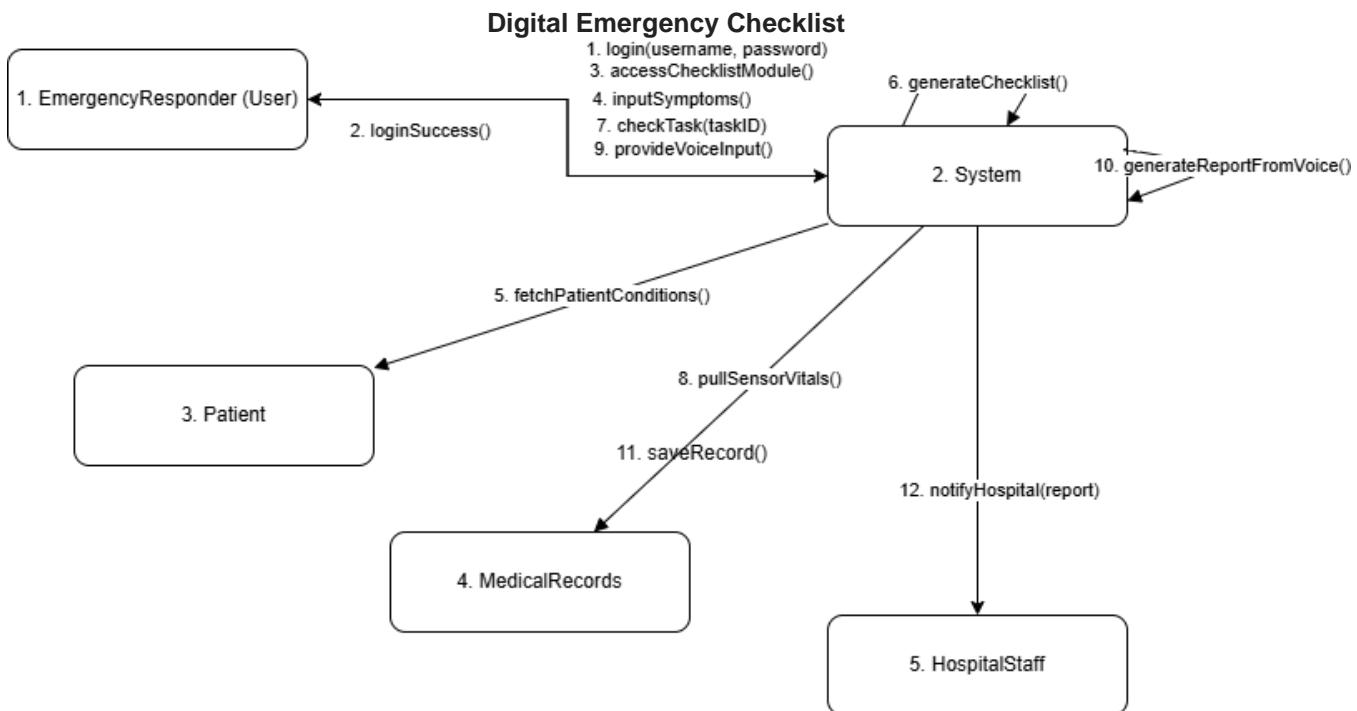


## **Electronic Healthcare System Requirements Specification**

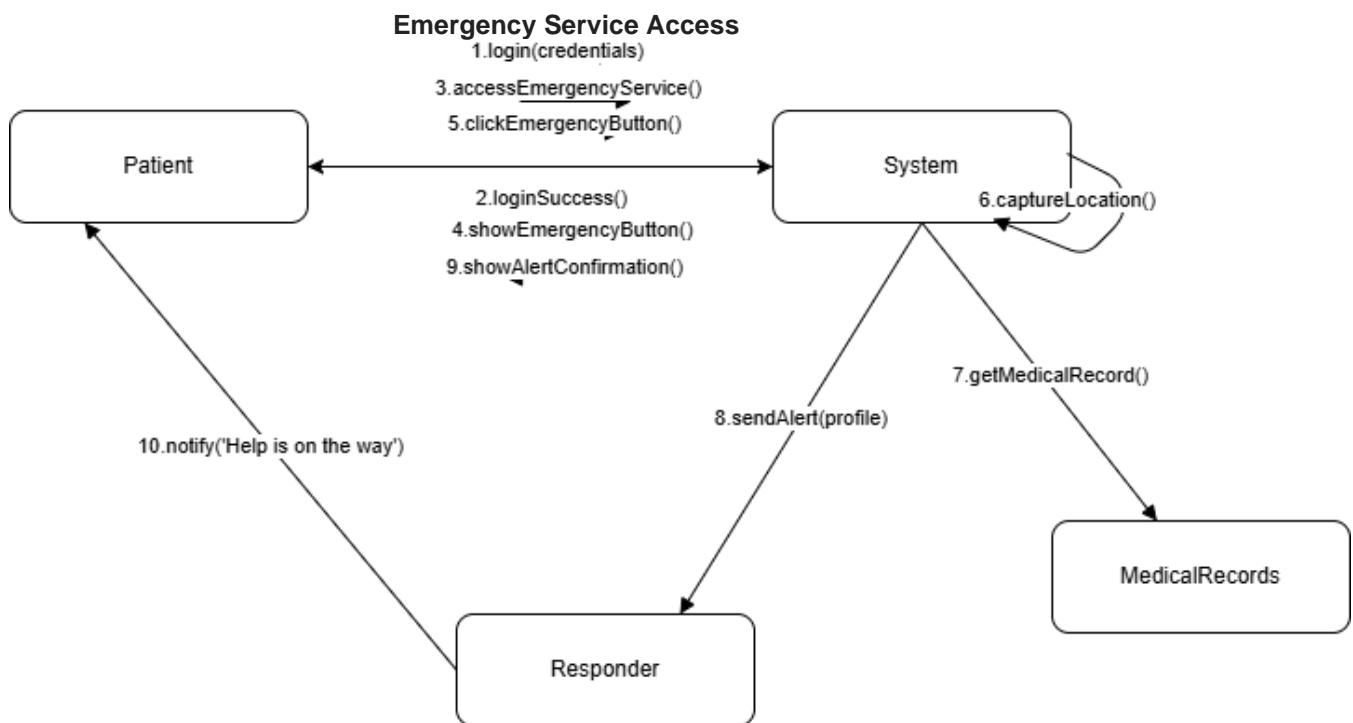


## Electronic Healthcare System Requirements Specification

**Elkier Ago:**

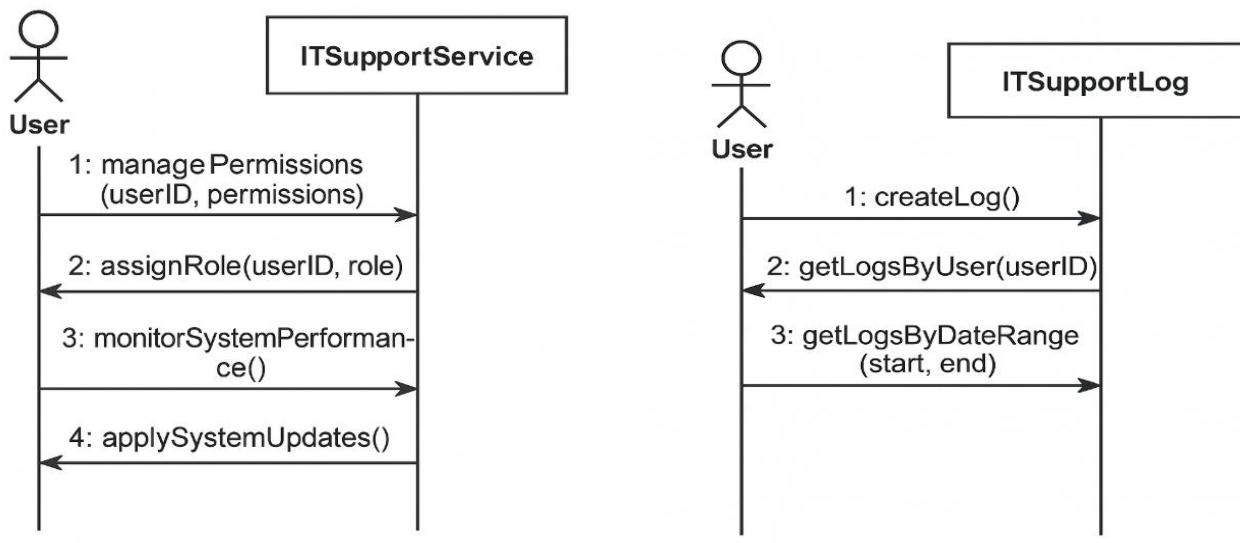


## **Electronic Healthcare System Requirements Specification**

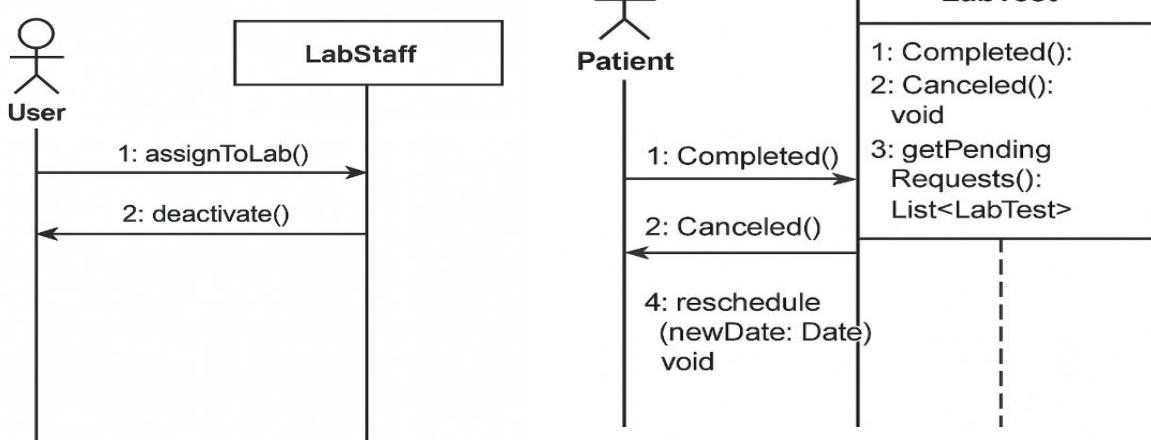


**Liza Koliqi:  
IT Support Service**

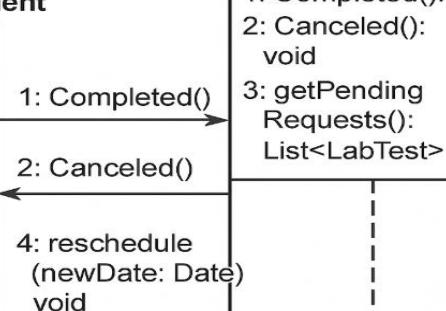
**IT Support**



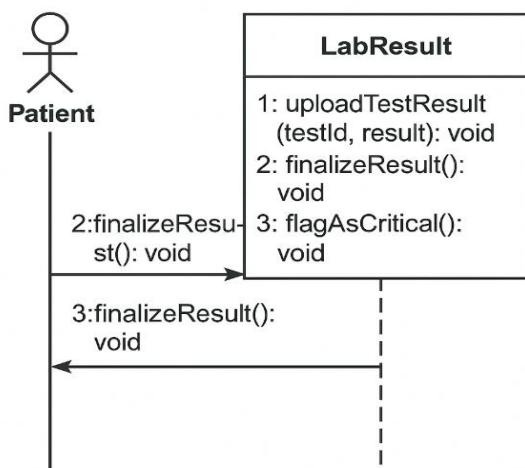
Lab Staff



Patient



Lab Result



## 6.Design Patterns

### 6.1 Abstract Factory-Antea Koxherri & Flavia Koco

The class diagram represents a diagnostic testing module within a healthcare system. This module is responsible for creating and processing two types of diagnostic services:

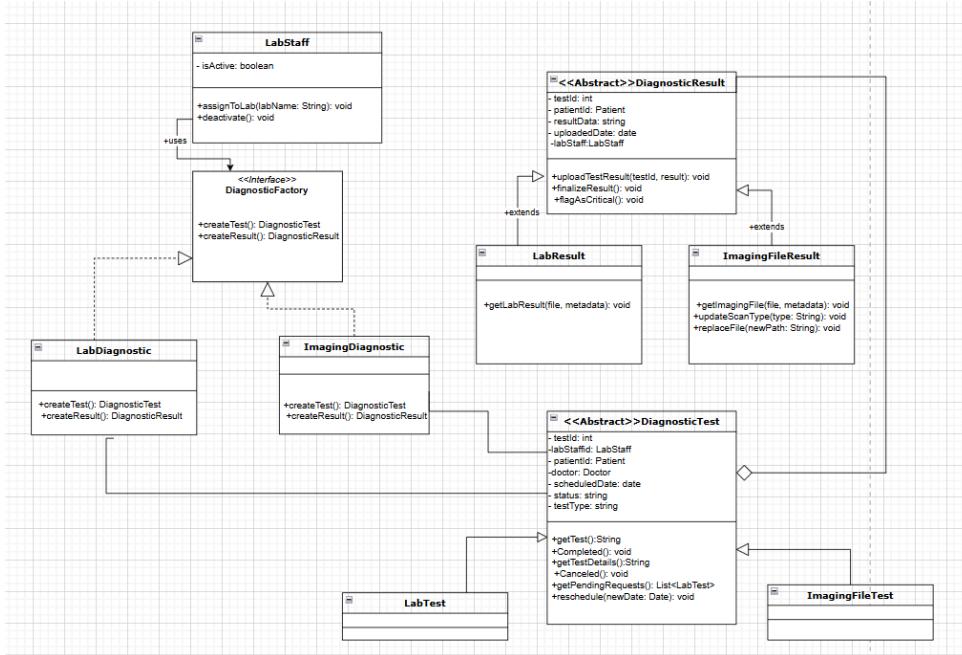
- **Lab Diagnostics** (e.g., blood tests, urinalysis)
- **Imaging Diagnostics** (e.g., X-rays, MRIs)

These services differ in how they produce and handle test results, yet share common behaviors and structures. To manage this complexity and enforce consistency, the **Abstract Factory Pattern** is used.

## Why Abstract Factory?

The **Abstract Factory Pattern** is used here to:

1. **Encapsulate the creation logic** of diagnostic tests and results based on diagnostic type (Lab or Imaging).
2. **Separate system concerns** — the **LabStaff** class only knows how to use a **DiagnosticFactory**, not the details of each diagnostic type.
3. **Promote consistency** — each factory ensures the right type of test/result pair is created.
4. **Make the system scalable** — new diagnostic types (e.g., **GeneticTestingDiagnostic**) can be added with minimal changes by introducing new factories.
5. **Reduce tight coupling** — **LabStaff** doesn't need to know the exact concrete classes like **LabTest** or **ImagingFileTest**. It works through abstraction.



## **6.2 Prototype Pattern- Belina Durmishi**

Justification for Using the Prototype Design Pattern in Selected Classes

The Prototype Design Pattern allows for efficient object creation by duplicating existing instances rather than constructing them from scratch. This is particularly beneficial in systems where object configuration is repetitive, resource-intensive, or involves minimal variation across instances. The following classes within the healthcare system architecture are well-suited for applying this pattern:

### **1. Report**

Rationale: Reports maintain a consistent format across different patients and time intervals. Often, generating a new report involves reusing the structure of a previous report with minimal changes.

Example Scenario: A clinician generates a new monthly progress report for a patient by duplicating a previous report and modifying fields such as patientId, dateGenerated, and progressScore.

### **2. Consultation**

Rationale: Consultations, especially recurring or follow-up sessions, typically involve similar content and structure with minor updates.

Example Scenario: A nutritionist creates a follow-up consultation by cloning an earlier session record, updating the date, and appending new notes.

### **3. EducationalMaterial**

Rationale: Educational content is frequently reused across patients, with occasional modifications to tailor the material for specific needs.

Example Scenario: A standard dietary guide is cloned and personalized by a nutritionist before assigning it to a new patient.

### **4. Medication**

Rationale: Medications often share the same core properties (e.g., name, manufacturer) while differing in attributes like dosage or expiry.

Example Scenario: A previously defined medication is duplicated, and only the dosage and expiry date are updated for a new prescription.

### **5. DietaryRecord**

Rationale: Dietary records for patients often follow predictable and repetitive structures, especially in planned meal tracking.

Example Scenario: A nutritionist duplicates a dietary record from the previous week and updates it with new meal entries and revised BMI values.

### **6. Meal**

## Electronic Healthcare System Requirements Specification

Rationale: Standardized meals (e.g., heart-healthy breakfast) are reused across patient plans, typically with slight modifications.

Example Scenario: A nutritionist clones a template meal and adjusts its contents by substituting one or two food items.

### 7. FoodItem

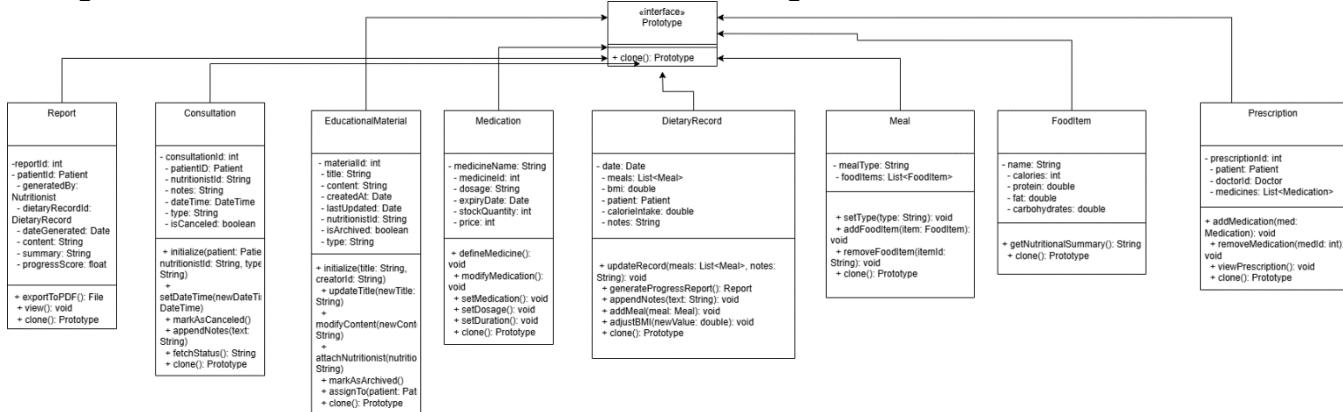
Rationale: Nutritional attributes of food items such as calories, protein, or fat remain constant, making them ideal for reuse through cloning.

Example Scenario: A common food item like “boiled egg” is cloned and included in multiple different meals without re-entering its nutritional data.

### 8. Prescription

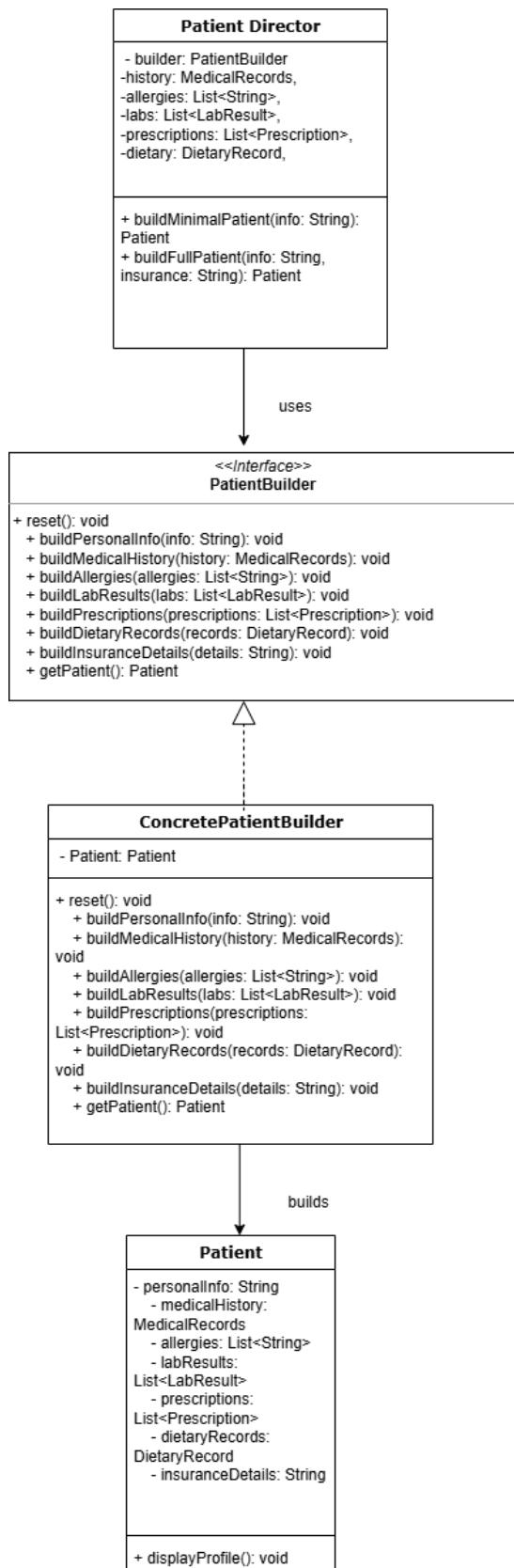
Rationale: Prescriptions for chronic conditions often consist of recurring medications with occasional adjustments.

Example Scenario: A doctor replicates a patient’s previous prescription and modifies the dosage of one medication to reflect recent clinical changes.



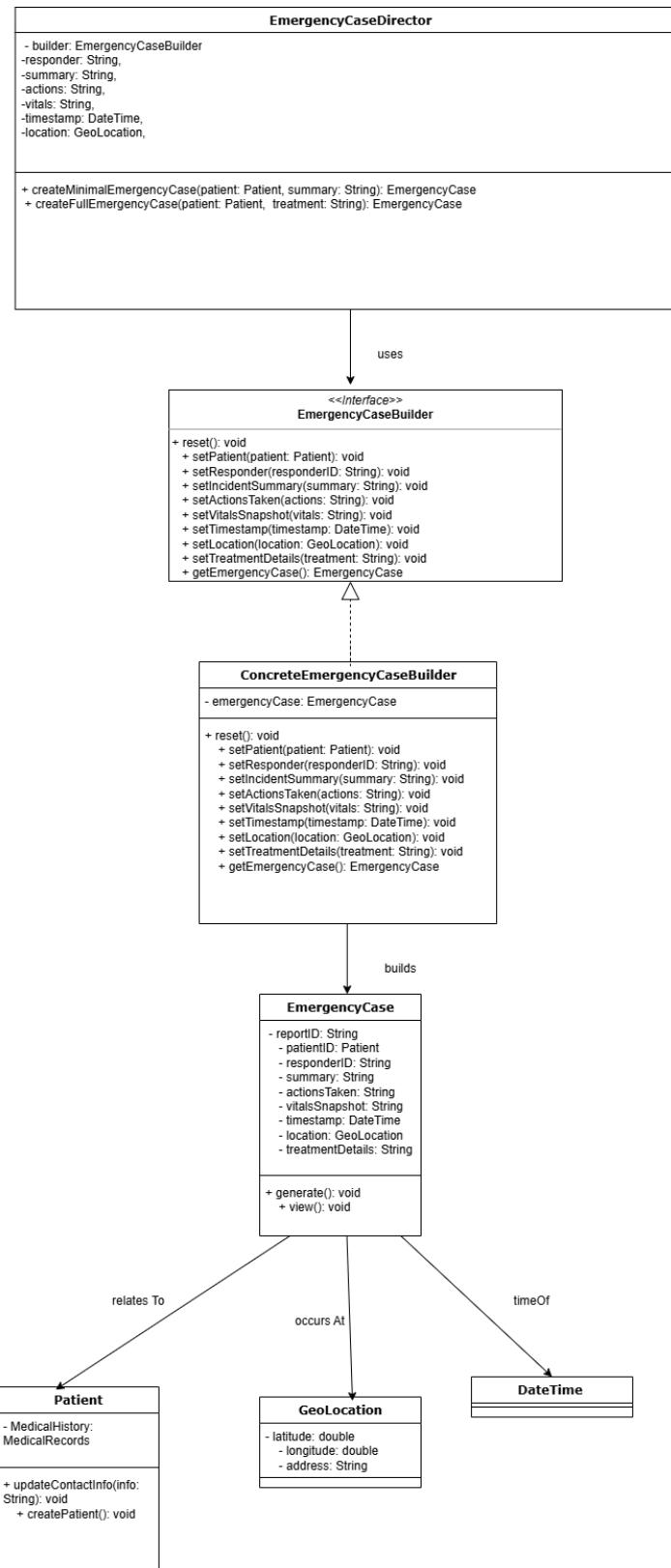
### 6.3 Builder Pattern

Evelina Gace -Patient Builder Pattern



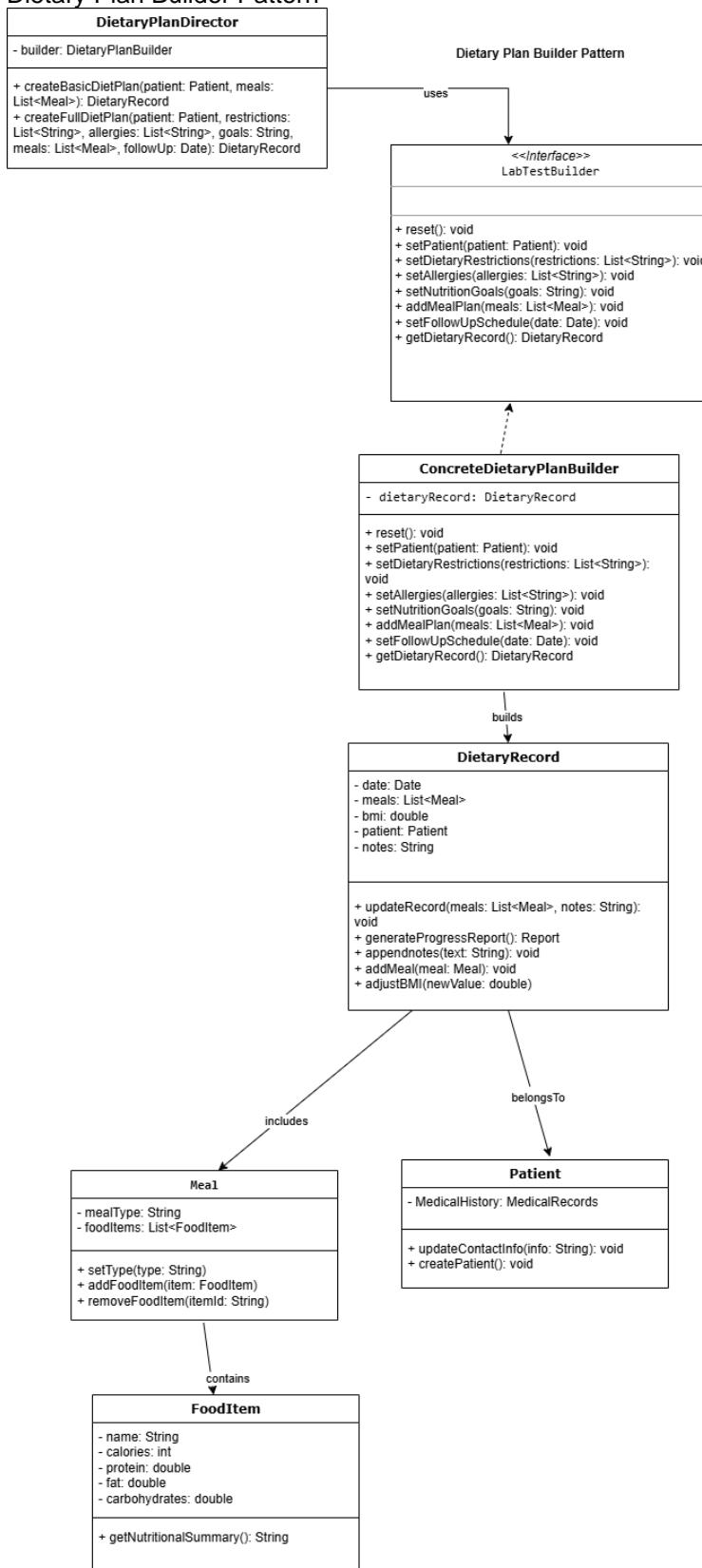
## Electronic Healthcare System Requirements Specification

### Emergency Case Report Builder Pattern



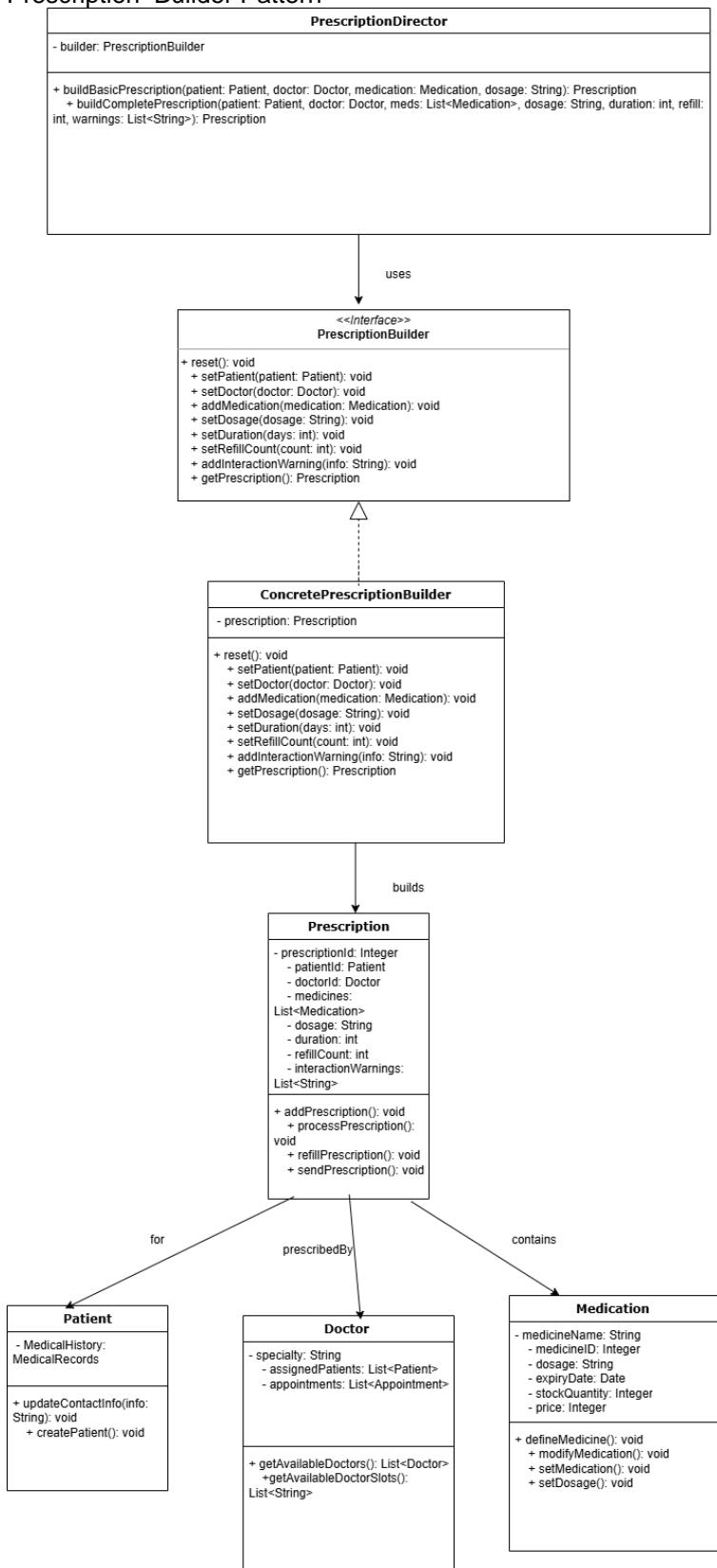
## Electronic Healthcare System Requirements Specification

### Dietary Plan Builder Pattern



## Electronic Healthcare System Requirements Specification

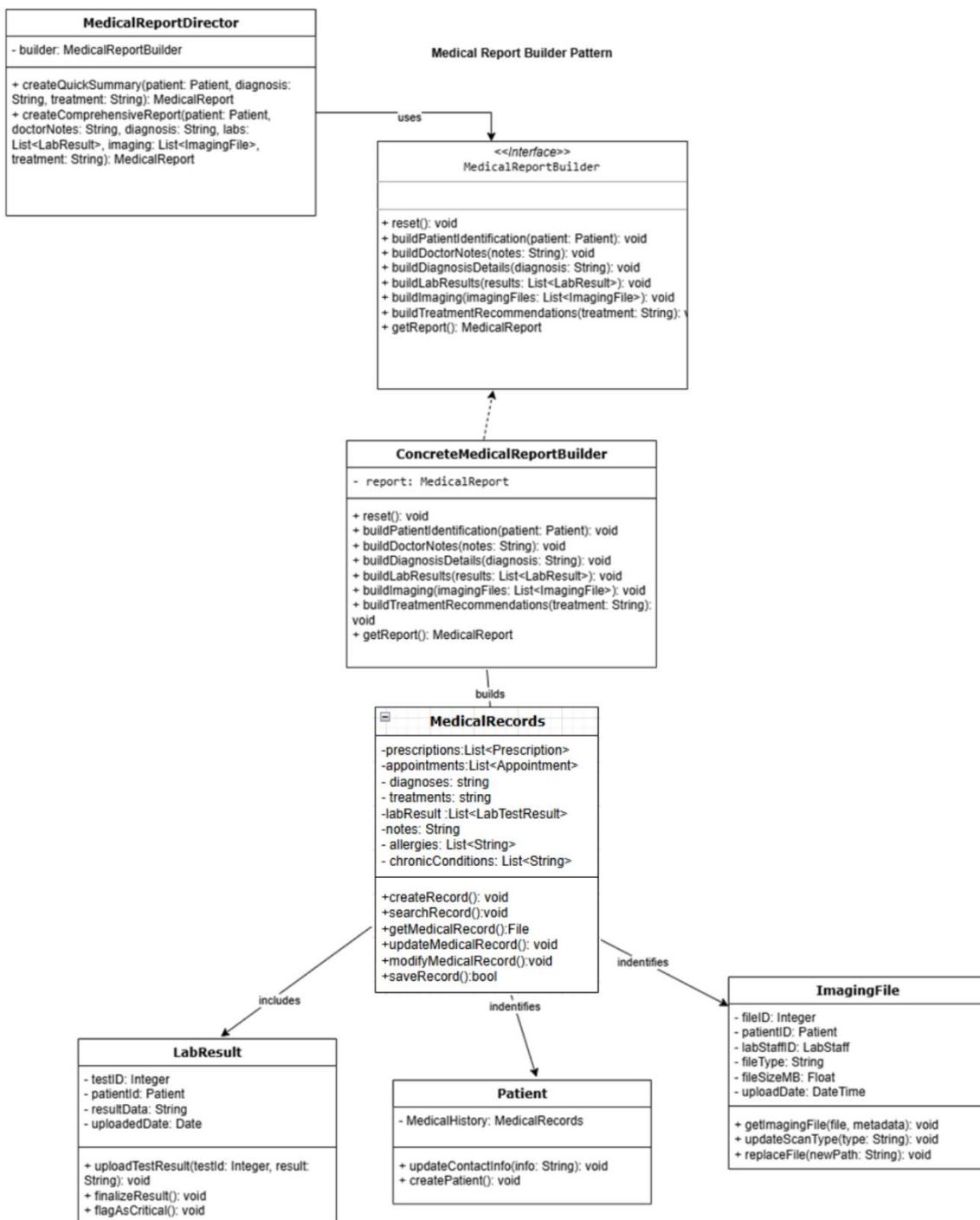
### Prescription Builder Pattern



## Electronic Healthcare System Requirements Specification

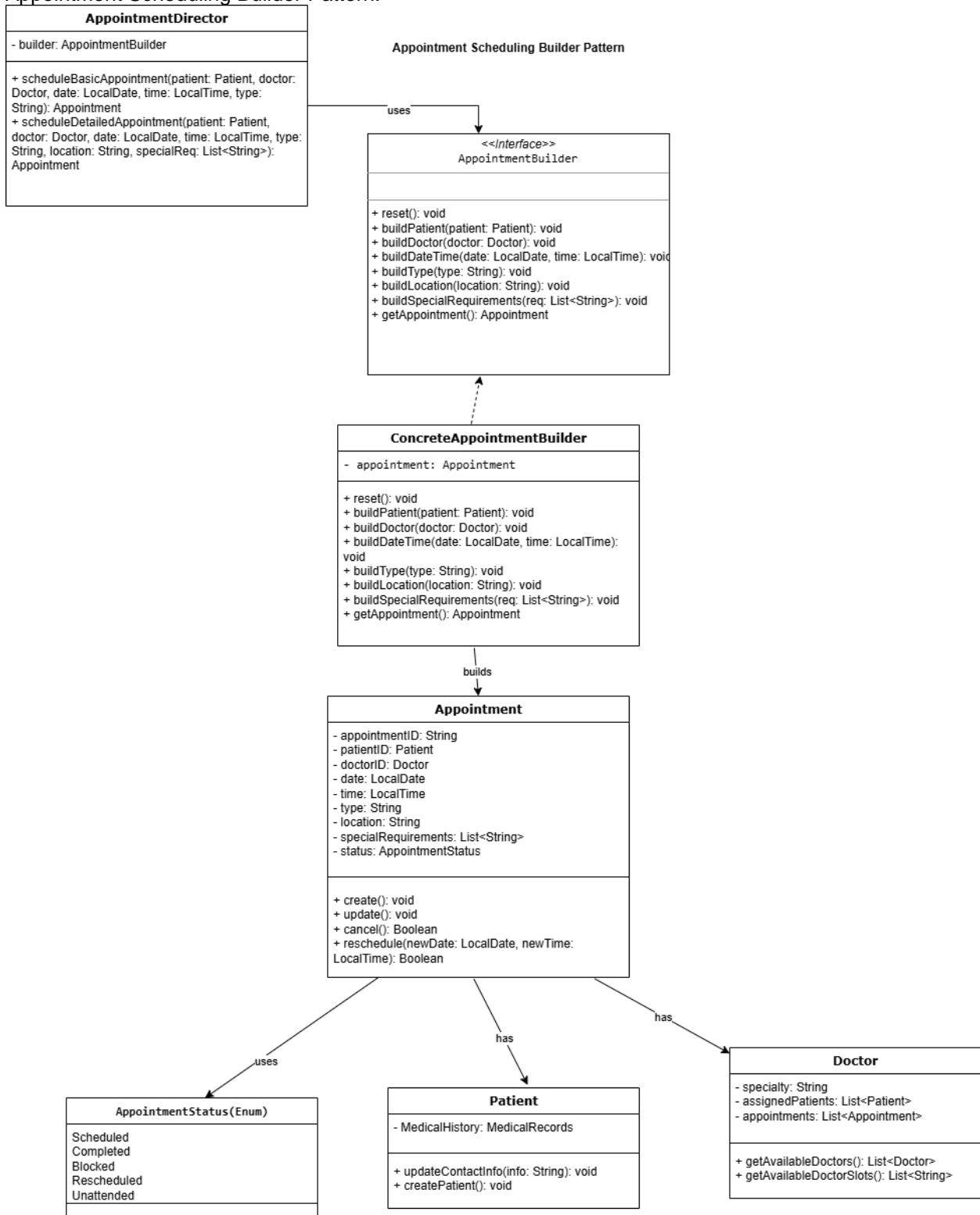
### Elisona Doku

Medical Report Builder Pattern:



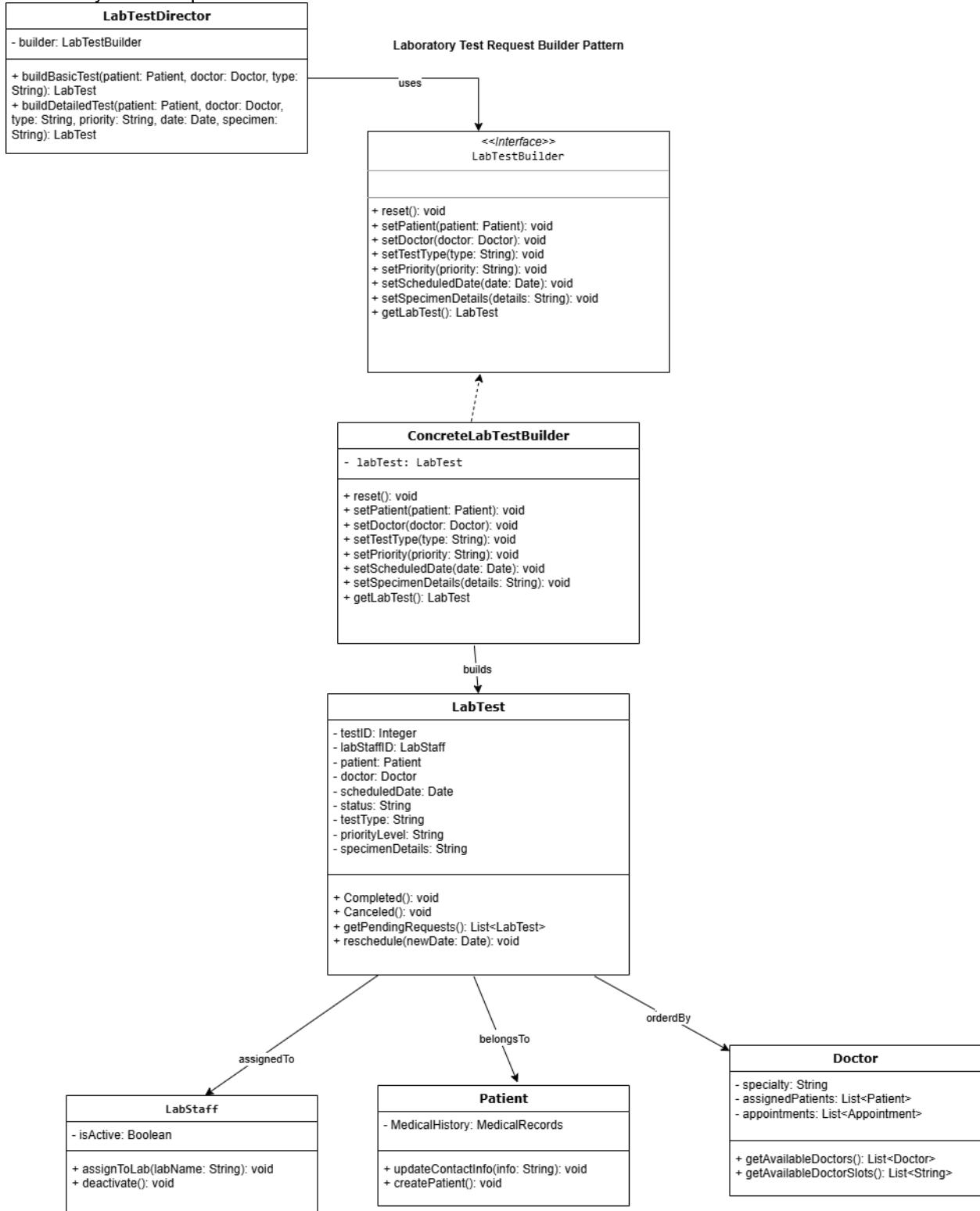
## Electronic Healthcare System Requirements Specification

### Appointment Scheduling Builder Pattern:



## Electronic Healthcare System Requirements Specification

### Laboratory Test Request Builder Pattern:



## **6.4 Factory Method**

**Elkier Ago:**

# Electronic Healthcare System Requirements Specification

## 6.5 Singleton Liza Koliqi:

<p><b>PatientRecordManager</b></p> <ul style="list-style-type: none"> <li>- instance: PatientRecordManager (private static)</li> <li>- PatientRecordManager() (private constructor)</li> </ul> <p>+ getInstance(): PatientRecordManager</p> <p>+ getPatientRecord(patientId): PatientRecord</p> <p>+ updatePatientRecord(patientId, data): void</p> <p>+ createPatientRecord(data): void</p> <p>+ validateRecord(data): boolean</p>	<p><b>Doctor</b></p> <p><b>ElectronicPrescriptionManager</b></p> <ul style="list-style-type: none"> <li>- instance: ElectronicPrescriptionManager (private static)</li> <li>- ElectronicPrescriptionManager()</li> </ul> <p>+ getInstance(): ElectronicPrescriptionManager</p> <p>+ createPrescription(patientId, medicationData): void</p> <p>+ checkDrugInteractions(patientId, medicationData): InteractionResult</p> <p>+ sendToPharmacy(prescription): void</p> <p>+ notifyPatient(prescriptionStatus): void</p> <p>+ markAsDispensed(prescriptionId): void</p>	<p><b>TimetableManager</b></p> <ul style="list-style-type: none"> <li>- instance: TimetableManager (private static)</li> <li>- TimetableManager() (private constructor)</li> </ul> <p>+ getInstance(): TimetableManager</p> <p>+ addAppointment(doctorId, appointmentData): void</p> <p>+ rescheduleAppointment(appointmentId, newTime): boolean</p> <p>+ blockTimeSlot(doctorId, timeSlot): void</p> <p>+ validateSlotAvailability(doctorId, timeSlot): boolean</p> <p>+ notifyPatient(appointmentId, changeType): void</p>	
<p><b>Organ Donor Coordinator</b></p> <p><b>OrganDonorRegistryManager</b></p> <ul style="list-style-type: none"> <li>- instance: OrganDonorRegistryManager (private static)</li> <li>- OrganDonorRegistryManager() (private constructor)</li> </ul> <p>+ getInstance(): OrganDonorRegistryManager</p> <p>+ registerDonor(personalData, medicalData): boolean</p> <p>+ validateDonorData(data): boolean</p> <p>+ generateMatchDetails(MatchResult): void</p> <p>+ notifyCoordinator(matchDetails): void</p> <p>+ getDonorById(donorId): Donor</p>	<p><b>OrganReportManager</b></p> <ul style="list-style-type: none"> <li>- instance: OrganReportManager (private static)</li> <li>- OrganReportManager() (private constructor)</li> </ul> <p>+ getInstance(): OrganReportManager</p> <p>+ generateReport(type, timePeriod): Report</p> <p>+ retrieveData(type, timePeriod): ReportData</p> <p>+ exportReport(type, fileFormatType): File</p> <p>+ exportReport(report, destination): void</p> <p>+ logError(error): void</p> <p>+ scheduleEmailDelivery(report, email): void</p>	<p><b>DietaryRecordManager</b></p> <ul style="list-style-type: none"> <li>- instance: DietaryRecordManager (private static)</li> <li>- DietaryRecordManager()</li> </ul> <p>+ getInstance(): DietaryRecordManager</p> <p>+ getPatientDietaryRecord(patientId): DietaryRecord</p> <p>+ createDietaryRecord(patientId, data): void</p> <p>+ updateDietaryRecord(patientId, data): void</p> <p>+ generateProgressReport(patientId): void</p>	
<p><b>Pharmacy Staff</b></p> <p><b>PrescriptionManager</b></p> <ul style="list-style-type: none"> <li>- instance: PrescriptionManager (private static)</li> <li>- PrescriptionManager() (private constructor)</li> </ul> <p>+ getInstance(): PrescriptionManager</p> <p>+ recordPurchase(prescription: Prescription): void</p> <p>+ verifyPrescription(prescription: Prescription): boolean</p> <p>+ checkInteractions(prescription: Prescription): List&lt;Alert&gt;</p> <p>+ checkDuplicates(prescription): boolean</p> <p>+ fulfillPrescription(prescriptionId): void</p> <p>+ notifyPatient(prescriptionId): void</p> <p>+ requestStockVerification(prescriptionId): void</p> <p>+ logPrescriptionActivity(prescriptionId, activity): void</p>	<p><b>InventoryManager</b></p> <ul style="list-style-type: none"> <li>- static instance: InventoryManager</li> <li>- inventoryData: Map&lt;String, Medication&gt;</li> <li>- constructor() [private]</li> </ul> <p>+ getInstance(): InventoryManager</p> <p>+ updateStock(medId: String, qty: int): void</p> <p>+ correctStock(medId: String, qty: int): void</p> <p>+ checkForOrder(medId: String): void</p> <p>+ removeExpired(medId: String): void</p> <p>+ generateReport(): InventoryReport</p>	<p><b>ConsultationManager</b></p> <ul style="list-style-type: none"> <li>- instance: ConsultationManager (private static)</li> <li>- ConsultationManager()</li> </ul> <p>+ static instance: ConsultationManager</p> <p>+ updateConsultation(consultationId, newTimeSlot): boolean</p> <p>+ cancelConsultation(consultationId): boolean</p> <p>+ recordConsultationNotes(consultationId, notes): void</p> <p>+ recordConsultationVotes(consultationId): void</p> <p>+ getAvailableTimeSlots(nutritionId): List&lt;TimeSlot&gt;</p> <p>+ notifyParticipants(consultationId): void</p> <p>+ logMissedConsultation(consultationId): void</p>	<p><b>Nutritionist</b></p> <p><b>CollaborationManager</b></p> <ul style="list-style-type: none"> <li>- instance: CollaborationManager (private static)</li> <li>- CollaborationManager()</li> </ul> <p>+ getInstance(): CollaborationManager</p> <p>+ sharePatientData(patientId, recipientId): boolean</p> <p>+ verifyAccessPermissions(nutritionId, recipientId): boolean</p> <p>+ getPatientDietaryRecord(patientId): DietaryRecord</p> <p>+ notifyRecipient(recipientId, patientId): void</p> <p>+ logActivity(action, userId, patientId): void</p>
<p><b>LoyaltyManager</b></p> <ul style="list-style-type: none"> <li>- static instance: LoyaltyManager</li> <li>- loyaltyDatabase: Map&lt;String, LoyaltyAccount&gt;</li> <li>- constructor() [private]</li> </ul> <p>+ getInstance(): LoyaltyManager</p> <p>+ earnPoints(patientId: String): void</p> <p>+ recordPurchase(patientId: String, Purchase): void</p> <p>+ calculateDiscount(patientId: String): double</p> <p>+ generateReceipt(patientId: String): Receipt</p> <p>+ generateReports(): LoyaltyReport</p> <p>+ optOut(patientId: String): void</p>	<p><b>LabTestManager</b></p> <ul style="list-style-type: none"> <li>- static instance: LabTestManager</li> <li>- testDatabase: Map&lt;String, LabTest&gt;</li> <li>- constructor() [private]</li> </ul> <p>+ getInstance(): LabTestManager</p> <p>+ viewPendingRequests(): List&lt;LabTest&gt;</p> <p>+ uploadResult(testId: String, result: String): void</p> <p>+ markPending(testId: String): void</p> <p>+ notifyUsers(testId: String): void</p>	<p><b>Laboratory Staff</b></p> <p><b>ImagingManager</b></p> <ul style="list-style-type: none"> <li>- static instance: ImagingManager</li> <li>- imagingStore: Map&lt;String, ImagingFile&gt;</li> <li>- constructor() [private]</li> </ul> <p>+ getInstance(): ImagingManager</p> <p>+ uploadImage(file: File, metadata: Metadata): boolean</p> <p>+ validateImage(file: File): boolean</p> <p>+ retryUpload(file: File): void</p> <p>+ encryptAndStore(file: File): void</p> <p>+ getImagesByPatient(patientId: String): List&lt;ImagingFile&gt;</p>	<p><b>Patient</b></p> <p><b>AppointmentManager</b></p> <ul style="list-style-type: none"> <li>- static instance: AppointmentManager</li> <li>- appointments: Map&lt;String, Appointment&gt;</li> <li>- cancellationPolicy: Duration</li> <li>- AppointmentManager() [private]</li> </ul> <p>+ getInstance(): AppointmentManager</p> <p>+ scheduleAppointment(patientId, providerId, time): boolean</p> <p>+ modifyAppointment(appointmentId, newTime): boolean</p> <p>+ cancelAppointment(appointmentId): boolean</p> <p>+ getAvailableSlots(providerId, date): List&lt;TimeSlot&gt;</p> <p>+ getAppointments(patientId): List&lt;Appointment&gt;</p> <p>+ notify(patientId, message): void</p>
<p><b>TelemedicineSessionManager</b></p> <ul style="list-style-type: none"> <li>- static instance: TelemedicineSessionManager</li> <li>- sessionRepo: SessionRepository</li> <li>- scheduleService: ScheduleService</li> <li>- videoService: VideoCallService</li> <li>- notificationService: NotificationService</li> <li>- auditLogger: AuditLogger</li> </ul> <p>+ getInstance(): TelemedicineSessionManager</p> <p>+ scheduleConsultation(patientId, doctorId, slot): boolean</p> <p>+ startSession(sessionId): void</p> <p>+ endSession(sessionId): void</p> <p>+ handleConnectionLoss(sessionId, actor): void</p> <p>+ notify(actorId, message): void</p> <p>+ logSession(sessionId, details): void</p>	<p><b>FeedbackManager</b></p> <ul style="list-style-type: none"> <li>- instance: FeedbackManager &lt;&gt; static &gt;</li> <li>- feedbackStorage: List&lt;Feedback&gt;</li> </ul> <p>+ getInstance(): FeedbackManager &lt;&gt; static &gt;</p> <p>+ submitFeedback(patient: Patient, service: Service, rating: int, comment: String): void</p> <p>+ validateFeedback(rating: int, comment: String): void</p> <p>+ getFeedbackReports(): List&lt;FeedbackReport&gt;</p>	<p><b>EmergencyAlertManager</b></p> <ul style="list-style-type: none"> <li>- instance: EmergencyAlertManager</li> <li>- alertLog: List&lt;EmergencyAlert&gt;</li> </ul> <p>+ getInstance(): EmergencyAlertManager</p> <p>+ submitFeedback(patient: Patient, service: Service, rating: int, comment: String): void</p> <p>+ relayAlert(alert: EmergencyAlert): void</p> <p>+ getAlertStatus(alertId: String): AlertStatus</p> <p>+ logAlert(alert: EmergencyAlert): void</p>	<p><b>IT Support</b></p> <p><b>UserAccountManager</b></p> <ul style="list-style-type: none"> <li>- instance: UserAccountManager</li> <li>- userAccounts: Map&lt;String, UserAccount&gt;</li> <li>- auditLog: List&lt;AccountChangeLog&gt;</li> </ul> <p>+ getInstance(): UserAccountManager</p> <p>+ createAccount(user: UserAccount): boolean</p> <p>+ modifyAccount(userId: String, updatedUser: UserAccount): boolean</p> <p>+ deactivateAccount(userId: String): boolean</p> <p>+ getAccount(userId: String): UserAccount</p> <p>+ logChange(change: AccountChangeLog): void</p> <p>+ getAuditLogs(): List&lt;AccountChangeLog&gt;</p>
<p><b>PatientVitalsStreamingManager</b></p> <ul style="list-style-type: none"> <li>- instance: PatientVitalsStreamingManager</li> <li>- activeConnections: Map&lt;PatientID, ConnectionInfo&gt;</li> <li>- gpsTracker: GPSTracker</li> <li>- notificationService: NotificationService</li> </ul> <p>+ getInstance(): PatientVitalsStreamingManager</p> <p>+ loginResponder(credentials): bool</p> <p>+ selectPatient(patientId: String): boolean</p> <p>+ connectToVariable(deviceId: String): boolean</p> <p>+ storeVitalsData(data: VitalsData): void</p> <p>+ receiveVitalsData(patientId: String): void</p> <p>+ updateETA(patientId: String): void</p> <p>+ notifyHospitalStaff(patientId: String): void</p> <p>+ handleConnections(patientId: String): void</p> <p>+ storeVitalsData(patientId: String, data: VitalsData): void</p>	<p><b>EmergencyChecklistManager</b></p> <ul style="list-style-type: none"> <li>- instance: EmergencyChecklistManager</li> <li>- currentChecklist: Checklist</li> <li>- voiceRecognition: VoiceRecognitionService</li> <li>- sensorDataService: SensorDataService</li> <li>- reportService: ReportGenerationService</li> </ul> <p>+ getInstance(): EmergencyChecklistManager</p> <p>+ loginResponder(credentials): bool</p> <p>+ accessRouter(credentials): bool</p> <p>+ getChecklistStep(): ChecklistStep</p> <p>+ completeStep(stepId: int): void</p> <p>+ updateChecklistWithSensorData(sensorData: SensorData): void</p> <p>+ receiveVoiceInput(audioInput: AudioData): void</p> <p>+ generateReport(): void</p> <p>+ saveAndSendReport(report: Report): void</p> <p>+ handleVoiceRecognitionFailure(): void</p>	<p><b>AmbulanceReroutingManager</b></p> <ul style="list-style-type: none"> <li>- instance: AmbulanceReroutingManager</li> <li>- currentRoute: Route</li> <li>- gpsService: GPSService</li> <li>- trafficService: TrafficMonitoringService</li> <li>- hospitalService: HospitalBedAvailabilityService</li> <li>- notificationService: NotificationService</li> </ul> <p>+ getInstance(): AmbulanceReroutingManager</p> <p>+ loginResponder(credentials): bool</p> <p>+ accessReroutingModule(): void</p> <p>+ monitorTraffic(): void</p> <p>+ suggestRoute(): Route</p> <p>+ connectAmbulanceToRouter(): void</p> <p>+ checkMetrics(): void</p> <p>+ logPerformance(record: PerformanceRecord): void</p> <p>+ addAlertListener(listener: AlertListener): void</p> <p>+ notifyAlert(alert: PerformanceAlert): void</p>	<p><b>Emergency Service</b></p>