

Topic: Hospital Management System

Subtopic: Requirements Engineering

TEAM MEMBERS

- **Shpëtim Shabanaj**
- **Artjol Zaimi**
- **Arjan Muka**
- **Eglis Braho**
- **Nikola Rigo**
- **Arlin Bashllari**
- **Marin Tartaraj**

Executive Summary

Project Overview

This project aims to develop a Hospital Management System (HMS) designed to streamline hospital operations and enhance patient care. The intended audience includes hospital administrators, medical staff (doctors, nurses, lab technicians, pharmacists), and supporting personnel (receptionists, housekeeping, ambulance services). The HMS will optimize appointment scheduling, patient record management, bed allocation, billing, pharmacy inventory, and emergency handling. It will also improve communication and resource allocation across hospital departments.

Product/Service Description

The Hospital Management System (HMS) is a comprehensive, integrated software solution designed to optimize hospital operations, improve patient care, and enhance administrative efficiency. It serves as a centralized platform that connects various hospital departments, ensuring seamless coordination between medical staff, administrative personnel, and patients. The system facilitates patient registration, admission, discharge, and transfer, allowing real-time tracking of bed availability and efficient scheduling of medical services. Doctors and nurses can access electronic health records, manage prescriptions, order diagnostic tests, and monitor patient progress, ensuring accurate and timely medical decisions.

The HMS streamlines billing and financial management by automating invoice generation, insurance claim processing, and payment tracking, reducing errors and improving financial transparency. Laboratories and radiology departments benefit from integrated test ordering and result management. Pharmacy and inventory control functions ensure proper medication tracking, stock level monitoring, and compliance with drug regulations, reducing shortages and waste. The system also includes emergency management, coordinating ambulance dispatch, prioritizing critical cases, and enabling rapid triage to provide timely medical attention.

Resource allocation features optimize staff scheduling, track equipment usage, and ensure the availability of necessary medical supplies. Security and compliance mechanisms safeguard patient data through role-based access controls,

audit logs, and encrypted communication channels. Additionally, the system includes a library and literature search module, providing medical staff with access to research papers, case studies, and clinical guidelines to support evidence-based decision-making. By integrating all hospital functions into a single digital ecosystem, the HMS enhances operational efficiency, reduces manual errors, and ensures high-quality patient care while complying with healthcare regulations and standards.

Project Goals & Objectives

The Hospital Management System aims to:

- Optimize hospital workflows by automating patient registration, appointments, and medical record management.
- Enhance patient experience through faster service, reduced waiting times, and online access to records.
- Improve hospital efficiency by reducing manual errors, minimizing paperwork, and streamlining internal processes.
- Enable real-time reporting and analytics to support better decision-making and resource allocation.

Product Context

The HMS will interface with multiple external and internal systems, including:

- Electronic Health Records (EHR) for patient history and medical details.
- Pharmacy Inventory Management for medication tracking.
- Billing & Payment Processing for seamless financial transactions.
- Laboratory Information System for test order and result uploads.
- Radiology and Imaging Systems for managing scans and reports.
- Ambulance & Emergency Response coordination for critical cases.
- Library and Literature Search for medical reference and research.

The system will be web-based with mobile accessibility, ensuring remote access for doctors, nurses, and administrators.

Assumptions

The project assumes:

- All medical staff and administrators have basic IT literacy to operate the system.
- The hospital has the necessary IT infrastructure, including servers, internet access, and devices.
- Government and industry regulations will be followed for data security.
- Medical records will be digitized and available for integration into the system.
- A stable internet connection is available for cloud-based access and real-time updates.

Constraints and Dependencies

- System Security & Access Control:
 - Only authorized personnel will have access to sensitive patient records.
 - Multi-factor authentication may be required for critical modules.
- Integration with Existing Systems:
 - There might be an integration with GPS system and an external library API.
- Parallel Operation with Old Systems:
 - During the initial deployment phase, the HMS may operate alongside existing manual or semi-automated systems.
- Audit & Logging:
 - The system will maintain a comprehensive audit log for tracking changes to patient records, staff access, and medication administration.
- Module Dependencies:
 - The Electronic Health Record module must be functional before enabling appointment scheduling and medical profiling.
 - Resource Allocation and Staff Scheduling modules must be in place before emergency handling features go live.
 - Billing & Payment Processing must integrate with existing financial systems before being fully operational.
- Training & Adoption – Hospital staff must undergo training sessions to familiarize themselves with the system's workflows and functionalities.

Stakeholders

1. Primary Stakeholders (Directly Involved in System Use & Management)

These stakeholders actively interact with the system and rely on its functionalities.

| Stakeholder | Role & Interaction with HMS |
|-------------------------|--|
| Patients | Register, book appointments, view medical records, make payments, receive notifications (via SMS or platform), and consult/contact doctors. |
| Doctors | Manage appointments, access patient records, prescribe medication, and interact with lab & pharmacy modules. |
| Nurses | Monitor patients, administer medications, update records, and track assigned tasks. |
| Receptionists | Register patients, schedule appointments, and manage inquiries/requests. |
| Pharmacists | Manage prescriptions, issue medicines and track stock. |
| Lab Technicians | Process lab tests, update test results, and communicate findings to doctors. |
| Radiologists | Review and upload diagnostic imaging to the system. |
| Hospital Administrators | Manage doctors, departments, finances, patient records, and system security. |
| IT Support Team | Maintain system infrastructure, troubleshoot issues, and ensure system security and backups. |
| Financial Manager | Oversee hospital finances, track expenditures, manage budgets, process payments, ensure financial compliance, and generate financial reports. |
| Inventory Manager | Monitor stock levels, track hospital supplies, manage restocking, prevent shortages, and coordinate with procurement for timely orders. |
| HR Manager | Manage hospital staff records, oversee recruitment, track attendance, handle payroll, ensure compliance with labor laws, and coordinate training programs. |
| Procurement Manager | Manage hospital procurement processes, track supplier contracts, purchase orders, approve and purchase requests from different departments |

2. Secondary Stakeholders (Indirectly Affected by HMS Usage)

These stakeholders benefit from HMS indirectly or use system-generated data for decision-making.

| Stakeholder | Role & Interaction with HMS |
|-----------------------------------|--|
| Hospital Management/Board Members | Oversee system performance, generate reports, and track operational efficiency. |
| Government Health Departments | Ensure compliance with health regulations, collect patient statistics, and audit hospital activities. |
| Insurance Companies | Process claims, verify patient details, and ensure smooth transactions between hospital and citizens. |
| Medical Researchers | Use anonymized patient data for medical studies and trend analysis. |
| Healthcare Regulators | Ensure compliance with safety and medical standards, audit hospital records, and grant certifications. |
| Medical Equipment Suppliers | Manage procurement, track hospital inventory, and provide maintenance updates. |

Users Of System

End users are individuals who access HMS functionalities daily for medical or operational purposes.

| End User | Description |
|---------------|---|
| Patients | Use the HMS to book appointments, check records, view appointments, view scheduled treatment, pay bills, and receive medical care. |
| Doctors | Access patient records, manage schedules, issue prescriptions, request examination from lab or radiology sector, and coordinate treatments. |
| Nurses | Access patient records, view schedule/appointments, update patient medical data (ex: patient was provided a medication at a time, etc.) |
| Pharmacists | Access medical inventory management system, view issued prescriptions by doctors, handle medical request from different hospital sectors. |
| Receptionists | Access appointment schedule, view patient/doctor/nurse information, register patients. |

| | |
|---------------------|---|
| Lab Technicians | Access requests from doctors for analysis of specific patients, post results on patients' online folder. |
| Radiologists | Access requests from doctors for radiology of specific patients, post images on patients online folder. |
| Housekeeping | Access requests from nurses or reception for specific room cleaning. |
| Administrators | Maintain hospital operations, ensure system security, and manage role-based access control(permission granting) and register different end-users to system. |
| Financial Manager | Access hospital finances, track expenditures, manage budgets, process payments, ensure financial compliance, and generate financial reports. |
| Inventory Manager | Monitor stock levels, track hospital supplies, manage restocking, prevent shortages, and coordinate with procurement for timely orders. |
| HR Manager | Access staff records, uploads recruitment, handle payroll, track attendance and coordinate training programs |
| Procurement Manager | Uses HMS to manage every process that has to do with product ordering such as approving and presenting orders to the board . |

User Profile Details

1. Patients

- Type: General public, including students, faculty, staff, and other individuals seeking healthcare services.
- Experience: Varies widely
- Technical Expertise: Basic familiarity with smartphones and online booking; some may need assistance.
- Other Characteristics: Require a user-friendly interface with clear navigation, mobile accessibility, etc.

2. Doctors

- Type: Medical professionals, including specialists and general practitioners.

Experience: High experience in medical field; varying familiarity with hospital IT systems.

Technical Expertise: Moderate to high; comfortable using medical software but require intuitive design for efficiency.

- Other Characteristics: Need fast access to patient records, reliable prescription management, and seamless integration with other hospital modules.

3. Nurses

- Type: Healthcare professionals assisting doctors and monitoring patients.
- Experience: High experience in medical care, varying experience in digital record-keeping.
- Technical Expertise: Moderate; used to electronic health records but need efficient workflow management.
- Other Characteristics: Require a straightforward interface to update records quickly and track medication administration.

4. Pharmacists

- Type: Medical professionals managing prescriptions and drug inventory.
- Experience: High experience in pharmaceuticals, varying experience in HMS.
- Technical Expertise: Moderate to high; familiarity with inventory and prescription management software.
- Other Characteristics: Need reliable stock management, prescription tracking, and integration with doctors and billing modules.

5. Receptionists

- Type: Front-desk staff handling appointments and inquiries.
- Experience: Customer service experience, varying experience with medical software.
- Technical Expertise: Moderate; require training on appointment scheduling and patient registration features.
- Other Characteristics: Need an intuitive, error-proof interface to manage high patient volumes efficiently.

6. Lab Technicians & Radiologists

Type: Medical professionals performing diagnostics.

Experience: Highly skilled in their fields, with some familiarity with digital lab/radiology systems.

- Technical Expertise: Moderate; need seamless integration between doctor requests and test result uploads.
- Other Characteristics: Require precise workflow tracking and easy-to-use result upload functions.

7. Administrators & Managers

- Type: Hospital management and IT personnel.
- Experience: High experience in management, finance, or IT.
- Technical Expertise: Advanced; responsible for data security, role-based access, and system maintenance.
- Other Characteristics: Require robust control over user permissions, analytics tools, and system customization features.

8. Housekeeping

- Type: Cleaning staff
- Experience: -
- Technical Expertise: Basic familiarity with smartphones

- Other Characteristics: Require a user-friendly interface with clear navigation.

9. Inventory Manager

- Type: Hospital staff responsible for managing medical and non-medical inventory.
- Experience: Experienced in supply management and inventory control, with knowledge of hospital logistics.
- Technical Expertise: Moderate to high; familiar with inventory management software and procurement systems.
- Other Characteristics: -

10. Financial Manager

- Type: Hospital staff overseeing financial operations, including billing, budgeting, and revenue management.
- Experience: Extensive experience in financial planning, accounting, and hospital revenue cycle management.
- Technical Expertise: High; proficient in financial software, billing systems, and regulatory compliance tools.

Other Characteristics: Require detailed financial reporting, real-time tracking of hospital expenditures, integration with insurance providers, and compliance with healthcare financial regulations.

11. HR Manager

- **Type:** Hospital staff responsible for recruitment, payroll, employee records, and workforce management.
- **Experience:** Strong background in human resource management, employee relations, and labor laws.
- **Technical Expertise:** Moderate to high; familiar with HR management software, payroll systems, and compliance tracking.
- **Other Characteristics:** Require efficient employee record-keeping.

12. Procurement Manager

- **Type:** Hospital administrative staff responsible for managing procurement and supply chain operations.
- **Experience:** Experienced in procurement, supplier management, and hospital inventory control.
- **Technical Expertise:** Moderate familiarity with procurement software, financial tracking systems, and inventory management tools.
- **Other Characteristics:** Requires an intuitive dashboard for supplier tracking, purchase order approvals, and budget monitoring. Needs automated alerts for low stock, pending approvals, and delivery delays.

Requirements

| ID | Module Name | Applicable End Users | Requirement Definition | Detailed System Functionality & Comments | Priority | Date Reviewed | Member Approved |
|--------|-------------|--|--|---|----------|---------------|-----------------|
| REQ-01 | Login | Patient, Doctor, Admin, Nurse, Receptionist, Pharmacist, Lab Technician, Radiologist | Users shall log in using unique credentials using government issued ID and a password for secure access. | The system verifies credentials entered by user and redirects users to their specific dashboard, where each dashboard is accommodated to user role. User role is automatically determined by the system. | 1 | | |

| | | | | | | | |
|--------|--------------|---------------------|--|--|---|--|--|
| REQ-02 | Registration | Admin, Receptionist | Users of this feature shall be able to register in the system other end-users. | <p>Admin: Will be able to register all kind of users. Receptionist: Will be able to register patients.</p> <p>Users of this features will be required to fill a registration form of user data. They must select the new user role. Note that receptionist shall be able to only add patients to system.</p> <p>The system shall validate correctness and existence of this data before saving data to database.</p> <p>The system validates and checks for duplicate registrations before approval.</p> | 1 | | |
| | | | | <p>If registration is successful a message will be displayed on screen and a randomly generated password for the new user.</p> <p>Else a message will be displayed with errors encountered such as wrong or skipped fields.</p> | | | |

| | | | | | | | |
|--------|------------------------|-----------------------|--|--|---|--|--|
| REQ-03 | Appointment Scheduling | Patient, Receptionist | <p>These users shall be able to book an appointment with a specific doctor via this feature. Patients can select a doctor, date, and time to request an appointment.</p> | <p>Patient: Patients will have the ability to book appointments directly by selecting a specific doctor from a list provided within the system. Once a doctor is chosen, patients can view their available dates and times, select a preferred slot, and confirm the appointment after providing essential details such as name, contact information, and the reason for the visit. In cases where the selected time is unavailable, the system will dynamically check the doctor's schedule and suggest the nearest possible alternatives. Booking confirmations and reminders will then be sent to the patient via email or SMS.</p> <p>Receptionist: Receptionists, on the other hand, will have the capability to create appointments on behalf of patients for requests made in person or over the phone. Using their interface, receptionists can search</p> | 1 | | |
|--------|------------------------|-----------------------|--|--|---|--|--|

for doctors, view their availability, and finalize bookings by entering patient details and confirming the selected slot. Additionally, they will be able to update or cancel appointments as needed.

The system will ensure that doctor's schedules are kept accurate, accounting for holidays, vacations, and existing bookings, thereby avoiding conflicts or double bookings. In cases where a doctor is fully booked, the system will propose alternative doctors with similar specialties or the earliest possible available slots.

| | | | | | | | |
|--------|--------------------------|-----------------|--|--|---|--|--|
| REQ-04 | Appointment Cancellation | Patient, Doctor | Patients can cancel an appointment; doctors can also cancel an appointment | Patients have the option to cancel their appointments directly through the system, while doctors can also make cancellations for their scheduled appointments if necessary. In both cases, the system ensures a prompts user for rescheduling before the cancellation is finalized. This feature allows patients or doctors to select a new date and time that works for them. Once a cancellation or rescheduling is confirmed, the system updates all relevant schedules and sends notifications to the affected parties, ensuring clear communication and a streamlined experience. | 1 | | |
| | | | | | | | |

| | | | | | | | |
|--------|-------------------------------------|-----------------------|--|---|---|--|--|
| REQ-05 | Payment Processing (Billing System) | Reception, Pharmacist | Reception can initiate billing processing for each paid service offered by hospital. | The hospital billing system will allow receptionists to handle transactions efficiently and accurately. When a patient completes an appointment with a doctor or other hospital services, the receptionist can initiate the billing process through this feature. The receptionist will input or select service details which will be listed on screen (such as consultation fees, surgeries, lab tests, radiography, etc.). The system will then generate a detailed bill, itemizing each service and its associated cost. Patients can provide their payment details, whether through card or cash to complete the transaction. Before finalizing the payment, the system will verify its success, ensuring there are no issues. After a successful payment, a receipt will be issued to the patient, and the system will update the financial records automatically. Note that payment have to be full and no partial payments would be accepted. This feature shall be used by reception after the service is offered by hospital to patient. | 2 | | |
|--------|-------------------------------------|-----------------------|--|---|---|--|--|

| | | | | | | | |
|--------|--------------------|-----------|---|---|---|--|--|
| REQ-06 | Profile Management | All users | This feature enables all users to view and update their personal information. | The system provides a feature that allows all users to view and update their personal information with ease. When a user accesses their profile, they can review existing details such as their name, contact information, and other displayed data. If user tends to update his data, the user can make the necessary changes, and the system will validate these modifications to ensure they meet predefined rules. Once the changes pass validation, the system securely updates the information in the database, ensuring data integrity constraints and predefined constraints. | 2 | | |
|--------|--------------------|-----------|---|---|---|--|--|

| | | | | | | | |
|--------|-------------------------|------------------------|---|---|---|--|--|
| REQ-07 | Patient Medical Profile | Doctor, Nurse, Patient | <p>Doctors and Nurses shall be able to search patient names in order to view their medical history and test/radiology results.</p> <p>Patients shall be able to view their medical portfolio.</p> | <p>Doctors and nurses will have the ability to search for patients by name through the system, giving them access to the patient's general information, comprehensive medical history, and results from lab tests or radiology analyses. This feature enables them to efficiently review and interpret critical medical data, ensuring they can deliver precise and effective care.</p> <p>→After users of this feature enter patient name, the patient's portfolio will be displayed. Some data that will be displayed are: personal</p> | 1 | | |
| | | | | <p>information, blood group, previous medical tracks, lab analysis(if any), radiology results(if any).</p> <p>Furthermore, in cases when patients are accommodated in hospital for medical care, data such as reason of hospitalization, date of arrival, medical approach that is being used and some other technical fields.</p> <p>Patients shall also be able to see the allowed fields of their medical portfolio.</p> | | | |

| | | | | | | | |
|--------|-------------------------------------|---------------------------------|-----------------------------------|--|---|--|--|
| REQ-08 | Medication Prescription And Viewing | Doctor Pharmacist Patient | Doctors can prescribe medications | <p>Doctors will have the ability to prescribe medications directly through the system, creating detailed prescriptions for their patients. They will input the patient's ID, full name, and medical condition, then select the prescribed medications. Upon submission, the system will require confirmation before finalizing the prescription. Once issued, the system securely stores the prescription and makes it accessible to both patients and pharmacists.</p> <p>Pharmacists receive real-time notifications for new prescriptions, making them able to review and prepare prescribed medication.</p> <p>Patients can access their</p> | 1 | | |
|--------|-------------------------------------|---------------------------------|-----------------------------------|--|---|--|--|

| | | | | | | |
|-------|----------------------------------|--------|---|---|---|--|
| | | | | <p>prescriptions to stay informed about their treatment plans, while pharmacists can view issued prescriptions to ensure accurate dispensing.</p> <p>The system will display all relevant prescription details in an organized format, offering search and filter options based on patient name, medication type, or issue date. Pharmacists will also be able to check inventory availability for prescribed medications, ensuring efficient fulfillment of prescriptions.</p> | | |
| REQ-9 | Electronic Health Records Update | Doctor | Doctors can update patient medical history by adding new medical conditions or adding new treatment to patient's portfolio. | <p>Doctors will have the ability to update a patient's medical history by adding new medical conditions or recording treatments administered during their care. When a patient enters the hospital for examination, their medical portfolio is made available to the attending doctor, who can review it and document any updates, such as diagnoses, ongoing treatments, or changes in the patient's condition.</p> <p>Doctor will be able to add new updates by filling fields like date,</p> | 1 | |

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | <p>new medical condition, description of situation, necessary treatment, etc.</p> | | | |
|--|--|--|--|--|--|--|--|

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | <p>(Doctor should enter updates for both consultations and hospitalized patients)</p> | | | |
|--|--|--|--|--|--|--|--|

| | | | | | | | |
|--------|--|--|--|---|---|--|--|
| REQ-10 | Lab Test Ordering and Result Upload | Doctor Nurse Lab Technician | Doctors and nurses can order lab tests for a specific patient and should provide test type. Lab staff can upload results for doctor review and patient access | Doctors and nurses can order lab tests for specific patients through the system. They must specify the test type (ex: blood test), patient ID, full name, and desired due time to ensure clarity for the laboratory. Once submitted, the system securely transmits the details to the lab for processing. Lab technicians will have the ability to upload test results directly into the system, making them accessible for review by doctors and, when appropriate, by patients. Lab technicians can search for patient or doctor details and upload test result documents. Once uploaded, the system automatically notifies relevant parties, ensuring doctors are promptly informed and patients can access results if permitted. | 1 | | |
| REQ-11 | Radiology & Imaging Ordering and Result Upload | Doctor, Nurse, Radiology Technician | Doctors and nurses can order radiology service by using this feature. Radiology staff can upload results | Doctors and nurses can order radiology services for specific patients through the system. When placing an order, they must specify the type of service required (ex: arm X-ray), patient ID, full name, and desired due time to ensure clarity for the radiology laboratory. Once | 1 | | |

| | | | | | | |
|--|--|--|-------------------------------|--|--|--|
| | | | for doctor review and access. | submitted, the system securely transmits the details to the lab for processing. Radiology technicians will have the ability to upload radiology results directly into the system, making them accessible for review by doctors and, when appropriate, by patients. Radiology technicians can search for patient or doctor details and upload result documents. Once uploaded, the system automatically notifies relevant parties, ensuring doctors are promptly informed and patients can access results if permitted. | | |
|--|--|--|-------------------------------|--|--|--|

| | | | | | | | |
|--------|----------------------------|-------|---|---|---|--|--|
| REQ-12 | Inpatient & Bed Management | Nurse | Assign, update, and track patient bed availability. | The system will enable nurses to efficiently assign, update, and track patient bed availability in real-time. During patient admission, staff can input patient details such as ID, condition, and assign an appropriate available bed which will be listed on the fields. As a result, the system will update its status to "Occupied." When the patient is released, the bed's status will change to "Needs Cleaning.". Once cleaning is complete, the status updates to "Available." In the case of patient transfers, the current bed will be marked as "Needs Cleaning," while a new bed matching the patient's requirements will be reserved. A real-time dashboard with userfriendly GUI and search filters provides nurses with an organized overview of bed statuses across the facility. This functionality improves workflow and reduces delays. | 1 | | |
|--------|----------------------------|-------|---|---|---|--|--|

| | | | | | | | |
|--------|-----------------------|--------|--|---|---|--|--|
| REQ-13 | Nurse Task Assignment | Doctor | Doctors shall be able to assign tasks to nurses regarding a specific patients treatment. | Doctors shall have the capability to assign tasks to nurses related to a specific patient's treatment through the system. When a doctor creates a task, they can include details such as the patient's information, the nature of the task, date & time, and its urgency. The system processes this information and sends real-time notifications to the assigned nurse, clearly indicating the task's priority level and automatically adding task to the nurse's task schedule. When selecting the nurse for a task at a time, system must notify doctor for nurse availability at that time. Nurses will receive these notifications on their platform. This functionality enhances coordination between doctors and nurses, ensuring timely and efficient patient care. | 2 | | |
|--------|-----------------------|--------|--|---|---|--|--|

| | | | | | | | |
|--------|--|---------------|--|--|---|--|--|
| REQ-14 | Medication Administration for patients | Nurse, Doctor | When nurse completes a task assigned by the doctor for hospitalized or non-hospitalized patient, she shall be able to add this update on patient's portfolio | When a nurse completes a task assigned by a doctor for either a hospitalized or non-hospitalized patient, she will have the ability to document this update directly in the patient's portfolio. The nurse shall be able to fill fields such as date & time of update, type of medication, optional observations or comments, inflammations encountered (if any), etc. These updates will be added to patients medical portfolio. | 1 | | |
|--------|--|---------------|--|--|---|--|--|

| | | | | | | | |
|--------|-----------------------------|------------|--|--|---|--|--|
| REQ-15 | Pharmacy & Stock Management | Pharmacist | Track medicine stock, update inventory, and manage restocking. | The system shall enable Pharmacists option for management of medicine stock by tracking inventory levels, updating records, and facilitating restocking processes. It continuously monitors the stock of medicines across various storage units and updates inventory in real time as medications are issued or replenished. When stock levels for any medicine fall below a predefined number (like 10 units), the system automatically generates low-stock alerts, notifying the pharmacist to initiate the restocking process. Additionally, the system keeps track of medicine expiry dates and sends timely alerts when medications are nearing | 1 | | |
|--------|-----------------------------|------------|--|--|---|--|--|

expiration. This ensures that expired medicines are promptly removed from stock and replaced, maintaining the safety and reliability of the hospital's inventory. This feature must provide the user with functionality of adding new medicines that are not registered before in system, by entering data such as barcode, name, expiration date, type of medicament, purpose, quantity, price, restock date, etc. When a unit of medication is sold, stock will automatically be decremented. In case of restock pharmacist shall be able to update fields such as quantity, restock date and new expiration date.

| | | | | | | | |
|--------|-------------------------|--------------|---|---|---|--|--|
| REQ-16 | Medical Staff Timetable | Doctor Nurse | Doctors and Nurses shall be able to view their work timetable and tasks assigned. | The system will provide doctors and nurses with access to their work timetables, ensuring that despite the differences in their roles and responsibilities, the timetable's representation will follow a consistent format. Doctor's timetable shall display appointments booked by patients and schedule for meeting with hospitalized patients. Nurse's timetable shall display tasks assigned by doctors at defined timeslots. This feature is straight forward and does not require user input. | 2 | | |
| | | | | | | | |

| | | | | | | | |
|--------|------------------|--------|--|--|---|--|--|
| REQ-17 | Surgery Planning | Doctor | Doctor shall be able to place surgeries for a patient if necessary | The system will enable doctors to plan and schedule surgeries for patients whenever necessary. Through the surgery planning feature, doctors can select a patient and input details such as the type of surgery, the proposed date and time, the operating room, and any required surgical team members or equipment. The system will verify the availability of necessary resources, including operating rooms and staff, before finalizing the scheduling. Once the surgery is scheduled, notifications will be sent to relevant departments, such as the surgical team, anesthesiologists, and nursing staff, ensuring proper coordination and preparation. The system will also update the patient's portfolio with the planned surgery details, allowing tracking and monitoring of their treatment plan. This feature ensures efficient and accurate | 1 | | |
|--------|------------------|--------|--|--|---|--|--|

| | | | | | | | |
|--------|-----------------------------|---------------|---|---|---|--|--|
| | | | | surgery planning while supporting clear communication across all involved teams. | | | |
| REQ-18 | Emergency Handling & Alerts | Doctor, Nurse | Handle emergency cases with priority assignment and notification. | The system will provide a dedicated emergency handling mechanism to efficiently manage urgent medical cases. When an emergency arises, doctors, nurses, administrators, or members of the emergency response team can initiate the process by flagging the case as high priority. The system will immediately override standard scheduling protocols, reallocating resources such as staff, rooms, and equipment to address the emergency. Notifications will be sent in real time to all relevant personnel, including the emergency response team, ensuring they are promptly informed and prepared to act. The system will prioritize the case within the workflow, dynamically adjusting schedules and reallocating | 1 | | |

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | <p>resources as needed, without disrupting critical ongoing activities.</p> | | | |
|--|--|--|--|--|--|--|--|

| | | | | | | | |
|--|--|--|--|---|--|--|--|
| | | | | <p>If needed, patients meetings with doctors can be reallocated on another time.</p> | | | |
|--|--|--|--|---|--|--|--|

| | | | | | | | |
|--------|--------------------------|--|--|---|---|--|--|
| REQ-19 | Room Cleaning Management | Patient Receptionist Nurse Housekeeping | <p>Users of this feature shall be able to make requests for room cleaning. Their request should be pushed into the stack of actions.</p> <p>Housekeeping shall access the stack of room cleaning calls and shall be able to update room cleaning status.</p> | <p>The system will provide a feature allowing patients, receptionists, and nurses to request room cleaning efficiently. Users can submit a cleaning request directly through the system by selecting the room and specifying any additional details if needed, such as urgency or specific cleaning requirements. Once a request is made, it will be added to a prioritized stack of cleaning actions. The system will maintain this action stack dynamically, ensuring that requests are processed in the order of submission unless marked as high priority, which will move them to the top of the stack.</p> <p>Notifications will be sent to housekeeping staff, enabling them to view and address the requests promptly.</p> <p>Housekeeping staff can view these requests and select one to mark as "In Progress" when they begin cleaning a room. Once the cleaning is completed, they will update the request status to "Completed,"</p> | 2 | | |
|--------|--------------------------|--|--|---|---|--|--|

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | ensuring accurate tracking and transparency. The system will automatically update the stack to reflect the current status of each request, ensuring that pending tasks remain visible and prioritized appropriately. | | | |
|--|--|--|--|--|--|--|--|

| | | | | | | | |
|--------|------------------|---------|---|---|---|--|--|
| REQ-20 | Staff scheduling | Manager | Create and display work schedules for doctors, nurses, and other staff. | The system will allow managers to create and display work schedules for doctors, nurses, and other staff members. Managers can input schedule details such as staff names, roles, assigned shifts, oncall hours and days off. The system will provide a user-friendly interface that ensures schedules are clear and easy to manage. It will also include features like conflict detection to prevent overlapping shifts or scheduling errors. Once schedules are finalized, they will be made accessible to staff members in a standardized format, ensuring consistency and clarity. Changes on doctors shift directly affects their availability on appointment scheduling feature done by patients, so if the manager wills to change doctors shift at some days, the system shall trigger an alert. This functionality ensures efficient staff management and helps maintain smooth hospital operations. | 2 | | |
| | | | | | | | |

| | | | | | | | |
|--------|-------------------|---------|--|--|---|--|--|
| | | | | | | | |
| REQ-21 | Report Generation | Manager | Manager shall require a report generation based on different filter. | The system will provide managers with the ability to generate reports based on various filters to meet their specific requirements. Managers will be able to select from predefined filters, such as staff performance, patient admission trends, bed occupancy rates, emergency cases arrived in hospital, inventory levels on Pharmacy and financial data. Additionally, they will have the option to apply custom filters, like date ranges, departments, or specific criteria. Once the desired filters are applied, the system will compile and generate detailed, accurate reports in a user-friendly format, such as tables, charts, or graphs. This functionality ensures that managers can easily access actionable insights and make informed decisions based on the data. | 2 | | |

| | | | | | | | |
|--------|-------------------------------|---------------|---|---|---|--|--|
| REQ-22 | Library and Literature search | Doctor Nurses | The system should provide a digital library and literature search feature for doctors and | The system will include a robust digital library and literature search feature specifically designed for doctors and nurses. This feature will allow healthcare professionals to access a wide range of medical | 3 | | |
| | | | nurses, to access medical research, journals, books and clinical guidelines. | resources, including research papers, journals, eBooks, clinical guidelines, and other relevant publications. Users will be able to perform searches using keywords, topics, authors, or publication dates to locate specific materials efficiently. The system will support features like bookmarking and downloading required literature. This feature aims to enhance knowledgesharing and support evidence-based medical practices for improved patient care. | | | |

| | | | | | | | |
|--------|-----------------------------------|-----------------|---|---|--|--|--|
| REQ-23 | Library and Literature Management | Library Manager | The system will include a system which lets the library manager to add, update and delete literature from the system. | <p>The system shall provide a comprehensive Library and Literature Management feature that enables the Library Manager to efficiently manage medical literature, journals, research papers, and reference materials.</p> <p>The Library Manager will have the ability to add new literature by entering details such as title, author, publication date, category, keywords, and a brief description. The system shall support file uploads, allowing PDF versions, research documents, or links to online resources to be attached for easy access.</p> <p>Additionally, the Library Manager shall have the ability to update</p> | | | |
| | | | | existing literature, replacing outdated files, or appending supplementary information to keep the database current. If a piece of literature is no longer relevant or has been substituted by newer research, the Library Manager shall be able to delete it, ensuring the library remains accurate and up to date. | | | |

| | | | | | | | |
|--------|------------------------|-------------------|---|---|---|--|--|
| REQ-24 | Vaccination Management | Patient | The system will include a vaccination management feature to help patients effectively manage their vaccination records. | The system will include a vaccination management feature to help patients effectively manage their vaccination records. This feature will allow patients to view their vaccination history, including details such as the type of vaccine, date of administration, and the age that it should be taken. The system will also enable scheduling upcoming vaccinations based on the recommended timelines, helping patients stay on track with their immunization plans. Automated reminders via SMS will be sent to patients as their vaccination dates approach, ensuring they are wellinformed and prepared. | 1 | | |
| REQ-25 | Resource Allocation | Inventory Manager | The system should enable | The system will enable managers to efficiently allocate and track hospital | 2 | | |

| | | | | | | | |
|--|--|--|---|---|--|--|--|
| | | | efficient allocation and tracking of hospital resources, including medical staff and equipment. | physical resources and equipment. It will provide real-time visibility into the availability, location, and status of critical assets, such as medical devices, hospital beds, wheelchairs, and diagnostic machines. Managers will be able to assign these resources to specific departments or cases based on current requirements and priorities. It will also track inventory levels for essential supplies, ensuring timely restocking and minimizing disruptions | | | |
|--|--|--|---|---|--|--|--|

| | | | | | | | |
|--------|---|---|--|---|---|--|--|
| REQ-26 | Inventory Item & Procurement Management | Inventory Manager, Procurement Officer, Board (admins & managers) | The system manages inventory, procurement, and supplier interactions, ensuring accurate tracking, approvals, and order processing. | The system provides inventory management for authorized users to add, edit, update, and delete items with validated details like name, category, quantity, cost, and storage location. Edits and deletions require confirmation, and all changes are logged for accountability. Procurement officers can generate and approve purchase orders, track deliveries, verify invoices, and notify staff of delays. They can access supplier details for procurement. The inventory manager submits purchase requests for low-stock items, which the procurement officer can approve, modify, or reject with a reason. The board makes the final decision via voting about the best choice. Approved requests generate purchase orders sent to suppliers and the financial manager. | 2 | | |
| | | | | | | | |

| | | | | | | | |
|--------|----------------------|--|--|---|---|--|--|
| REQ-27 | Supplier Management | Inventory Manager, Procurement Officer | The system shall maintain a list of suppliers and enable basic CRUD to that list. | The system ensures seamless supplier management by allowing users to add, update, or remove supplier details, such as name, contact information, and company details, while maintaining accurate records. Historical data for each inventory item, including past transactions and suppliers, is preserved for transparency and informed decision-making. Changes to supplier details do not affect historical records, and all modifications are logged with user and timestamp information for accountability. This feature streamlines inventory operations and supports reliable supplier tracking. | 2 | | |
| REQ-28 | Ambulance Management | Reception | Reception shall be able to track availability of ambulances and location status of occupied ambulances | The system will include an ambulance management feature that allows receptionists to efficiently track the availability and location of ambulances. Receptionists will have access to real-time information showing whether ambulances are available, on route, or occupied with patients. For occupied ambulances, | 1 | | |

| | | | | | | |
|--------|---------------------|-------|---|--|---|--|
| | | | | the system will display their current location by usage of GPS. This feature will assist in coordinating ambulance services effectively, ensuring prompt response times during emergencies and minimizing delays in patient transport. The system will also update the status of ambulances dynamically when ambulances are detected in hospital position. | | |
| REQ-29 | Permission Granting | Admin | Assign different access levels based on user roles. | The system enforces permissions to prevent unauthorized access. | 2 | |

| | | | | | | | |
|--------|--------------------|-----------|--|--|---|--|--|
| REQ-30 | Visitor Management | Reception | Receptionists shall be able to register visitor for all hospitalized patients and record them at patients portfolio. | The system will include a visitor management feature that allows receptionists to register visitors for all hospitalized patients. When a visitor arrives, the receptionist can log their details, such as name, contact information, relationship to the patient, and visit timings. This information will be securely stored and automatically linked to the corresponding patient's portfolio, providing a detailed record of all visitors. This functionality ensures accurate tracking, supports hospital security protocols, and allows for quick reference if needed. | 3 | | |
|--------|--------------------|-----------|--|--|---|--|--|

| | | | | | | | |
|--------|----------------------------|------------|--|--|---|--|--|
| REQ-31 | Employee Management System | HR Manager | HR Manager shall be able to manage staff and all their records including personal data, compensation and skills inventory. | The system shall provide a comprehensive Employee Management feature to enhance HR processes. It will enable the HR team to manage employee records, including personal details, job roles, performance evaluations, compensations, and attendance. The system will support features such as adding new employees, updating existing records, and archiving former employees. It will include a search functionality to locate employee records quickly using filters like name, department, or role. This feature shall provide manager the ability to manage skills and qualifications of all employees. | 1 | | |
|--------|----------------------------|------------|--|--|---|--|--|

| | | | | | | | |
|--------|----------------------|-------------------|--|--|---|--|--|
| REQ-33 | Financial Management | Financial Manager | The Finance Manager shall set and monitor department budgets while ensuring proper financial recordkeeping for auditing purposes | <p>The system shall enable the Finance Manager to define budgets annually or by department for effective financial planning. Users can input budget details such as total amounts, expense categories, and department names. The system will track expenses in real-time and calculate all hospital expenditures, including HR expenses and inventory costs. Additionally, the system shall log all financial transactions, including purchases, payments, and budget updates. Users input</p> <p>transaction details such as type, amount, date, and relevant identifiers, while the system validates and stores the data for proper financial tracking and audits. The system shall also define an implied surplus or shortage of funds and maintain accurate financial logs to keep track of all hospital expenditures.</p> | 1 | | |
|--------|----------------------|-------------------|--|--|---|--|--|

Non-Functional Requirements

| ID | Type | Requirement |
|-----------|-------------------------|---|
| NF-REQ-01 | Usability Requirement | The system shall provide an intuitive, user-friendly interface that allows medical professionals to efficiently access patient records, schedule appointments, and perform administrative tasks without extensive training. |
| NF-REQ-02 | Usability Requirement | The system shall include comprehensive user documentation, tutorials, and an interactive help feature to assist staff in learning the system quickly. |
| NF-REQ-03 | Usability Requirement | The system shall offer a role-based dashboard with customized layouts for doctors, nurses, receptionists, and administrative staff to improve efficiency. |
| NFREQ-04 | Usability Requirement | The system shall be accessible to people with disabilities, ensuring keyboard navigation, screen reader compatibility, and color contrast options. |
| NFREQ-05 | Usability Requirement | The system shall include a notification center that displays alerts for appointment reminders, pending lab results, system downtime notifications, and urgent medical updates. |
| NF-REQ-06 | Performance Requirement | The system shall support a minimum of 1000 concurrent users without experiencing performance degradation. |
| NF-REQ-07 | Performance Requirement | The system shall process at least 500 transactions per second under normal hospital workload. |

| | | |
|----------|-------------------------|--|
| NFREQ-08 | Performance Requirement | The system shall retrieve patient records in less than 2 seconds, ensuring quick access to medical data. |
|----------|-------------------------|--|

| | | |
|-----------|---------------------------|--|
| NF-REQ-09 | Performance Requirement | The system shall maintain an average response time of 200 milliseconds for database queries. |
| NF-REQ-10 | Performance Requirement | The system shall ensure that 95% of all user interactions complete within 1 second, including appointment scheduling and billing transactions. |
| NF-REQ-11 | Performance Requirement | The system shall be designed to handle peak traffic periods such as mass emergency admissions without performance degradation. |
| NF-REQ-12 | Availability Requirement | The system shall have an uptime of 99.9%, ensuring uninterrupted operation of critical hospital services. |
| NF-REQ-13 | Availability Requirement | The system shall allow hospitals to schedule maintenance downtime with prior notifications to all users to minimize disruptions. |
| NF-REQ-14 | Availability Requirement | The system shall ensure that no more than 1 hour of unscheduled downtime occurs per year, meeting high-reliability standards. |
| NF-REQ-15 | Data Backup Requirement | The system shall automatically back up patient records and hospital data every 15 minutes to prevent data loss. |
| NF-REQ-16 | Data Backup Requirement | The system shall store daily, weekly, and monthly backups in multiple geographically distributed locations for disaster recovery. |
| NF-REQ-17 | Data Recovery Requirement | The system shall ensure that in the event of data corruption or loss, full data recovery can be performed within a maximum of 30 minutes. |

| | | |
|-----------|-----------------------------------|--|
| NF-REQ-18 | System Error Handling Requirement | The system shall log all errors, including hardware failures and software crashes, and notify system administrators immediately. |
| NF-REQ-19 | Security Requirement | The system shall require multi-factor authentication for hospital staff accessing sensitive information. |
| NF-REQ-20 | Security Requirement | The system shall implement role-based access control to limit access to patient data based on user roles. |

| | | |
|-----------|--|--|
| NF-REQ-21 | Security Requirement | The system shall log all failed login attempts and notify administrators if unauthorized access is suspected. |
| NF-REQ-22 | Security Requirement | The system shall implement biometric authentication (fingerprint or facial recognition) for high-security areas like surgery places. |
| NF-REQ-23 | Security Requirement | The system shall include intrusion detection mechanisms, ensuring that any unauthorized access attempts trigger alerts for security personnel. |
| NF-REQ-24 | Security Requirement | The system shall be regularly tested for vulnerabilities, including penetration testing and security audits. |
| NF-REQ-25 | Organizational Requirement | The system shall comply with hospital IT policies, ensuring alignment with existing medical record-keeping practices. |
| NF-REQ-26 | Organizational Requirement | The system shall allow customizable workflows that can be adapted based on different hospital policies and procedures. |
| NF-REQ-27 | System Downtime Management Requirement | The system shall display real-time notifications to users if system downtime is scheduled, with at least 24-hour prior notice. |

| | | |
|-----------|--|--|
| NF-REQ-28 | System Downtime Management Requirement | The system shall provide offline capabilities for critical functionalities, allowing emergency data entry and retrieval during network failures. |
| NF-REQ-29 | Organizational Requirement | The system shall allow hospital administrators to define operational rules and permissions, ensuring compliance with internal governance policies. |
| NF-REQ-30 | Organizational Requirement | The system shall maintain historical patient records for a minimum of 10 years, in compliance with medical record retention laws. |
| NF-REQ-31 | External Requirement | The system shall comply with local healthcare regulations, ensuring legal compliance for patient data protection. |
| NF-REQ-32 | External Requirement | The system shall integrate with government health databases, allowing hospitals to report public health data and access national patient records. |
| NF-REQ-33 | System Alert Requirement | The system shall send automated alerts via email and SMS to hospital IT personnel in case of critical system failures. |
| NFREQ-34 | External Requirement | The system shall include multi-language support, ensuring usability for diverse staff and patients in international hospital networks. |
| NF-REQ-35 | Security Requirement | The system must ensure secure transaction processing and encrypt sensitive data. |
| NF-REQ-36 | Data backup Requirement | Changes to a patient's history should be logged for audit purposes. |

Subtopic: Use Case Tables & Diagram

Author: Artjol Zaimi

| UC Name | UC-01 User Login |
|-------------------------------------|--|
| Summary | This use case describes how a user logs into the Hospital Management System using a government-issued ID and a password. After successful authentication, the system redirects users to their respective dashboards based on their assigned roles (like doctor, nurse, receptionist). |
| Dependency | - |
| Actors | <ul style="list-style-type: none"> ▪ Primary Actor: Patient, Doctor, Admin, Nurse, Receptionist, Pharmacist, Lab Technician, Radiologist, Financial Manager, Procurement Manager, Housekeepers, HR Manager |
| Preconditions | <ul style="list-style-type: none"> ▪ The user must have a valid account in the system. ▪ The system must be connected to the user database to verify credentials. ▪ The user must have a government-issued ID and a password registered in the system. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The user navigates to the Login Page. The system prompts the user to enter their government-issued ID and password. 2. The user enters their credentials and clicks the Login button. 3. The system validates the credentials by checking the database records. 4. If credentials are valid: <ul style="list-style-type: none"> -The system verifies the user role (like doctor, nurse, patient). -The system redirects the user to their rolespecific dashboard (like a doctor sees patient charts, a receptionist sees appointment schedules). -The login session is established and logged audit purposes. 5. The user is now successfully logged into the system and can access relevant functionalities. |
| Description of Alternative Sequence | <ol style="list-style-type: none"> 1. Step 4: If the user enters an incorrect ID or password, the system displays an error message: "Invalid credentials. Please try again." |

| | |
|-----------------------------|--|
| | <ol style="list-style-type: none"> 2. Step 4: If the user attempts more than 5 failed logins, the system temporarily locks the account for 15 minutes and notifies the user via email/SMS. 3. Step 4: If the database is temporarily unavailable, the system displays a message: "Login service is currently down. Please try again later." 4. Step 5: If the user is logging in from an unrecognized device, the system triggers multifactor authentication |
| Non-Functional Requirements | <ul style="list-style-type: none"> • Security: NF-REQ-21: The system shall log all failed login attempts and notify administrators if unauthorized access is suspected. • Security: NF-REQ-20: The system shall implement role-based access control to limit access to patient data based on user roles. • Security: NF-REQ-19: The system shall require multi-factor authentication for hospital staff accessing sensitive information. • Performance: NF-REQ-08: The system shall retrieve patient records in less than 2 seconds, ensuring quick access to medical data. • Usability: NF-REQ-01: The system shall provide an intuitive, user-friendly interface that allows medical professionals to efficiently access core system functions. |
| Postconditions | <ul style="list-style-type: none"> • The user is authenticated successfully. • The system identifies the user's role and grants access to the appropriate dashboard. • The login event is recorded in the system logs with a timestamp and user ID. |

Author: Marin Tartaraj

| UC Name | UC-02 Registration |
|-------------------------------------|--|
| Summary | This use case describes how an Admin or Receptionist registers new users in the system. The Admin can register any type of user, while the Receptionist can only register patients. The system validates user input, prevents duplicate entries, and generates a random password upon successful registration. |
| Dependency | - |
| Actors | <ul style="list-style-type: none"> • Primary Actors: Admin, Receptionist |
| Preconditions | <ul style="list-style-type: none"> • The actor must have the necessary permissions to perform registration. • The system must be online and accessible. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. Admin or Receptionist selects the "Register New User" option. 2. The system displays the registration form. 3. The actor fills in the user details and selects the user role. 4. The actor submits the registration form. 5. The system validates the entered data. 6. The system checks for duplicate users in the database. 7. If no duplicates are found and data is valid, the system registers the user. 8. The system generates a random password for the new user. 9. The system displays a success message along with the generated password. |
| Description of Alternative Sequence | Step 5: If mandatory fields are missing, the system displays an error message and prompts the user to complete all fields. |

| | |
|-----------------------------|---|
| | <p>Step 6: If the user already exists, the system prevents duplicate registration and displays an error message.</p> <p>Step 7: If any other validation errors occur (like invalid email format), the system provides an error message explaining the issue.</p> <p>Step 8: The actor corrects the errors and resubmits the form.</p> |
| Non-Functional Requirements | <ul style="list-style-type: none"> • NF-REQ-01 (Usability Requirement): The registration process should have an intuitive, user-friendly interface for Admins and Receptionists. • NF-REQ-02 (Usability Requirement): Documentation and tutorials should assist Admins and Receptionists in understanding the registration process. • NF-REQ-03 (Usability Requirement): A role-based dashboard ensures Admins and Receptionists have access to the registration feature. • NF-REQ-09 (Performance Requirement): Quick response time (200ms for database queries) ensures efficient validation and registration. • NF-REQ-10 (Performance Requirement): 95% of user interactions should complete within 1 second, ensuring fast registration. • NF-REQ-19 (Security Requirement): Multi-factor authentication ensures that only authorized staff can register users. • NF-REQ-20 (Security Requirement): Role-based access ensures only Admins can register all users, while Receptionists can only register patients. |
| Postconditions | <ul style="list-style-type: none"> • If successful, the new user is registered and receives a randomly generated password. • If unsuccessful, the system displays an error message, and the user is not registered. |

Author: Artjol Zaimi

| UC Name | UC-03 Appointment Scheduling |
|------------------------------|--|
| Summary | This use case enables patients and receptionists to schedule appointments with doctors through the Hospital Management System. Patients can choose doctors and available time slots, while receptionists can book on behalf of patients during visits or calls. The system checks real-time doctor schedules to avoid conflicts and automatically sends booking confirmations and reminders via SMS or email. |
| Dependency | - |
| Actors | Primary Actors: Patient, Receptionist |
| Preconditions | <ul style="list-style-type: none"> • The user must be logged into the system with a valid patient or receptionist account. • The user must have access rights to use the appointment scheduling feature. • Doctors must be registered and have active schedules available in the system. • The system must be connected to the hospital database for retrieving availability and saving appointments. • Patient contact details must be available for sending notifications. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. User (Patient or Receptionist) selects the "Appointment Scheduling" option from their dashboard. 2. The system determines the user's role and: -If Receptionist: Displays a patient search field. -If Patient: Auto-loads their own profile in the background. 3. The user selects or confirms the patient for the appointment: <ul style="list-style-type: none"> -Receptionist enters/searches the patient ID or name. -Patient skips this step as it's auto-filled. 4. The system loads the patient's basic info and shows the appointment booking form. 5. The user selects: <ul style="list-style-type: none"> -Department/Specialty -Preferred Doctor (filtered by selected department) 6. The system shows real-time available time slots for the selected doctor (from UC16 and UC20). 7. The user selects a time slot and fills in the purpose of visit. |

| | |
|-------------------------------------|--|
| | <p>8. The system:</p> <ul style="list-style-type: none"> -Validates input -Locks the selected slot -Stores appointment in the database <p>9. The system sends a confirmation message to the patient via SMS/email.</p> <p>10. The appointment appears in the patient's upcoming appointment list.</p> |
| Description of Alternative Sequence | <ul style="list-style-type: none"> • Step 3- Patient Not Found: System shows an error if the patient ID or name is incorrect. Suggests retry or correct input. • Step 6 – No Time Slots Available: If the doctor has no free slots, the system shows the next available time or similar doctors. • Step 7 – Missing Info: If required fields are empty or wrong, the system highlights the issues and blocks submission. • Step 9 – Notification Fails: If confirmation can't be sent, the appointment is still saved. System shows a warning and retries later. |
| Non-Functional Requirements | <ul style="list-style-type: none"> • Performance: NF-REQ-10: The system shall have an uptime of 99.9%, ensuring uninterrupted access to the appointment scheduling feature. • Availability: NF-REQ-12: Access to patient records must be secured through role-based authentication. • Performance: NF-REQ-8: The system shall retrieve patient records in less than 2 seconds, ensuring quick access to doctor availability and scheduling data. |
| Postconditions | <ul style="list-style-type: none"> • The appointment is saved in the system with doctor, patient, date, time, and reason. • The doctor's availability is updated to prevent doublebooking. • A confirmation is shown to the user and sent by SMS/email. • The appointment can be viewed, edited, or canceled later (if allowed). • An audit log is recorded with user info and timestamp. |

Author: Arlin Bashllari

| UC Name | UC-04 Appointment Cancellation |
|-------------------------------------|---|
| Summary | This use case allows patients and doctors to cancel appointments. The system ensures that users are prompted for rescheduling before finalizing cancellations. It updates schedules and sends notifications to affected users to maintain clear communication. |
| Dependency | - |
| Actors | Primary Actors: Patient, Doctor |
| Preconditions | -The patient or doctor must be logged into the system. - There must be an existing appointment to cancel or reschedule. |
| Description of Main Sequence | 1. The patient or doctor initiates the cancellation process. 2. The system prompts the user to reschedule instead of canceling. 3. The user either selects a new date/time or confirms cancellation. 4. The system updates the schedules accordingly. 5. The system sends notifications to affected users. 6. The process is completed successfully. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> □ If the user does not wish to reschedule, the system proceeds with cancellation and updates records. □ If the system encounters an error, it notifies the user and logs the issue for administrators. □ If the user attempts to cancel a past appointment, the system denies the request. |
| Non Functional Requirements | <ul style="list-style-type: none"> - Performance : NF REQ-08 : The system must process cancellations and rescheduling within 5 seconds. - Notifications should be sent instantly upon confirmation. - Usability: NF REQ-01 : The interface should be userfriendly and intuitive. - Availability: NF REQ-12 : The system should maintain high availability (99.9% uptime). |
| Postconditions | <ul style="list-style-type: none"> □ The appointment is either canceled or rescheduled. □ The schedule is updated accordingly. □ Affected users receive notifications. □ The system is ready for further appointment modifications. |

Author: Shpetim Shabanaj

| UC Name | UC-05 Billing Component & Payment Processing |
|-------------------------------------|--|
| Summary | The hospital billing system enables receptionists to process payments for services offered by the hospital. The system ensures efficient and accurate billing, verifies transactions, and updates financial records. |
| Dependency | - |
| Actors | <ul style="list-style-type: none"> • Primary Actor: Receptionist, Pharmacist • Indirect Actor: Patient |
| Preconditions | <ul style="list-style-type: none"> • The patient has existing records in the system about the services that hospital has provided to him, or a list of medical prescription issued by the doctor on patient's name(or medical portfolio) • The primary actors shall have permission to access this feature. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The receptionist logs into the Billing Component. 2. The receptionist selects the patient and retrieves the list of services provided from his portfolio saved in database. 3. The system displays each service costs and total cost (service cost * quantity). 4. The receptionist confirms the charges with the patient. 5. The patient provides payment details (cash or card). In case of debit card payment, system checks the patient's debit card for the purchase amount and, if approved, creates a debit card purchase authorization number. 6. The system processes the payment and verifies transaction success. 7. Upon successful payment, the system generates a receipt. 8. The system updates financial records automatically, including adding the amount of money received to the total cash in. 9. The receptionist provides the receipt to the patient. <p>NOTE: Pharmacist performs the same steps as receptionist.</p> |
| Description of Alternative Sequence | Step 5: If the payment fails, the system notifies the receptionist and the receptionist asks the patient to retry payment or use a different method. |

| | |
|-----------------------------|--|
| | <p>... If the system experiences an error, an IT administrator is alerted.</p> <p>Step 5: If the patient refuses to pay, hospital policy for unpaid services is followed.</p> <p>Step 2: If the system cannot retrieve the patient data, the receptionist shall contact IT administrator.</p> <p>Step 7: If the system does not generate the receipt, IT administrator should be notified.</p> <p>Step 4: If the patient reports not valid service list on his receipt, proper validation shall undergo.</p> |
| Non-Functional Requirements | <ul style="list-style-type: none"> • Performance: NF-REQ-08: Payment processing should be completed within 2 seconds. • Security: NF-REQ-35: The system must ensure secure transaction processing and encrypt sensitive data. • Availability: NF-REQ-12: The billing system must be available 99.9% of the time. • Usability: NF-REQ-01: The system should provide a user-friendly interface for receptionists to reduce billing errors. |
| Postconditions | <ul style="list-style-type: none"> • The payment is successfully recorded. • The financial records(cash flows) are updated. • The patient receives a receipt. • The billing system ensures no partial payments are accepted. • The system shall be ready for next possible transaction. |

Author: Marin Tartaraj

| UC Name | UC-06 Profile Management |
|-------------------------------------|--|
| Summary | This use case allows users to view and update their personal information. The system validates changes before updating the database to ensure data integrity. |
| Dependency | - |
| Actors | Primary Actors: All Users |
| Preconditions | <ol style="list-style-type: none">1. The user must be logged into the system.2. The system must be online and accessible. |
| Description of Main Sequence | <ol style="list-style-type: none">1. The user selects "Profile Management" from the menu.2. The system displays the user's profile details.3. The user reviews their personal information.4. If the user wants to make changes, they edit the required fields.5. The user submits the updated information.6. The system validates the modifications against predefined rules.7. If validation is successful, the system updates the database with the new information.8. The system displays a confirmation message indicating a successful profile update. |
| Description of Alternative Sequence | <p>Step 4: If the user enters missing or incorrect data (like invalid phone number format), the system displays an error message and prompts for corrections.</p> <p>Step 6: If validation fails due to business rules violations (like email already exists, invalid characters), the system displays an error message.</p> <p>Step 7: If the database update fails due to system issues, the system notifies the user and suggests retrying later.</p> |

| | |
|-----------------------------|---|
| Non-Functional Requirements | <ul style="list-style-type: none"> • NF-REQ-01 (Usability Requirement): Profile management should be user-friendly for all users. • NF-REQ-02 (Usability Requirement): Interactive help should assist users in updating their profiles correctly. • NF-REQ-03 (Usability Requirement): A role-based dashboard ensures personalized access to profile management. • NF-REQ-04 (Usability Requirement): Accessibility support (keyboard navigation, screen readers) helps users with disabilities. • NF-REQ-09 (Performance Requirement): Profile updates should be processed quickly with a 200ms response time for queries. • NF-REQ-10 (Performance Requirement): 95% of updates should be completed within 1 second. • NF-REQ-19 (Security Requirement): Multi-factor authentication protects sensitive user data from unauthorized access. • NF-REQ-20 (Security Requirement): Role-based access ensures users can only update their own profiles. |
| Postconditions | <ol style="list-style-type: none"> 1. If successful, the user's updated information is stored securely in the database. 2. If unsuccessful, the user is notified of errors and prompted to make corrections. |

Author: Shpetim Shabanaj

| UC Name | UC-07 Patient Medical Profile |
|-------------------------------------|--|
| Summary | This use case allows doctors and nurses to search for a patient and access their medical history, lab tests, and radiology results. Patients can directly view specific fields of their own medical portfolio. |
| Dependency | - |
| Actors | Primary Actors: Doctor, Nurse, Patient |
| Preconditions | <ul style="list-style-type: none"> The patient must have a registered medical profile in the system. The user (doctor/nurse/patient) must be logged into the system with the appropriate access level, because this use case differs by the level and type of user. The system must be functional and connected to the hospital database. |
| Description of Main Sequence | <ol style="list-style-type: none"> The system determines the role of the user after his request for accessing this feature. <p>DOCTOR and NURSE</p> <ol style="list-style-type: none"> The doctor/nurse enters the patient's name or unique ID in the search bar. The system retrieves and displays the patient's profile, including personal details, medical history, diagnoses, prescribed treatments, and test results. If applicable, the system also displays hospitalization details such as admission date, reason for hospitalization, and ongoing medical approach. The doctor/nurse reviews the data and logs out when necessary. <p>PATIENT</p> <ol style="list-style-type: none"> The system automatically retrieves and displays the patient's personal medical data, including past diagnoses, lab/radiology results, and treatment history (only the allowed fields). The patient reviews their information and logs out when done. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> Step 3: If the patient does not exist in the system, an error message is displayed. |

| | |
|-----------------------------|---|
| | <ul style="list-style-type: none"> ▪ If the user does not have the required permissions, access is denied. ▪ If some medical records are missing, the system notifies the user and suggests updating or verifying the data. |
| Non-Functional Requirements | <ul style="list-style-type: none"> ▪ NF-REQ-08: The system must retrieve patient records within 2 seconds. ▪ NF-REQ-20: Access to patient records must be secured through role-based authentication. |
| Postconditions | <ul style="list-style-type: none"> ▪ The doctor/nurse/patient must have the required information displayed. |

Author: Nikola Rigo

| UC Name | UC-08 Medication Prescription & Viewing |
|------------------------------|--|
| Summary | Allows doctors to prescribe medications through the system. Once a prescription is confirmed, it is stored securely and made available to both patients and pharmacists. Pharmacists receive real-time notifications, review prescriptions, check inventory, and prepare the medication, while patients can view their treatment details. |
| Dependency | - |
| Actors | Primary Actor: Doctor Secondary Actor: Pharmacist, Patient |
| Preconditions | <ul style="list-style-type: none">- The doctor must be authenticated and logged into the system.- The patient's record must exist and be accessible in the system.- The system should have updated inventory information for medications. |
| Description of Main Sequence | <ol style="list-style-type: none">1. Login & Access: The doctor logs into the system and navigates to the prescription module.2. Input Patient Details: The doctor enters the patient's ID, full name, and medical condition.3. Select Medications: The doctor selects the appropriate medications from the system and the quantity.4. Review & Confirm: The doctor reviews the prescription details and confirms the prescription.5. Store & Notify: The system securely stores the prescription and sends real-time notifications to pharmacists.6. Pharmacist Review: Pharmacists review the prescription, check inventory availability, and prepare the medication.7. Patient Access: Patients can view their prescription details through the system via their medical profile. |

| | |
|-------------------------------------|--|
| Description of Alternative Sequence | <ul style="list-style-type: none"> - Prescription Cancellation: If the doctor cancels the prescription before confirmation, no data is stored and the process terminates. - Inventory Issues: If a prescribed medication is not available, the pharmacist receives an alert or an alternative workflow may be initiated. - System Errors: In case of system or network errors during confirmation, the system prompts the doctor to retry or contact support. |
| Non-Functional Requirements | <ul style="list-style-type: none"> - NF REQ-01: The system provides a user-friendly interface for doctors, pharmacists, and patients to efficiently access prescriptions. - NF REQ-02: The system includes tutorials and documentation for training medical staff. - NF REQ-03: Role-based dashboards ensure that doctors, pharmacists, and patients have customized views. - NF REQ-06: Notification system alerts pharmacists when a prescription is issued. - NF REQ-08: Prescription retrieval should take no more than 2 seconds. - NF REQ-12: Ensures 99.9% uptime so prescriptions are always accessible. - NF REQ-19: Multi-factor authentication required for accessing patient prescriptions. - NF REQ-20: Role-based access control restricts prescription visibility based on roles. |
| Postconditions | <ul style="list-style-type: none"> - The prescription is permanently stored and is accessible by both patients and pharmacists. - Pharmacists are alerted to review and process the prescription. - An audit trail is maintained for accountability. - The inventory is updated based on the medications dispensed. |

Author: Shpetim Shabanaj

| UC Name | UC-09 Electronic Health Records Update |
|------------------------------|--|
| Summary | This use case allows doctors to update a patient's medical history by adding new diagnoses, treatments, or conditions. The update applies to both consultations/appointments and hospitalized patients. |
| Dependency | - |
| Actors | Primary Actor : Doctor |
| Preconditions | <ul style="list-style-type: none"> • The patient must have a registered medical profile in the system. • The doctor must be logged into the system with appropriate permissions because this use-case is only for doctors. • The system must be operational and connected to the hospital database. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The doctor navigates to "Patient Records" feature. 2. The doctor searches for the patient by name or unique ID. 3. The system retrieves and displays the patient's medical profile. 4. The doctor selects between "Add Medical Update" and "Update Medical Condition for Hospitalized Patient" 5. In case of "Add Medical Update"→ The doctor enters details such as: <ul style="list-style-type: none"> - Date of consultation - New medical condition (if applicable) - Description of symptoms/diagnosis - Prescribed treatment (if any) - Additional notes In case of "Update Medical Condition for Hospitalized Patient"→ The doctor enters details such as: <ul style="list-style-type: none"> - Date and Time of update - Changes in condition - Adjustments to treatment/medication - Any new medical procedures performed 6. The doctor submits the update and then is asked to confirm update. 7. System validates the update or data entered. 8. System saves the change in database. |

| | |
|-------------------------------------|--|
| Description of Alternative Sequence | <ul style="list-style-type: none"> Step 2: If the patient record does not exist, an error message is displayed. Step 6: If doctor does not confirm changes he will be redirected to the form page. Step 7: If data entered is not valid, the doctor shall be notified and prompted for other data. |
| Non Functional Requirements | <ul style="list-style-type: none"> NF-REQ-15: The system shall automatically back up patient records and hospital data every 15 minutes to prevent data loss. NF-REQ-20: The system must ensure role-based access control to prevent unauthorized edits. NF-REQ-36: Changes to a patient's history should be logged for audit purposes. |
| Postconditions | <ul style="list-style-type: none"> The patient's medical record is successfully updated. The system logs the update with timestamps and doctor's identification. Nurses and other authorized users can view the updated records. |

Author: Arjan Muka

| | |
|------------|--|
| UC Name | UC-10 Lab Test Ordering and Result Upload |
| Summary | <p>The hospital system enables medical staff (nurses and doctors) to order lab tests for patients and allows lab technicians to upload the results. The system ensures that lab test orders are accurately recorded, and results are updated in the patient's medical profile for review by medical staff.</p> |
| Dependency | - |

| | |
|------------------------------|---|
| Actors | <ul style="list-style-type: none"> ● Primary Actors: Medical Staff (Nurse, Doctor), Lab Technician ● Secondary Actor: Patient |
| Preconditions | <ul style="list-style-type: none"> ● The patient must have an existing medical profile in the system with relevant medical history. ● The medical staff (nurse or doctor) must have permission to order lab tests. ● The lab technician must have permission to access the system and upload test results. ● The system must be operational and accessible to both medical staff and lab technicians. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The medical staff (nurse or doctor) logs into the hospital system. 2. The medical staff selects the patient's profile from the system database. 3. The medical staff orders the required lab tests for the patient, specifying the type of tests needed. 4. The system records the lab test order and notifies the lab technician of the pending tests. 5. The lab technician logs into the system and retrieves the list of ordered tests for the patient. 6. The lab technician performs the tests and uploads the results into the system. |
| | <ol style="list-style-type: none"> 7. The system updates the patient's medical profile with the new test results. 8. The system notifies the medical staff that the test results are available for review. 9. The medical staff reviews the test results and takes appropriate action (like updating the patient's treatment plan). |

| | |
|-------------------------------------|---|
| Description of Alternative Sequence | <p>Step 2: If the system fails to retrieve the patient's profile, an error is displayed, and the medical staff is prompted to contact the IT administrator.</p> <p>Step 6: If the lab technician cannot upload results due to a system error, an error is logged, and the IT administrator is notified.</p> <p>Step 8: If the system fails to notify the medical staff of uploaded results, an error is logged, and the IT administrator is notified.</p> |
| Non Functional Requirements | <ul style="list-style-type: none"> ● NF-REQ-08: Patient records must be retrieved in under 2 seconds for quick access during lab test ordering. ● NF-REQ-20: Secure role-based access must be enforced to ensure only authorized medical staff can order tests and lab technicians can upload results. ● NF-REQ-18: The system must log all errors (like failed uploads) and notify administrators for immediate resolution. ● NF-REQ-12: The system must maintain 99.9% uptime to ensure uninterrupted lab test ordering and result uploads. |
| Postconditions | <ul style="list-style-type: none"> ● The lab test order is successfully recorded in the system. ● The lab test results are uploaded and updated in the patient's medical profile. ● The medical staff is notified of the available test results. |
| | <ul style="list-style-type: none"> ● The system ensures that all actions (ordering, uploading, and reviewing) are logged for audit purposes. ● The system is ready for the next lab test order or result upload. |

Author: Nikola Rigo

| UC Name | UC-11 Radiology & Imaging Ordering and Result Upload |
|------------------------------|---|
| Summary | Enables doctors and nurses to order radiology services for patients and allows radiology technicians to upload imaging results. Once an order is submitted, the system securely transmits the details to the lab, and after processing, results are uploaded and made available for doctor review (and patient access when permitted). |
| Dependency | - |
| Actors | <p>Primary Actors: Medical Staff (Nurse, Doctor), Radiology Technician</p> <p>Secondary Actor: Patient</p> |
| Preconditions | <ul style="list-style-type: none"> - The doctor or nurse must be authenticated and logged into the system. - The patient's record exists in the system. - The radiology lab system is accessible for order processing. - Radiology technicians must have valid credentials to upload imaging results. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. Order Placement: <ol style="list-style-type: none"> a. Doctors or nurses log into the system and navigate to the radiology ordering module. b. They select the radiology service type (like arm X-ray) and input the patient ID, full name, and desired due time. 2. Order Submission: <ol style="list-style-type: none"> a. The system securely transmits the order details to the radiology lab for processing. 3. Result Upload: <ol style="list-style-type: none"> a. Radiology technicians access the order, search for the patient or doctor details, and upload the imaging result documents into the system. 4. Notification & Access: <ol style="list-style-type: none"> a. Once uploaded, the system automatically notifies doctors and (if permitted) patients that the results are available for review. |
| | |

| | |
|-------------------------------------|---|
| Description of Alternative Sequence | <p>Order Modification/Cancellation: If a doctor or nurse needs to modify or cancel an order before transmission, they can do so, and the system updates or cancels the order accordingly without processing.</p> <ul style="list-style-type: none"> - Upload Errors: If a radiology technician encounters an error during the result upload, the system prompts a retry or notifies support for assistance. - Permission Issues: If a patient is not permitted to view the results, the system restricts access accordingly and only notifies the doctor. |
| Non Functional Requirements | <ul style="list-style-type: none"> - NF REQ-01: Intuitive interface for doctors, nurses, and radiology technicians to place orders and upload results. - NF REQ-02: User documentation and tutorials for learning the system quickly. - NF REQ-03: Role-based dashboards provide customized views for each actor. - NF REQ-06: Notification alerts for new orders and uploaded results. - NF REQ-08: Radiology result retrieval should take less than 2 seconds. - NF REQ-12: System ensures 99.9% uptime to avoid disruptions. - NF REQ-19: Multi-factor authentication required for accessing imaging data. - NF REQ-20: Role-based access ensures only authorized personnel can view sensitive medical images. |
| Postconditions | <ul style="list-style-type: none"> - The radiology order is stored and transmitted to the lab for processing. - Imaging results are securely uploaded and associated with the patient's record. - Relevant parties (doctors and, if permitted, patients) are notified of the available results. - An audit trail is maintained for all orders and uploads. |

Author: Arjan Muka

| UC Name | UC-12 Inpatient & Bed Management |
|---------------|---|
| Summary | <p>The hospital system enables nurses to manage inpatient care by registering patients, assigning beds, transferring patients between beds, releasing patients, and tracking bed availability. The system ensures efficient bed management and updates bed status in real time to support hospital operations.</p> |
| Dependency | |
| Actors | Primary Actor: Nurse |
| Preconditions | <ul style="list-style-type: none">• The patient must have an existing medical profile in the system.• The nurse must have permission to access the bed management system and perform actions like registering patients, assigning beds, and releasing patients.• The system must have an up-to-date record of bed availability.• The system must be operational and accessible to the nurse. |

| | |
|-------------------------------------|---|
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The nurse logs into the hospital system. 2. The nurse selects the "Inpatient & Bed Management" module. 3. The nurse registers a new patient into the system (if not already registered) by entering the patient's details. 4. The system checks bed availability and displays a list of available beds. 5. The nurse assigns a bed to the patient, and the system updates the bed status to "occupied." 6. If the patient needs to be transferred to another bed (like due to medical requirements or ward changes), the nurse initiates a patient transfer, selects a new bed, and the system updates the bed statuses accordingly (previous bed becomes "available," new bed becomes "occupied"). 7. When the patient is ready to be discharged, the nurse releases the patient from the system, and the system updates the bed status to "available." 8. Throughout the process, the nurse can track bed availability to ensure efficient use of hospital resources. 9. The system logs all actions (registration, bed assignment, transfer, and release) for audit purposes. |
| Description of Alternative Sequence | <p>Step 3: If the system fails to retrieve the patient's profile, an error is displayed, and the nurse is prompted to contact the IT administrator.</p> <p>Step 5: If the system fails to update the bed status after assignment, an error is logged, and the IT administrator is notified.</p> <p>Step 6: If the system fails to update bed statuses during a patient transfer, an error is logged, and the IT administrator is notified.</p> |
| | |

| | |
|-----------------------------|---|
| Non-Functional Requirements | <ul style="list-style-type: none"> ● NF-REQ-08: Patient records must be retrieved in under 2 seconds for efficient bed assignment. ● NF-REQ-20: Secure role-based access must be enforced to ensure only nurses can manage bed assignments and transfers. ● NF-REQ-12: The system must maintain 99.9% uptime to ensure uninterrupted bed management operations. ● NF-REQ-28: Offline capabilities must be provided to allow emergency bed assignments during network failures. |
| Postconditions | <ul style="list-style-type: none"> ● The patient is successfully registered in the system (if not already registered). |
| | <ul style="list-style-type: none"> ● A bed is assigned to the patient, and the bed status is updated to "occupied." ● If a patient transfer occurs, the bed statuses are updated accordingly. ● Upon patient release, the bed status is updated to "available." ● The system ensures that bed availability is accurately tracked and updated in real time. ● All actions (registration, bed assignment, transfer, and release) are logged for audit purposes. ● The system is ready for the next bed management task. |

Author: Shpetim Shabanaj

| UC Name | UC-13 | Nurse Task Assignment |
|-------------------------------------|--|-----------------------|
| Summary | This use case allows doctors to assign treatment-related tasks to nurses, ensuring efficient coordination for patient care. | |
| Dependency | - | |
| Actors | Primary actor: Doctor Secondary actor: Nurse | |
| Preconditions | <ul style="list-style-type: none"> The patient must have an active medical profile and an active treatment plan. System must be connected to database for timetable and patient info retrieval. | |
| Description of Main Sequence | <ol style="list-style-type: none"> The doctor navigates to "Nurse Task assignment". The doctor searches for the patient for whom the task is being assigned. The system displays the patient's treatment details. The doctor clicks the button for adding new nurse Task Assignment The doctor fills in task details, including: <ul style="list-style-type: none"> Task type (like administering medication, monitoring, wound care) Date & time of task execution Priority level (urgent, standard, low) Additional instructions (if needed) The system retrieves a list of available nurses for the selected time slot. The doctor selects a nurse, and the system confirms the nurse's availability. If available, the system assigns the task and updates the nurse's schedule. The system sends a real-time notification to the assigned nurse with task details. Doctor shall track if the task is completed or not. | |
| Description of Alternative Sequence | <ul style="list-style-type: none"> Step 2: If the patient profile does not exist, an error message shall be displayed. Step 4: If there are missing fields or problems with validations of fields, the doctor shall be notified and redirected to form page. Step 5: If no nurses are available at the selected time, the system notifies the doctor and suggests alternative time slots or available nurses. | |
| Non Functional Requirements | <ul style="list-style-type: none"> The system should provide real-time updates on nurse availability. | |

| | |
|----------------|--|
| | <ul style="list-style-type: none"> - NF-REQ-20: Secure role-based access should be enforced to ensure only authorized users can assign or view tasks. - NF-REQ-08: The task assignment process must be completed in under 2 seconds for efficiency. - NF-REQ-36: The system should maintain a log of all assigned and completed tasks for audit and report. |
| Postconditions | <ul style="list-style-type: none"> - The task is successfully assigned to the nurse. - The nurse receives the task. - The task is added to the nurse's schedule and patient treatment records. |

Author: Nikola Rigo

| UC Name | UC-14 Medication Processing for patients |
|------------------------------|---|
| Summary | Enables nurses to document the completion of a medication administration task as assigned by a doctor. The nurse enters details into the patient's portfolio, including date & time, medication type, observations, and any adverse reactions or inflammations encountered. |
| Dependency | - |
| Actors | Nurse |
| Preconditions | <ul style="list-style-type: none"> - The nurse must be authenticated and logged into the system. - A valid task assigned by a doctor exists for either a hospitalized or non-hospitalized patient. - The patient's portfolio is available for updates. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. Task Assignment: <ul style="list-style-type: none"> a. After a doctor assigns a medication administration task to a nurse for a specific patient and the nurse completes the task as per the doctor's instructions. 2. Update Entry: <ul style="list-style-type: none"> a. The nurse accesses the patient's portfolio. b. The nurse fills in the update fields such as date & time, type of medication, optional observations/comments, and any inflammations encountered. 3. Save Update: <ul style="list-style-type: none"> a. The update is securely added to the patient's medical portfolio. |

| | |
|-------------------------------------|---|
| Description of Alternative Sequence | <ul style="list-style-type: none"> - Incomplete Task: If the nurse is unable to complete the task, she can mark it as pending with a note for follow-up, which is then visible to the doctor. - Data Entry Error: If there is an error during update entry (like missing fields), the system prompts the nurse to correct the input before submission. - System Failure: In case of system downtime or errors, the nurse is prompted to retry or log the update later. |
| Non Functional Requirements | <ul style="list-style-type: none"> - NF REQ-01: User-friendly interface for nurses to log medication administration. - NF REQ-02: Tutorials and interactive help features to guide new users. - NF REQ-03: Role-based dashboards for nurses and doctors. - NF REQ-06: System notifications alert doctors when nurses update patient records. - NF REQ-08: Medication record retrieval should take less than 2 seconds. - NF REQ-12: 99.9% uptime ensures real-time access to patient medication logs. - NF REQ-19: Multi-factor authentication required for nurses accessing medication records. - NF REQ-20: Role-based access prevents unauthorized modifications to medical records. |
| Postconditions | <ul style="list-style-type: none"> - The completed medication administration update is saved in the patient's portfolio. - An audit trail of the update is maintained. - The update is immediately available for review by the assigned doctor and relevant healthcare team members. - Notifications or alerts may be triggered for any critical observations noted during the update. |

Author: Eglis Braho & Artjol Zaimi

| UC Name | UC-15 Pharmacy & Stock Management |
|------------------------------|---|
| Summary | This use case allows pharmacists to manage medicine inventory. They can add new medications, update stock levels, handle restocking, and receive alerts for low stock or expiring items. The system ensures accurate tracking and real-time updates to keep the pharmacy well stocked and compliant. |
| Dependency | - |
| Actors | Primary Actor: Pharmacist |
| Preconditions | <ul style="list-style-type: none"> • The pharmacist must be logged into the system with valid credentials and pharmacist role access. • The system must display access to the Pharmacy & Stock Management interface. • The hospital inventory database must be connected and accessible for retrieving, updating, and saving medication records. • The pharmacist must have permission to add, update, or restock inventory items. • Real-time stock tracking and alert features must be active to ensure updates are reflected immediately. • Barcode scanning (if used) and notification services must be operational for efficient data entry and restock alerts. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The pharmacist selects the "Pharmacy & Stock Management" option from their dashboard. 2. The system displays the current list of medications with details like name, quantity, expiration date, and status. 3. The pharmacist chooses to either: <ul style="list-style-type: none"> -Add new medicine by entering barcode, name, type, expiration date, quantity, price, and restock info. -Update existing medicine details (like quantity, expiration date, price). 4. The system validates all input and saves changes to the inventory. 5. As stock is dispensed (like prescriptions issued), the system automatically reduces inventory quantities. 6. When stock falls below a defined threshold, the system sends low-stock alerts to the pharmacist. |

| | |
|-------------------------------------|--|
| | <p>7. The pharmacist processes restocking by updating relevant fields (quantity, restock date, etc.).</p> <p>8. The system logs all updates and changes in the inventory tracking log for auditing purposes.</p> <p>9. The system also sends alerts for upcoming or expired medications, prompting removal or replacement.</p> |
| Description of Alternative Sequence | <p>Step 4: If the medicine already exists in the system, it suggests updating the existing record instead of adding a new one.</p> <p>Step 5: If stock isn't reduced after a prescription, the system logs the issue and alerts the pharmacist.</p> <p>Step 6: If low-stock or expiry alerts fail, the system saves the changes but warns the user that alerts were not sent.</p> <p>Step 7: If another pharmacist is updating the same item, the system prevents changes and asks to try again shortly.</p> |
| Non Functional Requirements | <ul style="list-style-type: none"> • Usability: NF-REQ-01: The system must have a userfriendly interface for pharmacists to manage stock easily. • Availability: NF-REQ-012: The inventory system must be available 99.9% of the time. • System Error Handling: NF-REQ-18: Errors in inventory updates must be logged and reported immediately. • Notification: NF-REQ-05: The system must notify pharmacists when stock levels fall below the threshold or items are near expiration. |
| Postconditions | <ul style="list-style-type: none"> • Medicine stock is added, updated, or restocked successfully in the system. • Low-stock or expiry alerts are generated if conditions are met. • All changes are saved in the hospital inventory database. • The inventory log is updated with timestamp and user info for tracking. • Pharmacists can see the updated inventory in real time. |

Author: Arlin Bashllari

| UC Name | UC-16 Medical Staff Timetable |
|-------------------------------------|---|
| Summary | The system provides doctors and nurses with access to their work timetables in a consistent format. Doctors' timetables display patient appointments and schedules for hospitalized patients, while nurses' timetables display tasks assigned by doctors at defined time slots. This feature is automated and does not require user input. |
| Dependency | - |
| Actors | Primary Actor: Doctor, Nurse |
| Preconditions | <ul style="list-style-type: none"> - The system must have predefined schedules for doctors and nurses. - Doctors must have appointments and patient schedules stored in the system. - Nurses must have assigned tasks registered in the system. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The doctor or nurse accesses the system. 2. The system retrieves the relevant timetable data for the specified user. 3. The system displays the work schedule in a consistent format. 4. The user reviews their schedule. 5. The process is completed successfully. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> □ If no schedule is available, the system notifies the user that no tasks or appointments are currently assigned. □ If the system encounters an error in retrieving data, it logs the issue and alerts the administrator. |
| Non Functional Requirements | <ul style="list-style-type: none"> - Performance : NF REQ-08 : The system must retrieve and display timetables within 3 seconds. - Availability: NF REQ-12 : The system should ensure high availability (99.9% uptime). - Usability: NF REQ-01 ,NF REQ-03 : The interface should be user-friendly and easily navigable. - The timetable format should be uniform for both doctors and nurses. |
| Postconditions | <ul style="list-style-type: none"> □ The medical staff has accessed their up-to-date timetable. □ The system remains ready for further timetable queries. |

Author: Shpetim Shabanaj

| UC Name | UC-17 Surgery Planning |
|-------------------------------------|--|
| Summary | This use case allows doctors to schedule surgeries for patients by selecting a patient, entering surgery details, and ensuring necessary resources are available. |
| Dependency | - |
| Actors | Primary Actor: Doctor Indirect Actors: Nurses, Other team members, Patients |
| Preconditions | <ul style="list-style-type: none"> • The patient must have an existing medical profile |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The doctor navigates to "Surgery Planning" feature. 2. The doctor searches for and selects the patient requiring surgery. 3. The system displays the patient's medical history and relevant details via UC-09 and allows adding surgery details. 4. The doctor enters surgery details, including: <ul style="list-style-type: none"> - Type of surgery - Preferred date and time - Assigned operating room (if applicable) - Required surgical team (like lead surgeon, anesthesiologist, assisting nurses) - Necessary equipment and pre-surgical preparations. 5. The doctor submits the form and then is asked to confirm the request. 6. The system verifies the availability of the selected operating room, surgical team, and required equipment via resource allocation. 7. If resources are available, the system finalizes the surgery schedule and updates the patient's medical portfolio. 8. The system sends notifications to assigned staff and marks necessary items as reserved. |
| Description of Alternative Sequence | <p>Step 2: If the patient record is not found, an error message is displayed.</p> <p>Step 5: If the doctor does not confirm the prompt, he will be redirected to the form's page.</p> <p>Step 6: If the required staff and items are not available, the doctor will be notified and redirected to form's page to make changes.</p> |
| Non Functional Requirements | <ul style="list-style-type: none"> • NF-REQ-19: The system shall require multi-factor authentication for hospital staff accessing sensitive information. • NF-REQ-01: The interface should be user-friendly for efficient scheduling. |

| | |
|----------------|--|
| | <ul style="list-style-type: none"> • NF-REQ-36: The system must log all scheduled surgeries for auditing purposes. |
| Postconditions | <ul style="list-style-type: none"> • The surgery is successfully scheduled and recorded in the patient's profile. • All relevant staff members are notified. • The hospital's scheduling system is updated to reflect the planned surgery. • All necessary items are marked as reserved. |

Author: Eglis Braho

| UC Name | UC-18 Emergency Handling and Alerts |
|---------------|---|
| Summary | The system efficiently manages urgent medical cases by allowing doctors, nurses, administrators, or emergency response teams to flag emergencies as high priority. It overrides standard scheduling, reallocates resources, and sends real-time notifications to ensure a swift and coordinated response while minimizing disruptions to ongoing critical activities. |
| Dependency | None |
| Actors | <ul style="list-style-type: none"> • Primary Actor: Doctor (or Nurse) • Secondary Actors: Administrator, Emergency Response Team |
| Preconditions | <ul style="list-style-type: none"> • The system must have up-to-date resource availability information (rooms, staff, equipment). • The user (doctor, nurse, administrator, or emergency response team member) must be logged into the system with appropriate access rights. • Emergency cases can be flagged by authorized users as high priority. |

| | |
|---|---|
| Description of the Main Sequence | <ol style="list-style-type: none"> 1. A doctor, nurse, administrator, or emergency response team member identifies an emergency medical situation. 2. The nurse selects "Mark as Emergency" 3. The system overrides normal scheduling rules, placing this case at the top of the treatment queue. 4. The system checks for available operating rooms, assigns the closest specialist 5. A push alert is sent to the doctor, on-call nurse team. 6. An elective surgery is delayed and patients are notified automatically 7. If necessary, doctor-patient meetings are rescheduled to make room for the emergency. 8. The emergency handling mechanism remains active until the case is resolved and normal operations are restored. |
| Description of the Alternative Sequence | <p>Step 2-If the system is down or encounters a failure, the actor manually flags the emergency and notifies the response team via alternate communication methods (like phone, pager). Step 4-If resources (like rooms, staff) are unavailable, the system alerts the actor, who can then make manual adjustments or request additional resources from other</p> |
| | departments. |
| Non functional requirements | <ul style="list-style-type: none"> - Performance NF-REQ-11: The system shall handle peak traffic, ensuring resources are reallocated without performance degradation during emergencies. - Security - NF-REQ-20: The system shall implement role-based access control (RBAC) to manage access to emergency handling features based on user roles. - Availability - NF-REQ-12: The system shall have an uptime of 99.9% to ensure availability during critical emergencies. - NF-REQ-27: The system shall provide offline capabilities for critical functionalities, ensuring emergency operations can continue even during system downtime. - Usability - NF-REQ-06: The system shall include a notification center for critical system alerts, including emergency alerts. |

| | |
|----------------|--|
| Postconditions | <ul style="list-style-type: none"> ▪ The emergency case is flagged as high priority, and resources (staff, rooms, equipment) are reallocated to the emergency task. ▪ Relevant personnel receive real-time notifications and act according to the emergency protocol. ▪ The schedule is dynamically adjusted, with non-critical meetings rescheduled as needed to accommodate the emergency. ▪ The system logs all actions and adjustments made during the emergency handling process for future review. |
|----------------|--|

Author: Artjol Zaimi

| | | |
|---------|-------|--------------------------|
| UC Name | UC-19 | Room Cleaning Management |
|---------|-------|--------------------------|

| | |
|---------------|---|
| Summary | <p>This use case allows patients, receptionists, and nurses to request room cleaning. Requests are prioritized by urgency and sent to housekeeping staff, who update the status as tasks are completed.</p> |
| Dependency | |
| Actors | <p>Primary Actor: Patient, Nurse, Housekeeping Staff Secondary Actors: Receptionist</p> |
| Preconditions | <ul style="list-style-type: none"> • The user must be logged in with the correct role (patient, receptionist, nurse, or housekeeping staff). • The system must have access to valid room data and allow submitting or managing cleaning requests. • The hospital database and notification service must be active. • Housekeeping staff must be able to access the cleaning request list. |

| | |
|-------------------------------------|--|
| Description of Main Sequence | <ol style="list-style-type: none"> 1. User (Patient, Receptionist, or Nurse) selects the "Room Cleaning Request" option from their dashboard. 2. They select the room and optionally add cleaning type, urgency, and notes. 3. The request is submitted and added to the action stack. 4. If marked high priority, it goes to the top of the list. 5. Housekeeping receives a notification and views the request. 6. They mark it as "In Progress" when cleaning starts and "Completed" when done. 7. The system updates the request status and logs the action. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> • Step 2 – Missing or Invalid Input: If the user doesn't fill in the required fields, the system shows an error and blocks submission. • Step 3 – Duplicate Request: If the same room already has a pending request, the system notifies the user and prevents duplicate entries. • Step 5 – Notification Failure: If the housekeeping staff don't receive the alert, the request still appears in their dashboard list. • Step 6 – Request Not Started: If the request is not updated to "In Progress" after a set time, the system flags it as delayed. |
| Non Functional Requirements | <ul style="list-style-type: none"> • Usability: NF-REQ-01: The system shall provide an intuitive and easy-to-use interface for submitting and managing room cleaning requests. |
| | <ul style="list-style-type: none"> • Performance: NF-REQ-08: The system shall retrieve and update the cleaning request stack within 2 seconds to ensure responsive user interaction. • Availability: NF-REQ-12: The room cleaning request feature shall be available 24/7 with a minimum of 99.9% uptime. • Notification: NF-REQ-12: The system shall notify housekeeping staff of new high-priority cleaning tasks in real time via dashboard alerts. |
| Postconditions | <ul style="list-style-type: none"> • The cleaning request is successfully saved in the system, tagged with all relevant metadata (ID, timestamp, requester, priority). • The request appears in the dynamic action stack, visible to all housekeeping staff. • High-priority requests are positioned at the top of the stack. • Housekeeping staff can see and update the request status in real time. |

- Once marked "Completed," the request is removed from the active queue and archived.
- All actions (submission, progress, completion) are logged for traceability.

Author: Marin Tartaraj

| UC Name | UC-20 Staff Scheduling |
|---------|------------------------|
|---------|------------------------|

| | |
|------------------------------|--|
| Summary | This use case allows managers to create and display work schedules for doctors, nurses, and other staff. The system detects scheduling conflicts and notifies managers about availability changes affecting appointment scheduling. |
| Dependency | |
| Actors | Primary Actor: Manager |
| Preconditions | <ol style="list-style-type: none"> The manager must be logged into the system. The system must have access to staff data (names, roles, etc.). The system must be online and accessible. |
| Description of Main Sequence | <ol style="list-style-type: none"> The manager selects "Staff Scheduling" from the menu. The system displays the staff scheduling interface. The manager inputs schedule details for staff, including names, roles, shifts, on-call hours, and days off. The system validates the schedule for conflicts (like overlapping shifts). If there are no conflicts, the system saves the schedule. Once the schedule is finalized, the system displays the schedule to the staff in a standardized format. If a change in a doctor's shift affects appointment scheduling, the system triggers an alert to notify the manager of the potential conflict. |

| | |
|---|--|
| <p>Description of Alternative Sequence</p> | <p>Step 4: If the manager schedules a shift that overlaps with an existing shift, the system detects the conflict and displays an error message.</p> <p>Step 5: If the schedule change affects a doctor's availability for appointments, the system triggers an alert and prompts the manager to confirm or modify the change.</p> <p>Step 6: If required information (like staff role, shift time) is missing, the system displays an error message and requests completion.</p> |
| <p>Non-Functional Requirements</p> | <ul style="list-style-type: none"> • NF-REQ-01 (Usability Requirement): A user-friendly scheduling interface is required for Managers. • NF-REQ-03 (Usability Requirement): A role-based dashboard provides different scheduling views for staff members. • NF-REQ-05 (Usability Requirement): Notifications should alert staff about schedule changes and conflicts. • NF-REQ-06 (Performance Requirement): The system must support 1000+ concurrent users to handle scheduling efficiently. • NF-REQ-09 (Performance Requirement): Scheduling queries should have a quick response time (200ms). • NF-REQ-10 (Performance Requirement): 95% of interactions, including schedule changes, should complete within 1 second. • NF-REQ-11 (Performance Requirement): The system must handle peak traffic during shift changes without performance issues. • NF-REQ-27 (System Downtime Management Requirement): If downtime is scheduled, real-time notifications should be displayed to managers. |

| | |
|-----------------------|---|
| Postconditions | <ol style="list-style-type: none"> 1. If successful, the schedule is stored and displayed to staff members. 2. If unsuccessful (like due to conflicts), the system displays error messages and prompts the manager to make corrections. |
|-----------------------|---|

Author: Arlin Bashllari

| UC Name | UC-21 Report Generation |
|------------------------------|--|
| Summary | The system provides managers with the ability to generate reports based on various filters to meet their specific requirements. Reports can include predefined filters such as staff performance, patient admission trends, bed occupancy rates, emergency cases, pharmacy inventory levels, and financial data. Managers can also apply custom filters like date ranges, departments, or specific criteria. The system generates reports in user-friendly formats, including tables, charts, or graphs, ensuring that managers can make informed decisions based on the data. |
| Dependency | - |
| Actors | Primary Actor: All Managers |
| Preconditions | <ul style="list-style-type: none"> - The manager must have the necessary permissions to access reports. - The system must have stored data relevant to the selected filters. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The manager logs into the system. 2. The manager selects predefined filters or applies custom filters. 3. The system retrieves relevant data based on the selected criteria. 4. The system compiles and processes the data. 5. The system generates a report in the chosen format (table, chart, or graph). 6. The manager reviews and analyzes the report. 7. The process is completed successfully. |

| | |
|-------------------------------------|--|
| Description of Alternative Sequence | <ul style="list-style-type: none"> □ If no data is found for the selected filters, the system notifies the manager and suggests adjusting the criteria. □ If an error occurs during data retrieval, the system logs the issue and alerts the administrator. □ If the manager does not have the necessary permissions, the system denies access and provides an error message. |
| Non Functional Requirements | <ul style="list-style-type: none"> - Performance : NF REQ-08 : The system must generate reports within 5 seconds. - Usability: NF REQ-01 : Reports should be presented in an easy-to-read format (tables, charts, graphs). |
| | <ul style="list-style-type: none"> - Reliability: The system must ensure data accuracy and consistency. - Flexibility: The reporting module should support a high level of customization. - Availability: NF REQ-12 : The system must be available 99.9% of the time to ensure uninterrupted report generation. |
| Postconditions | <ul style="list-style-type: none"> □ The manager successfully generates and accesses the report. □ The system remains ready for further report requests. □ The generated report is available for download or further analysis. |

Author: Arlin Bashllari

| UC Name | UC-22 Library and Literature Search |
|-------------------------------------|---|
| Summary | The system provides a robust digital library and literature search feature for doctors and nurses. This feature allows healthcare professionals to access various medical resources, including research papers, journals, eBooks, clinical guidelines, and other relevant publications. Users can search using keywords, topics, authors, or publication dates to efficiently locate materials. Additional features include bookmarking and downloading literature, promoting knowledge-sharing and evidence-based medical practices. |
| Dependency | - |
| Actors | 1. Doctor 2. Nurse |
| Preconditions | - Users must have access permissions to the digital library. - The system must contain indexed literature for searching. |
| Description of Main Sequence | 1. The doctor or nurse accesses the digital library. 2. The user enters search criteria (keywords, topics, author, publication date, etc.). 3. The system retrieves and displays relevant literature. 4. The user reviews and selects the desired material. 5. The user can bookmark or download selected literature. 6. The process is completed successfully. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> □ If no results match the search, the system suggests related materials. □ If the system experiences an error, it logs the issue and alerts the administrator. □ If the user does not have the necessary permissions, access is denied. |

| | |
|-----------------------------|---|
| Non Functional Requirements | <ul style="list-style-type: none"> - Performance : NF REQ-08 : The system must return search results within 3 seconds. - Maintainability: The library database should be regularly updated. - Usability: NF REQ-01 : The interface should be userfriendly and support advanced search capabilities. - Security: The system must ensure secure access to licensed content. |
| | <ul style="list-style-type: none"> - Manageability: The library manager must be able to add, update, and delete literature records. |
| Postconditions | <ul style="list-style-type: none"> □ The user successfully retrieves and accesses relevant literature. □ The system remains ready for further searches. □ Any bookmarks or downloads are stored for future reference. |

Author: Eglis Braho

| UC Name | UC-23 Library and Literature Management |
|------------|---|
| Summary | The system offers a comprehensive library management feature, enabling the Library Manager to oversee medical literature, journals, research papers, and reference materials. It allows adding, updating, deleting entries, uploading files, and maintaining an up-to-date database of medical resources. |
| Dependency | None |
| Actors | <ul style="list-style-type: none"> • Primary Actor: Library Manager • Secondary Actors: System Administrator (for system maintenance and managing permissions), Other Library Staff (for assisting with day-to-day tasks) |

| | |
|----------------------------------|---|
| Preconditions | <ul style="list-style-type: none"> • The Library Manager must have proper login credentials to access the library management system. • The system must be connected to the database for storing and retrieving literature data. • File upload capability must be enabled and functioning correctly. |
| Description of the Main Sequence | <ol style="list-style-type: none"> 1. The Library Manager navigates to the Library Management Module. 2. The system presents options to Add New Literature, Update Existing Literature, or Delete Literature. 3. If adding new literature: <ul style="list-style-type: none"> • The Library Manager enters details like title, author, publication date, category, keywords, and description. • The Library Manager uploads a file (PDF version, research document, or link to an online resource) to the system. • The system saves the new literature entry into the database. 4. If updating existing literature: <ul style="list-style-type: none"> • The Library Manager selects a piece of literature to update. • The system presents the current details, allowing the Library Manager to replace the outdated file or append supplementary information. • The updated literature is saved to the system. 5. If deleting literature: |
| | <ul style="list-style-type: none"> • The Library Manager selects the outdated or irrelevant piece of literature to remove. • The system prompts for confirmation before deleting the record and associated files. • The literature is deleted from the system and the database is updated. |

| | |
|---|---|
| Description of the Alternative Sequence | <p>Step 3-If the file upload fails (like large file size or incompatible format), the system will notify the Library Manager and allow them to retry with a valid file.</p> <p>Step 2-In case of system downtime, the Library Manager may not be able to access the system to make changes. The system will notify the Library Manager of downtime, and the tasks will resume when the system is back online.</p> |
| Non functional requirements | <ul style="list-style-type: none"> • NF-REQ-01 (Usability): The system shall provide an intuitive, user-friendly interface for managing literature, minimizing the need for extensive training for the Library Manager. • NF-REQ-03 (Performance): The system shall allow for quick literature retrieval, with search functionality that can return results in less than 2 seconds. • NF-REQ-06 (Performance): The system shall support the upload and retrieval of large files (PDFs, research documents) without significant performance degradation. • NF-REQ-11 (Availability): The system shall ensure a 99.9% uptime to allow the Library Manager continuous access for adding and updating literature. • NF-REQ-19 (Security): The system shall require role-based authentication to ensure only authorized personnel can add, update, or delete literature. • NF-REQ-20 (Security): The system shall ensure data encryption for any uploaded files to maintain confidentiality and prevent unauthorized access. |
| Postconditions | <ul style="list-style-type: none"> ▪ The system updates the database with the newly added, updated, or deleted literature. ▪ All changes (additions, updates, deletions) are logged for auditing purposes. |
| | <ul style="list-style-type: none"> ▪ The system reflects the current and accurate status of the medical literature available in the library. |

Author: Arjan Muka

| UC Name | UC-24 Vaccination Management |
|------------------------------|--|
| Summary | <p>The hospital system enables patients to manage their vaccination records by viewing their vaccination history, scheduling new vaccinations, and receiving reminders for upcoming vaccinations. The system ensures that patients can easily access their vaccination information and stay up-to-date with their immunization schedule.</p> |
| Dependency | - |
| Actors | <ul style="list-style-type: none">● Primary Actor: Patient● Secondary Actor: Nurses |
| Preconditions | <ul style="list-style-type: none">● The patient must have an existing medical profile in the system with vaccination records.● The patient must have access to the hospital system (like via a patient portal or mobile app).● The system must be operational and accessible to the patient.● The system must have an up-to-date vaccination schedule and reminder settings configured. |
| Description of Main Sequence | <ol style="list-style-type: none">1. The patient logs into the hospital system (like through a patient portal or mobile app).2. The patient selects the "Vaccination Management" module.3. The patient chooses to view their vaccination history, and the system displays a list of all previous vaccinations, including dates and types of vaccines received.4. The patient selects the option to schedule a new vaccination, and the system displays available vaccination types and recommended dates based on the patient's age, medical history, and vaccination schedule. |

| | |
|-------------------------------------|---|
| | <p>5. The patient selects a vaccination type and a preferred date, and the system schedules the vaccination appointment.</p> <p>6. The system sets up a reminder for the upcoming vaccination and stores the scheduled appointment in the patient's profile.</p> <p>7. The system sends a reminder to the patient (like via email or SMS) a set number of days before the scheduled vaccination date.</p> <p>8. After the vaccination is administered (handled by a separate use case involving medical staff), the system updates the patient's vaccination history with the new record.</p> |
| Description of Alternative Sequence | <p>1. Step 2: If the system fails to retrieve the patient's profile, an error is displayed, and the patient is prompted to contact the IT administrator.</p> <p>2. Step 5: If the system fails to schedule the vaccination, an error is logged, and the IT administrator is notified.</p> <p>3. Step 7: If the system fails to send the vaccination reminder, an error is logged, and the IT administrator is notified.</p> |
| Non Functional Requirements | <ul style="list-style-type: none"> ● NF-REQ-08: Vaccination history must be retrieved in under 2 seconds for quick access by patients. ● NF-REQ-20: Secure role-based access must be enforced to ensure patients can only view and manage their own vaccination records. ● NF-REQ-33: Automated alerts via email and SMS must be sent to IT personnel in case of failures (like missed reminders). ● NF-REQ-34: Multi-language support must be provided to ensure usability for diverse patients. |
| Postconditions | <ul style="list-style-type: none"> ● The patient can view their up-to-date vaccination history. |

| | |
|--|--|
| | <ul style="list-style-type: none"> • A new vaccination appointment is successfully scheduled and stored in the system. • The patient receives a reminder for the upcoming vaccination. • The system ensures that all actions (viewing history, scheduling, and sending reminders) are logged for audit purposes. • The system is ready for the next vaccination management task. |
|--|--|

Author: Eglis Braho

| UC Name | UC-25 Resource Allocation |
|---------------|--|
| Summary | The system facilitates efficient allocation and tracking of hospital resources, including medical staff, equipment, and supplies, with realtime visibility into their availability, location, and status. It enables managers to assign resources to departments or cases as needed, tracks inventory levels, and ensures timely restocking to maintain smooth operations. |
| Dependency | - |
| Actors | <ul style="list-style-type: none"> • Primary Actor: Inventory Manager |
| Preconditions | <ul style="list-style-type: none"> • All hospital resources (beds, equipment, staff) must be registered and tracked within the system. • The system is integrated with real-time resource tracking (for beds, equipment, etc.). |

| | |
|----------------------------------|--|
| Description of the Main Sequence | <ol style="list-style-type: none"> 1. The Inventory Manager navigates to the Resource Allocation Module. 2. The system displays a real-time overview of available hospital resources, including medical devices, beds, and wheelchairs. 3. The Inventory Manager selects a resource (like medical device) to allocate to a specific department or patient. 4. The system checks resource availability and confirms the allocation. 5. The Inventory Manager assigns the resource to the relevant department or case (like ICU, surgery, emergency). 6. The system updates the resource status to reflect its allocation. 7. The system tracks inventory levels of supplies (like surgical gloves, medications) and provides alerts when restocking is required. 8. The Inventory Manager can reallocate or deallocate resources as needed, with the system automatically adjusting the inventory and status accordingly. |
| Description of the Alternative | <p>Step 4-If the requested resource is unavailable, the system</p> |
| Sequence | <p>will notify the Inventory Manager and suggest alternative resources.</p> <p>Step 7-If supplies (like medications or disposables) are low, the system alerts the Inventory Manager, who can reorder stock to avoid disruptions.</p> <p>Step 2-In case of downtime, the Inventory Manager will be notified, and resource allocations will be temporarily suspended until the system is restored.</p> |
| Non functional requirements | <ul style="list-style-type: none"> - NF-REQ-06 (Performance): The system must provide realtime updates on resource availability and status with minimal delays, ensuring smooth operations during resource allocation. - NF-REQ-11 (Availability): The system shall maintain 99.9% uptime, ensuring hospital resources are always accessible for allocation, especially during critical periods. |

| | |
|----------------|--|
| | <ul style="list-style-type: none"> ▪ NF-REQ-19 (Security): The system must require role-based authentication to ensure that only authorized personnel (Inventory Manager, Department Heads) can manage resource allocations. ▪ NF-REQ-10 (Performance): The system should provide quick response times for resource status queries and allocation requests, aiming for 95% of interactions to be completed in less than 1 second. ▪ NF-REQ-17 (Data Backup Requirement): The system shall automatically back up resource allocation data every 15 minutes to prevent data loss. ▪ NF-REQ-20 (Security): The system shall log all allocation and deallocation actions to maintain an audit trail for security and accountability. |
| Postconditions | <ul style="list-style-type: none"> ▪ The system updates the resource inventory to reflect current allocations. ▪ The system logs all resource allocation actions for audit and tracking purposes. ▪ The system tracks inventory levels and alerts the Inventory Manager when restocking is required. |

Author: Arjan Muka+ Nikola Rigo

| UC Name | UC-26 Inventory Item & Procurement Management |
|---------------|---|
| Summary | <p>The hospital system enables inventory managers to manage inventory items, submit purchase requests, and track deliveries, while procurement officers verify invoices and send offers to the board for approval. The board (managers and admins) reviews offers, approves the best choice, and approves, modifies, or rejects offers. The system ensures efficient inventory management, timely procurement, and accurate tracking of supplies.</p> |
| Dependency | |
| Actors | <ul style="list-style-type: none">• Primary Actors: Inventory Manager, Procurement Officer, Board (Manager)• Indirect Actor: Supplier |
| Preconditions | <ul style="list-style-type: none">• The system must have an up-to-date inventory of items, including current stock levels.• The inventory manager, procurement officer, and board members must have appropriate permissions to access the inventory management system.• The system must have a list of approved suppliers and their contact details.• The system must be operational and accessible to all actors. |

| | |
|-------------------------------------|--|
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The inventory manager logs into the hospital system and selects the "Inventory Management" module. 2. The inventory manager manages inventory items by checking current stock levels and identifying items that need restocking. 3. The inventory manager submits a purchase request for the required items, specifying quantities and preferred suppliers. |
| | <ol style="list-style-type: none"> 4. The system notifies the procurement officer of the new purchase request (via an internal notification). 5. The procurement officer logs into the system, reviews the purchase request, and views the list of available suppliers. 6. The procurement officer sends offers to the board for approval, including details of the items, quantities, and supplier quotes. 7. The board (manager or admin) logs into the system, reviews the offers, and approves the best choice based on cost, quality, and delivery timelines. 8. If the board approves the offer, the procurement officer verifies the invoices from the supplier and confirms the order. 9. The inventory manager tracks the delivery of the items in the system, updating the inventory once the items are received. 10. The system logs all actions (purchase request, offer approval, invoice verification, and delivery tracking) for audit purposes. |
| Description of Alternative Sequence | <p>Step 4: If the system fails to notify the procurement officer of the purchase request, an error is logged, and the IT administrator is notified.</p> <p>Step 6: If the system fails to send offers to the board, an error is logged, and the IT administrator is notified.</p> <p>Step 10: If the system fails to log actions (like offer approval), an error is logged, and the IT administrator is notified.</p> |

| | |
|-----------------------------|--|
| Non Functional Requirements | <ul style="list-style-type: none"> ● NF-REQ-08: Inventory records must be retrieved in under 2 seconds for efficient management. ● NF-REQ-20: Secure role-based access must be enforced to ensure only inventory managers, procurement officers, and board members can perform their respective tasks. ● NF-REQ-18: The system must log all errors (like failed notifications) and notify administrators for immediate resolution. ● NF-REQ-26: Customizable workflows must be supported to adapt to different hospital procurement policies. |
| Postconditions | <ul style="list-style-type: none"> ● The inventory manager successfully submits a purchase request for required items. ● The procurement officer sends offers to the board, and the board approves the best choice. ● The procurement officer verifies invoices, and the order is placed with the supplier. ● The inventory manager tracks the delivery and updates the inventory upon receipt of items. ● The system ensures that inventory levels are accurately updated and reflected in real time. ● All actions (purchase request, offer approval, invoice verification, and delivery tracking) are logged for audit purposes. ● The system is ready for the next inventory management task. |

Author: Marin Tartaraj

| UC Name | UC-27 Supplier Management |
|------------------------------|---|
| Summary | This use case allows the Inventory Manager to manage a list of suppliers by adding, updating, or removing supplier details, while preserving historical data for transparency. Modifications are logged for accountability. |
| Dependency | |
| Actors | Procurement Officer |
| Preconditions | <ol style="list-style-type: none">1. The Procurement Officer must be logged into the system.2. The system must be online and accessible.3. The manager must have sufficient permissions to manage supplier data. |
| Description of Main Sequence | <ol style="list-style-type: none">1. The Procurement Officer selects "Supplier Management" from the menu.2. The system displays the list of suppliers.3. The manager can choose to add, update, or remove supplier details such as name, contact information, and company details.4. If adding, the system prompts the manager to enter the new supplier's details.5. If updating, the system displays the current supplier information for modification.6. If removing, the system asks for confirmation before deleting the supplier record.7. The system validates the input to ensure correctness and completeness.8. The system logs the action (add, update, or remove) along with the user and timestamp for accountability.9. The system updates the supplier list and preserves historical data for each inventory item. |

| | |
|-------------------------------------|---|
| Description of Alternative Sequence | <p>3a. If the manager attempts to update or remove a non-existent supplier, the system displays an error message.</p> <p>4a. If the manager enters missing or incorrect data, the</p> |
| | <p>system displays an error message and prompts for corrections.</p> <p>6a. If the manager cancels the removal confirmation, the system aborts the deletion process.</p> |
| Non Functional Requirements | <ul style="list-style-type: none"> • NF-REQ-01 (Usability Requirement): A user-friendly interface helps Inventory Managers manage suppliers effectively. • NF-REQ-02 (Usability Requirement): Documentation and tutorials should support supplier management tasks. • NF-REQ-06 (Performance Requirement): The system must process supplier transactions efficiently, supporting 1000+ concurrent users. • NF-REQ-07 (Performance Requirement): Supplier transactions should be processed at a rate of at least 500 per second. • NF-REQ-09 (Performance Requirement): Quick database queries (200ms response time) for supplier information retrieval. • NF-REQ-10 (Performance Requirement): 95% of interactions (adding/updating suppliers) should complete within 1 second. • NF-REQ-15 (Data Backup Requirement): Supplier data should be backed up every 15 minutes to prevent data loss. • NF-REQ-16 (Data Backup Requirement): Backups should be stored in multiple locations for disaster recovery. • NF-REQ-18 (System Error Handling Requirement): The system should log errors and notify administrators if supplier management fails. |

| | |
|----------------|--|
| | <p>.. NF-REQ-27 (System Downtime Management Requirement): Real-time notifications should inform users if supplier management features are affected.</p> |
| Postconditions | <ol style="list-style-type: none"> 1. If successful, the system updates the supplier list and preserves historical records for each inventory item. 2. If unsuccessful, the system displays an error message and prompts the manager to correct the issue. |

Author: Eglis Braho

| UC Name | UC-28 Ambulance Management |
|---------------|--|
| Summary | The system enables receptionists to track ambulance availability and location in real-time, displaying their status as available, on the road, or occupied. Using GPS, it dynamically updates locations to improve emergency response and patient transport coordination. |
| Dependency | None |
| Actors | <ul style="list-style-type: none"> • Primary Actor: Receptionist • Indirect Actors: System Administrator, Ambulance Driver |
| Preconditions | <ul style="list-style-type: none"> • The system must be connected to the GPS tracking module. • Ambulance status and location updates must be active and synchronized. • The receptionist must have login credentials to access ambulance tracking. |

| | |
|---|--|
| Description of the Main Sequence | <ol style="list-style-type: none"> 1. The receptionist navigates to the Ambulance Management module. 2. The system displays the real-time status of ambulances (Available, On Route, Occupied). 3. For occupied ambulances, the system fetches and displays GPS location. 4. If an ambulance arrives at the hospital, its status updates automatically to "Available." 5. The receptionist monitors and updates dispatch details as needed. |
| Description of the Alternative Sequence | <p>Step 4- if the GPS tracking fails, the system notifies the administrator and defaults to last known location.</p> <p>Step 3- if the System goes down, the receptionist manually records ambulance status and updates once the system is restored.</p> <p>Step1-if the system catches an unauthorized access, the system prevents users from accessing ambulance data.</p> |
| Non functional requirements | <ol style="list-style-type: none"> 1. Performance <ul style="list-style-type: none"> - NF-REQ-08: The system shall retrieve ambulance location data in less than 2 seconds, ensuring quick access to real-time information. |
| | <ol style="list-style-type: none"> - NF-REQ-09: The system shall maintain an average response time of 200 milliseconds for all database queries related to ambulance status and location, ensuring near-instantaneous access to tracking data. 2. Security <ul style="list-style-type: none"> - NF-REQ-19: The system shall require multi-factor authentication for staff to access ambulance tracking data, ensuring that only authorized personnel can interact with sensitive information. - NF-REQ-20: The system shall implement role-based access control (RBAC) to limit access to ambulance tracking and status updates based on user roles, ensuring that only receptionists and authorized users can modify ambulance details. 3. Availability <ul style="list-style-type: none"> - NF-REQ-12: The system shall ensure 99.9% uptime, guaranteeing continuous availability of the ambulance tracking system, especially during critical emergency situations. |

| | |
|----------------|---|
| | <ul style="list-style-type: none"> NF-REQ-14: The system shall limit unscheduled downtime to no more than 1 hour per year, ensuring that ambulance tracking is always available for dispatch coordination. <p>4. Usability</p> <ul style="list-style-type: none"> NF-REQ-05: The system shall provide an intuitive user interface that allows receptionists to easily navigate and track ambulance status with minimal training. NF-REQ-06: The system shall include real-time visual indicators (such as color codes or icons) to clearly show the status of each ambulance (Available, On Route, Occupied). NF-REQ-07: The system shall provide easy-to-read notifications for any failures, such as GPS tracking issues or system downtimes, ensuring the receptionist can quickly address issues. |
| Postconditions | <ul style="list-style-type: none"> The system updates the status of ambulances based on their movements. The receptionist has accurate, real-time data on ambulance availability and location. Any issues (GPS failures, system downtime) are logged for review. |

Author: Arjan Muka

| | |
|------------|---|
| UC Name | UC-29 Permission Granting |
| Summary | <p>The hospital system enables admins to assign permissions to users, ensuring that hospital staff (like nurses, doctors, inventory managers, etc.) have appropriate access to system features based on their roles. The system ensures secure and controlled access to sensitive data and functionalities.</p> |
| Dependency | |

| | |
|-------------------------------------|---|
| Actors | Primary Actor: Admin |
| Preconditions | <ul style="list-style-type: none"> • The admin must have appropriate permissions to access the permission granting module. • The system must have a list of existing user accounts and predefined roles (like nurse, doctor, inventory manager, etc.). • The system must be operational and accessible to the admin. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The admin logs into the hospital system using their credentials. 2. The admin selects the "Permission Granting" module from the system dashboard. 3. The system displays a list of users and their current roles and permissions. 4. The admin selects a user (like a nurse) to assign or update permissions. 5. The admin assigns permissions to the user by selecting a role (like "Nurse") or specific permissions (like access to bed management, ability to order lab tests). 6. The system updates the user's profile with the new permissions and logs the change for audit purposes. 7. The system notifies the user (like via email or system notification) of the updated permissions. 8. The admin verifies that the permissions have been correctly applied by checking the user's updated profile. |
| Description of Alternative Sequence | <ol style="list-style-type: none"> 1. Step 3: If the system fails to retrieve the user's profile, an error is displayed, and the admin is prompted to contact the IT administrator. |

| | |
|-----------------------------|---|
| | <ol style="list-style-type: none"> 2. Step 6: If the system fails to update the user's permissions, an error is logged, and the IT administrator is notified. 3. Step 7: If the system fails to notify the user of updated permissions, an error is logged, and the IT administrator is notified. 4. If the system experiences downtime during the process, the admin can record the permission changes offline, and the data is synced once the system is back online (NF-REQ-28). 5. If the admin accidentally assigns incorrect permissions, they can immediately revoke or modify the permissions by repeating steps 4-6. |
| Non Functional Requirements | <ul style="list-style-type: none"> • NF-REQ-08: User records must be retrieved in under 2 seconds for efficient permission assignment. • NF-REQ-20: Secure role-based access must be enforced to ensure only admins can assign permissions. • NF-REQ-19: Multi-factor authentication must be required for admins to access the permission granting module. • NF-REQ-18: The system must log all errors (like failed permission updates) and notify administrators for immediate resolution. |
| Postconditions | <ul style="list-style-type: none"> • The user's permissions are successfully updated in the system. • The user is notified of the updated permissions. • The system ensures that the user can now access the appropriate features based on their new permissions. • All actions (permission assignment, updates, and notifications) are logged for audit purposes. • The system is ready for the next permission granting task. |

Author: Shpetim Shabanaj

| UC Name | UC-30 Visitor Management |
|-------------------------------------|--|
| Summary | This use case allows receptionists to register visitors for hospitalized patients and record visit details in the patient's portfolio. |
| Dependency | - |
| Actors | Primary Actor: Receptionist |
| Preconditions | The required patient must be hospitalized and have an active medical portfolio. |
| Description of Main Sequence | <ol style="list-style-type: none"> 1. The receptionist shall navigate to "Visitor Management" feature. 2. The receptionist searches for the hospitalized patient by name or ID. 3. The system displays the patient's details and current visitor logs. 4. The receptionist enters visitor information, including: <ul style="list-style-type: none"> - Visitor's full name - Contact information - Relationship to the patient - Check-in time 5. The system verifies the visitor's details and checks for any restrictions on visiting hours or visitor limits. 6. If all conditions are met, the system registers the visitor and updates the patient's portfolio with visit details. 7. Upon visitor departure, the receptionist records the check-out time in the system. 8. The system then displays the visit details and duration. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> • Step 3: If the patient has exceeded the allowed number of visitors, the system notifies the receptionist and prevents further registrations. • Step 6: If there are any missing fields, there is displayed an error and the receptionist is redirected to form's page. |
| Non Functional Requirements | <ul style="list-style-type: none"> • NF-REQ-36: The system should provide a real-time visitor log for security and auditing purposes. • NF-REQ-20: Secure role-based access should restrict visitor data modification to authorized personnel. • NF-REQ-08: The visitor registration process should be completed in under 2 seconds to avoid reception delays. |
| Postconditions | <ul style="list-style-type: none"> • The visitor is successfully registered in the system. |

| | |
|--|--|
| | <ul style="list-style-type: none"> The patient's portfolio is updated with visit details. The system maintains a log of visitor check-ins and check-outs for security and tracking. |
|--|--|

Author: Artjol Zaimi

| | |
|-------------------------------------|---|
| UC Name | UC-31 Employee Management System (HR management) |
| Summary | This use case allows the HR Manager to add, update, view, and manage employee records, including personal information, job details, compensation, attendance, and skills. It helps keep staff data organized and up to date for HR operations. |
| Dependency | |
| Actors | Primary actor: HR Manager |
| Preconditions | <ul style="list-style-type: none"> The HR Manager is logged in with the correct role and access. The system is connected to the employee database. The employee management module is accessible from the HR dashboard. |
| Description of Main Sequence | <ol style="list-style-type: none"> HR Manager opens the "Employee Management" module. The system shows a searchable list of all employees. HR Manager selects one of the following actions: <ul style="list-style-type: none"> -Add Employee – enters name, ID, job title, salary, contact, skills, etc. -Update Employee – modifies existing records such as role, salary, or attendance. -Archive Employee – marks an employee as inactive (like resigned/retired). The system validates input and saves the changes. A success message is displayed, and the database is updated. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> Step 3: If required fields are missing or contain invalid data, the system highlights the errors and prevents submission. Step 4: If the HR Manager tries to add an employee with an existing ID, the system shows a duplicate warning. |
| Non Functional Requirements | <ul style="list-style-type: none"> Security: NF-REQ-21: Only HR Managers should have access to add, update, or archive employee data. Performance: NF-REQ-08: Employee records must be retrieved and updated in under 2 seconds. |

| | |
|----------------|---|
| | <ul style="list-style-type: none"> System Error Handling: NF-REQ-18: The system must log any failed update attempts or access errors. |
| Postconditions | <ul style="list-style-type: none"> Employee records are created, updated, or archived successfully. All actions are saved in the hospital database. Audit logs are updated with timestamps and user info. HR Manager sees updated employee data in real time. |

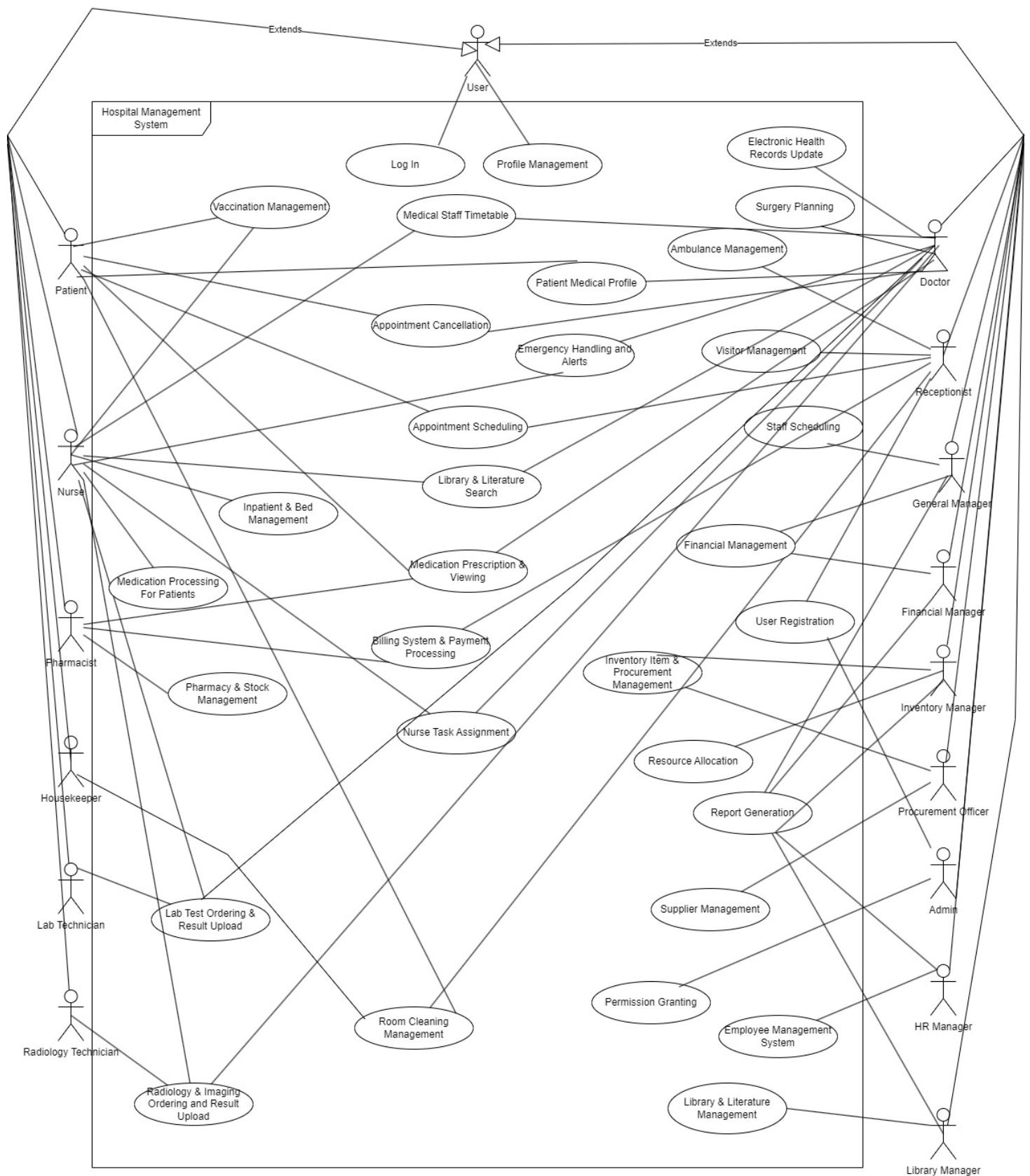
Author: Nikola Rigo

| UC Name | UC-32 Financial Management |
|---------------|--|
| Summary | Enables the Financial Manager to set and monitor departmental budgets, track expenditures, and maintain detailed financial logs for audit purposes. The system allows input of budget details (like total amounts, expense categories, department names), tracks real-time expenses (including HR expenses, inventory costs, etc.), calculates the implied surplus or shortage of funds, and logs all financial transactions for auditing. |
| Dependency | - |
| Actors | Primary Actor: Financial Manager |
| Preconditions | <ul style="list-style-type: none"> The Financial Manager must be authenticated and have the necessary permissions. The hospital's financial and expense data from various departments (HR, inventory, etc.) are available in the system. The financial management module is fully operational and integrated with other relevant systems. |

| | |
|-------------------------------------|--|
| Description of Main Sequence | <p>1. Budget Setup & Monitoring:</p> <ul style="list-style-type: none"> a. The Financial Manager logs into the system and navigates to the budget management section. b. The manager inputs budget details, such as total amounts, expense categories, and department names (either annually or by department). c. The system validates inputs and stores the budget details. <p>2. Expense Tracking & Calculation:</p> <ul style="list-style-type: none"> a. The system tracks expenses in real time from various hospital departments (like HR expenses, inventory costs). b. It calculates the overall expenditures and determines the implied surplus or shortage of funds. <p>3. Financial Logging & Audit:</p> <ul style="list-style-type: none"> a. All financial transactions (purchases, payments, budget updates) are logged automatically by the system. |
| | <ul style="list-style-type: none"> b. The Financial Manager can input or verify transaction details (type, amount, date, and relevant identifiers). c. The system validates and securely stores these logs for audit purposes. |
| Description of Alternative Sequence | <ul style="list-style-type: none"> - Data Entry Error: If the Financial Manager inputs incomplete or invalid budget or transaction details, the system prompts for corrections before saving. - Integration Disruption: If data from any department is temporarily unavailable, the system logs the missing data and notifies the manager for follow-up. - Audit Discrepancy: If discrepancies are found during audit logging, the system flags the issue for further investigation by the Financial Manager. |

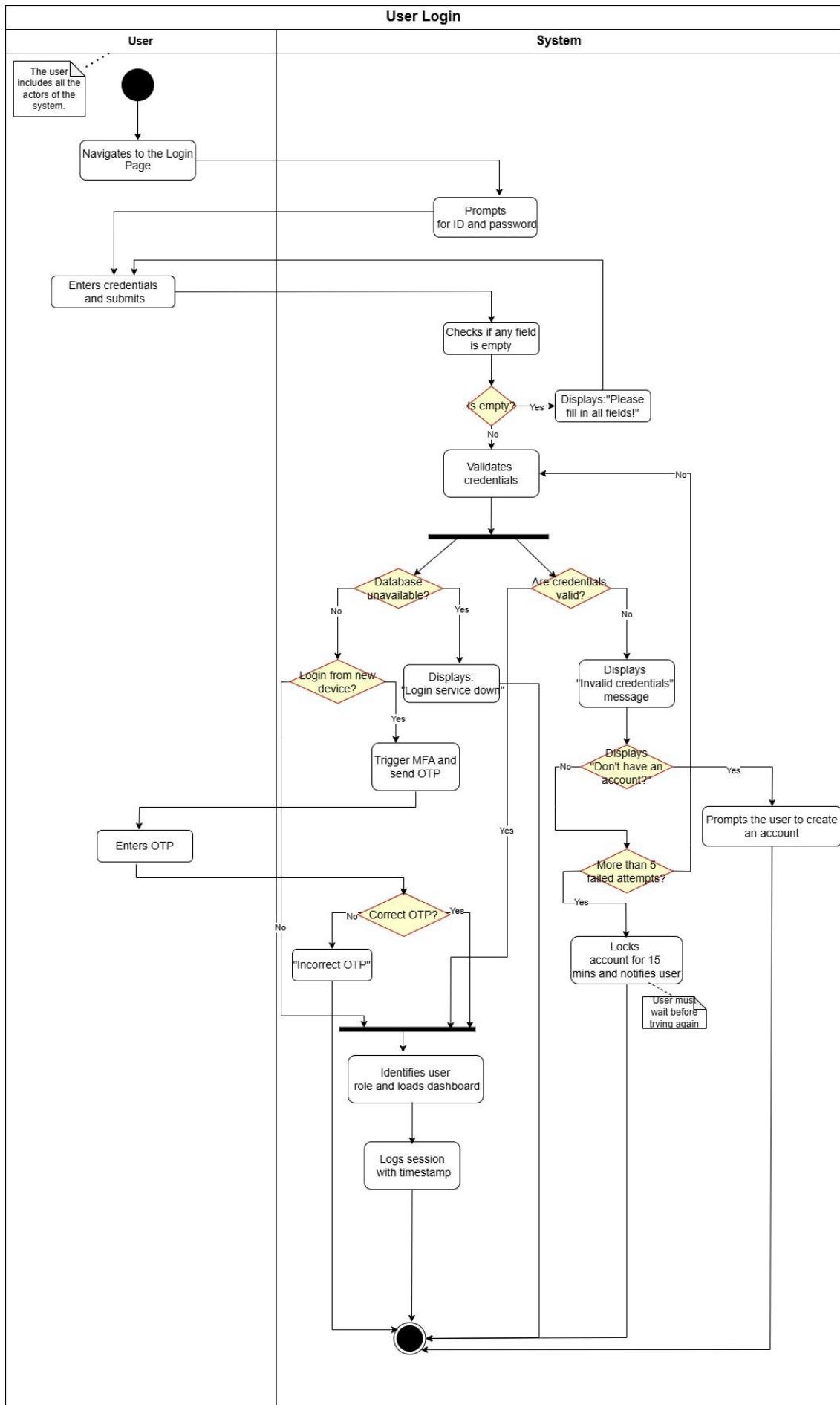
| | |
|-----------------------------|---|
| Non Functional Requirements | <ul style="list-style-type: none"> - NF REQ-01: Intuitive interface for financial managers to set budgets and track expenditures. - NF REQ-02: User documentation and tutorials to assist financial staff. - NF REQ-06: Notifications for budget overruns, high expenditures, or suspicious transactions. - NF REQ-07: The system must process at least 500 transactions per second. - NF REQ-09: Average response time for financial transactions should not exceed 200ms. - NF REQ-10: 95% of financial interactions should be completed within 1 second. - NF REQ-12: 99.9% uptime ensures continuous access to financial data. - NF REQ-15: Automatic backups every 15 minutes to prevent financial data loss. - NF REQ-16: Geographically distributed backups for disaster recovery. - NF REQ-17: Full data recovery within 30 minutes in case of corruption. - NF REQ-18: Logs all financial errors and notifies administrators. - NF REQ-19: Multi-factor authentication for financial staff. - NF REQ-20: Role-based access prevents unauthorized financial transactions. - NF REQ-21: Logs all failed login attempts for financial records access. |
| | <ul style="list-style-type: none"> - NF REQ-30: Maintains historical financial records for a minimum of 10 years. - NF REQ-33: Sends automated alerts for critical financial system failures. |
| Postconditions | <ul style="list-style-type: none"> - Department budgets are defined, stored, and actively monitored. - The system calculates and displays the surplus or shortage of funds based on real-time expense tracking. - All financial transactions are logged and stored, providing a comprehensive audit trail. - Financial reports and audit logs are updated and available for review by the Financial Manager. |

HMS Use-case Diagram

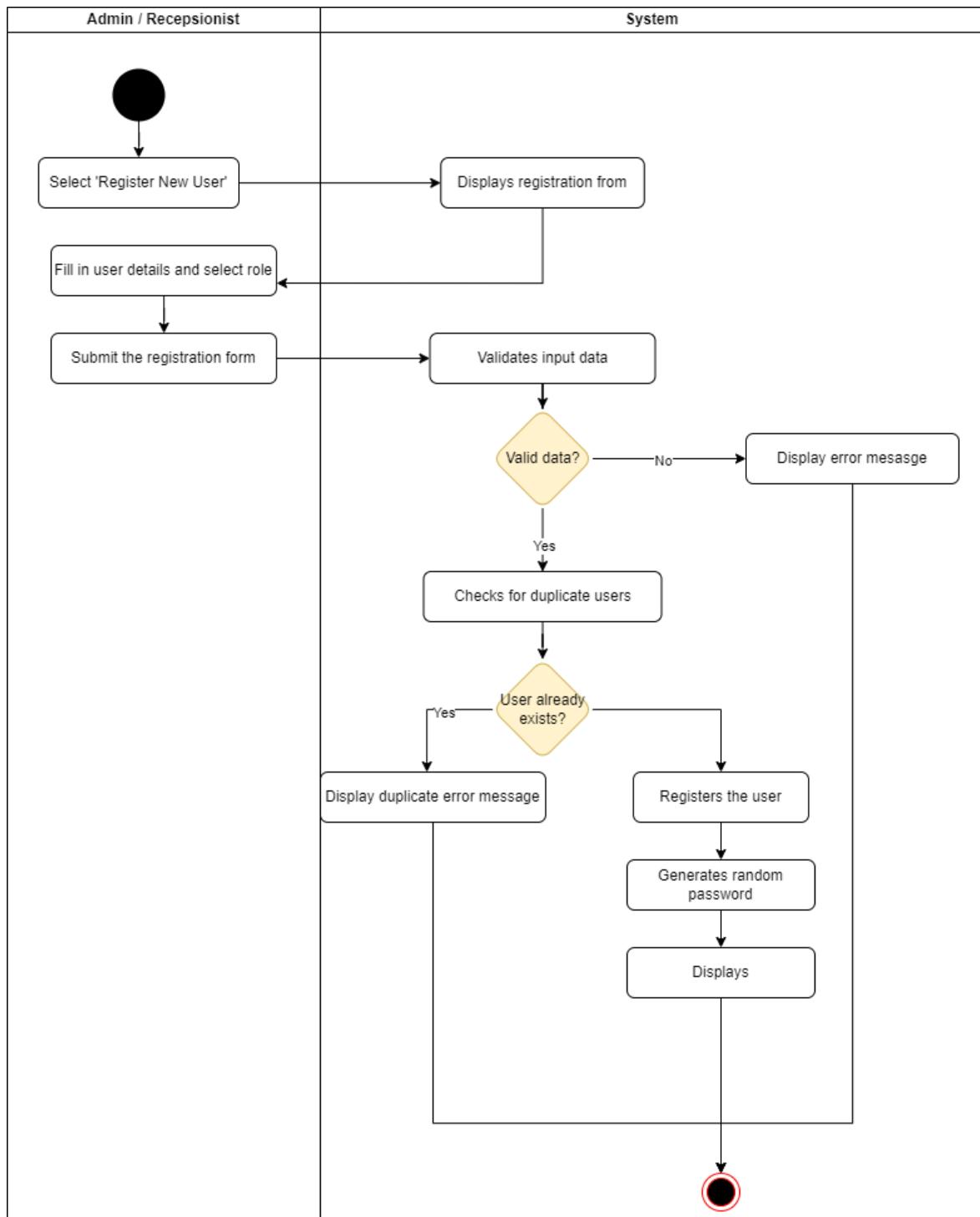


Subtopic: Activity Diagrams

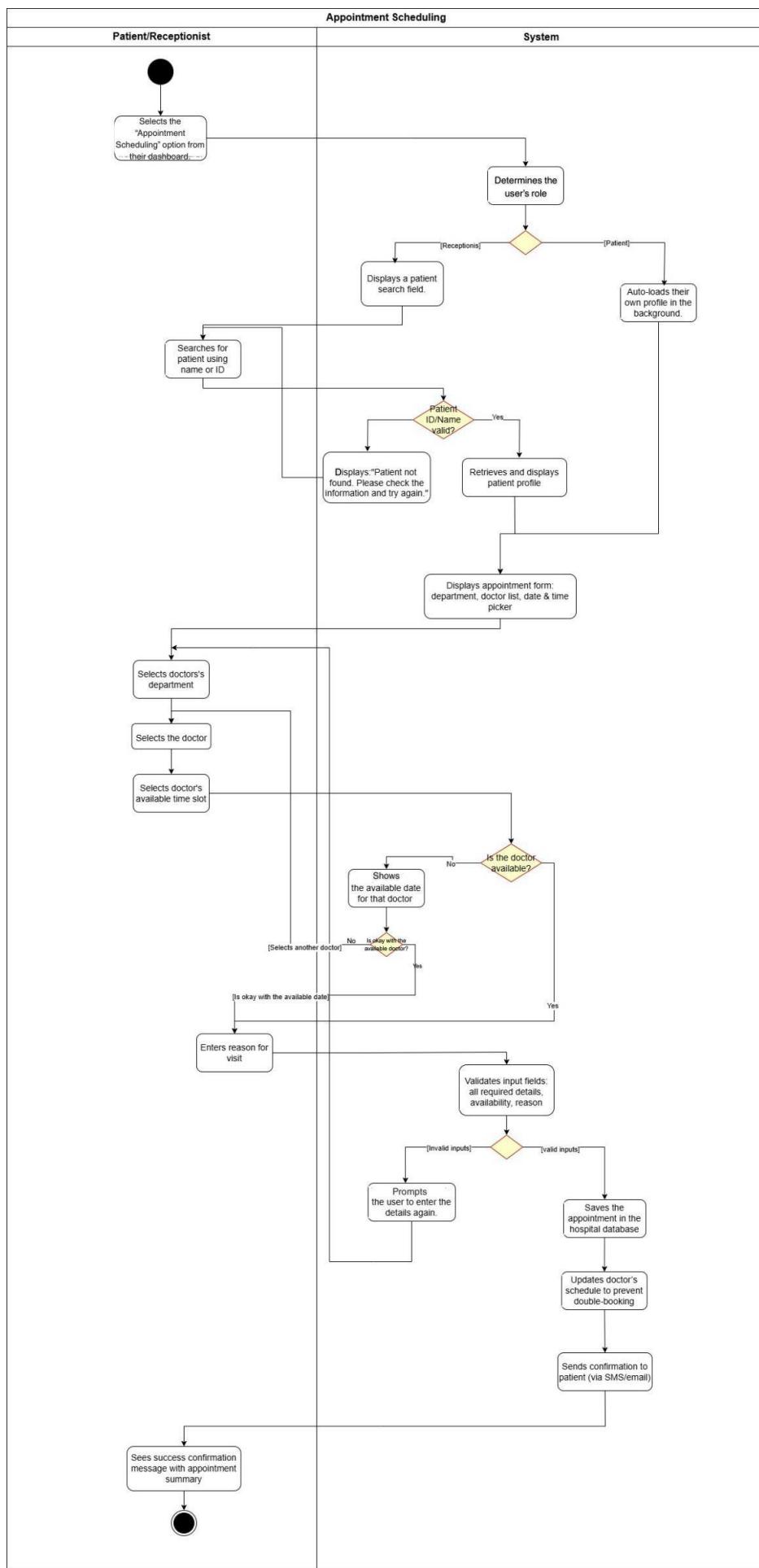
Author: Artjol Zaimi AD-01 User Login



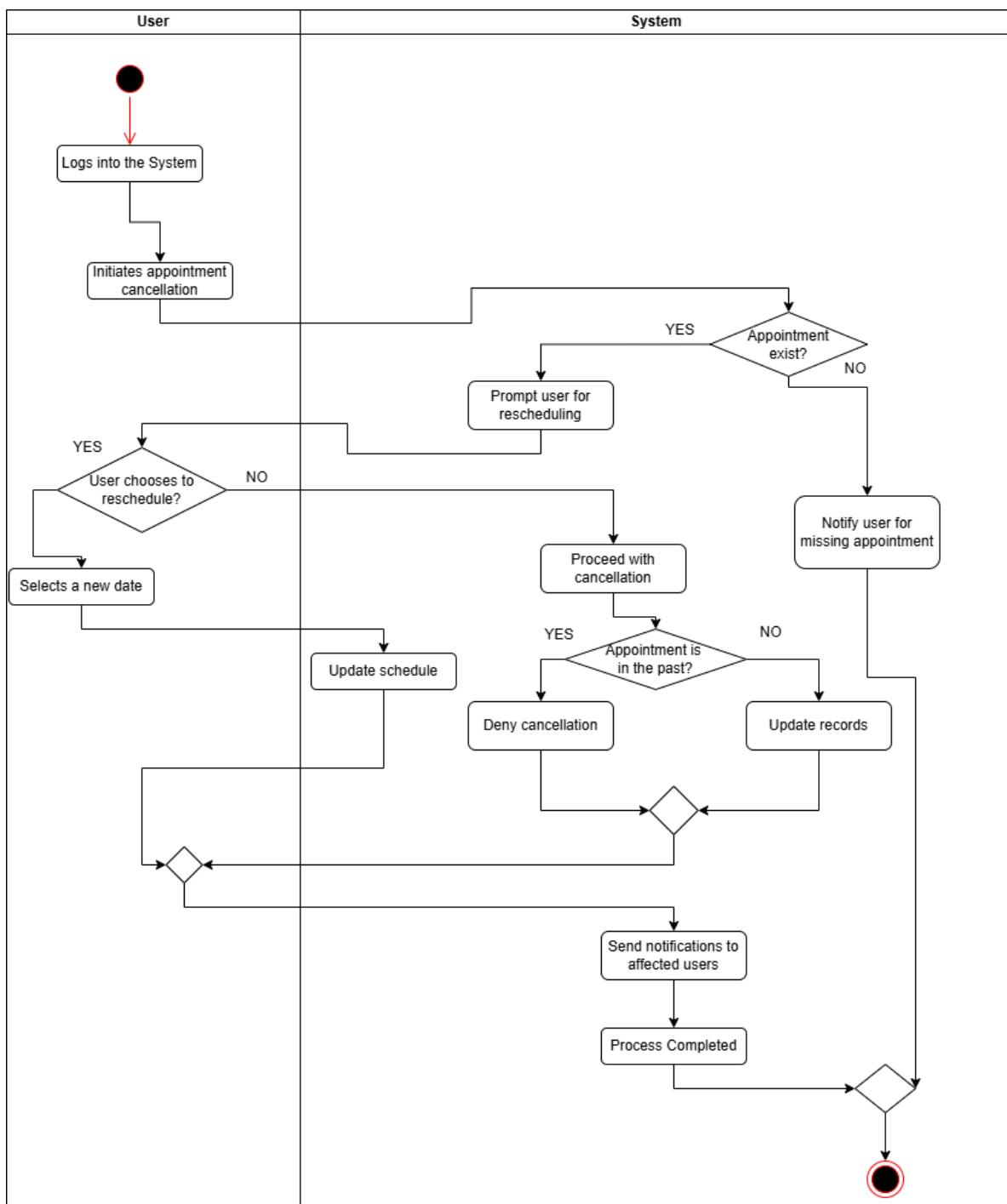
**Author: Marin Tartaraj AD-
02 Registration**



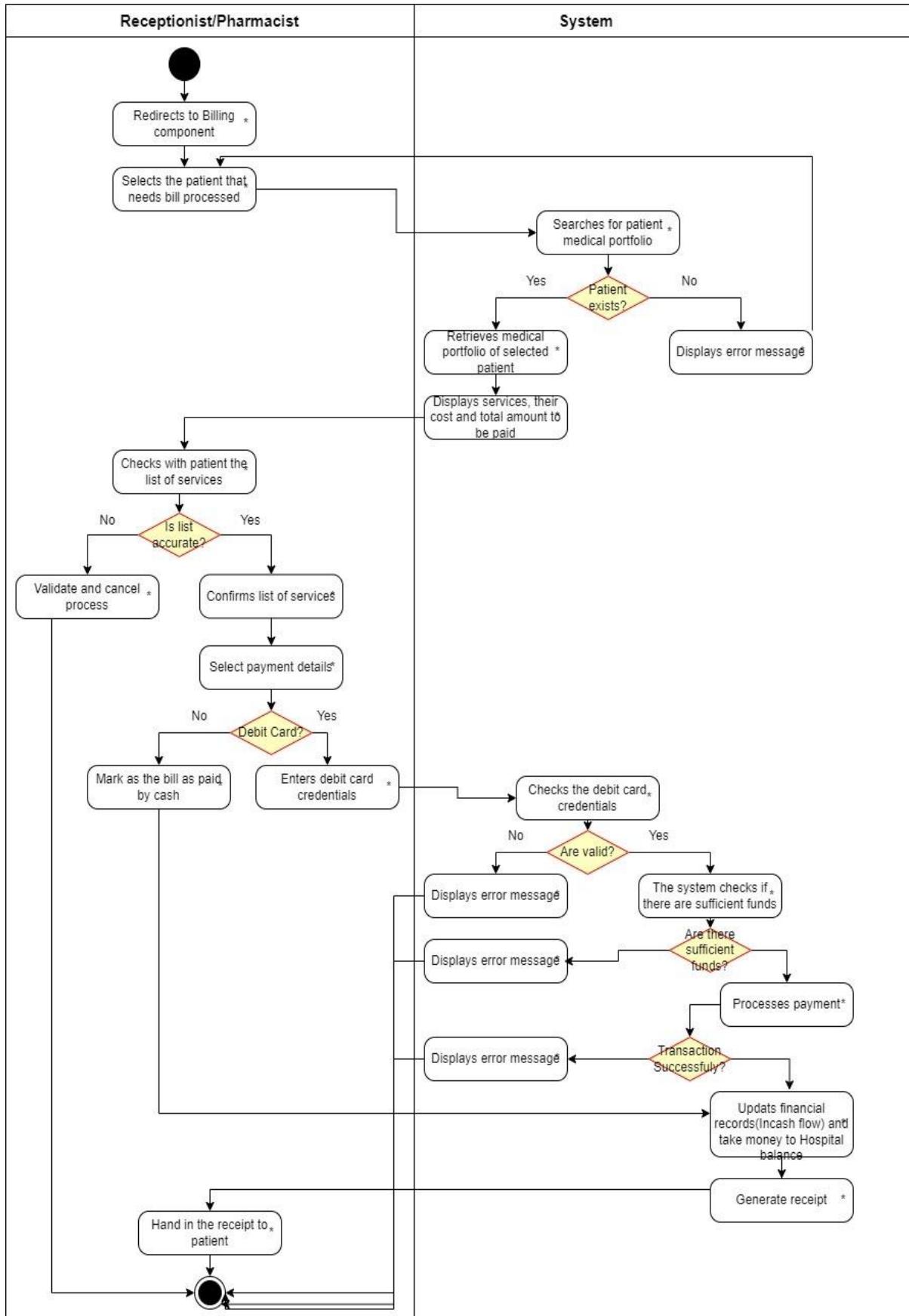
Author: Artjol Zaimi AD-03 Appointment Scheduling



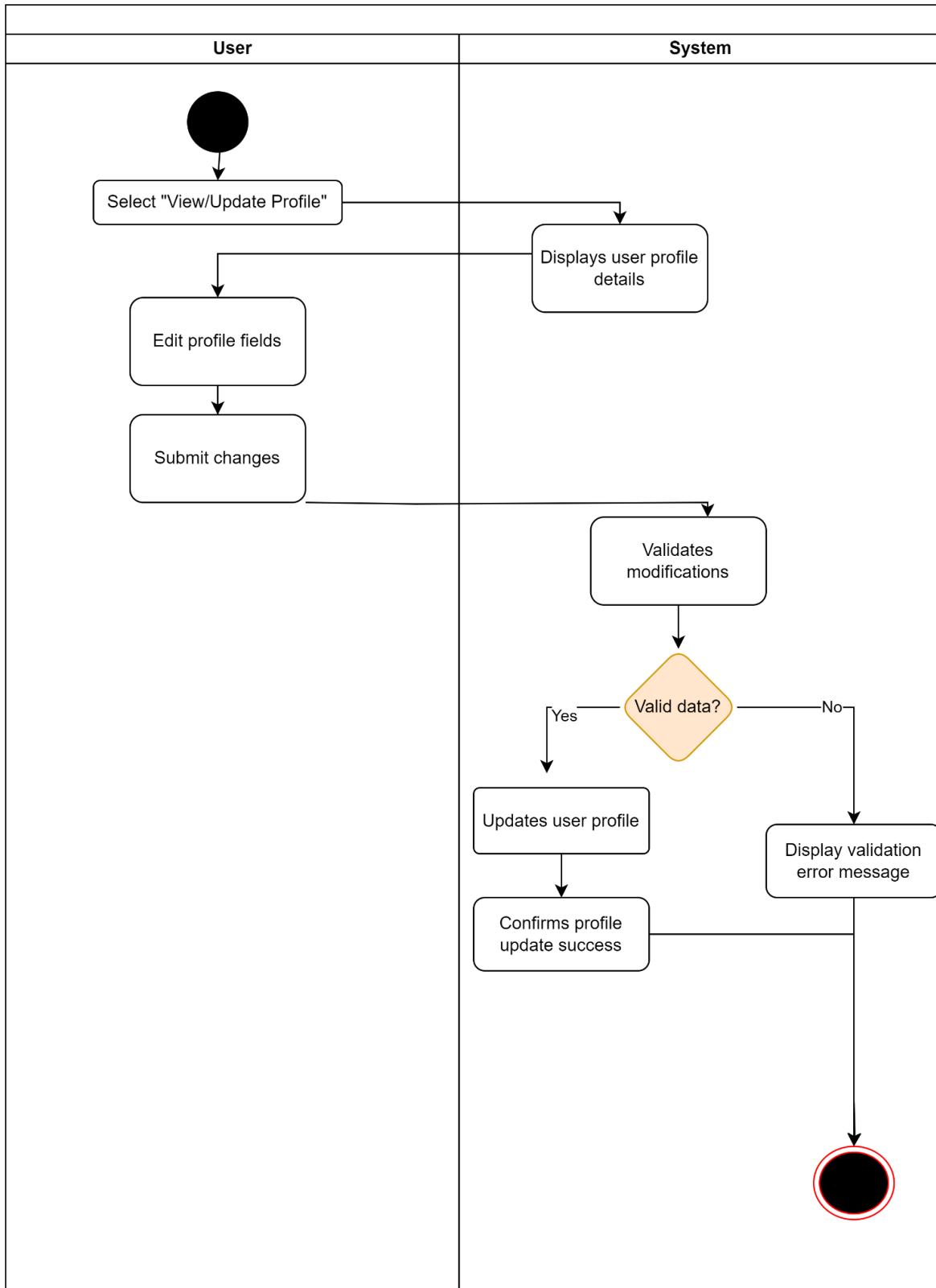
Author: Arlin Bashllari AD-04 Appointment Cancellation



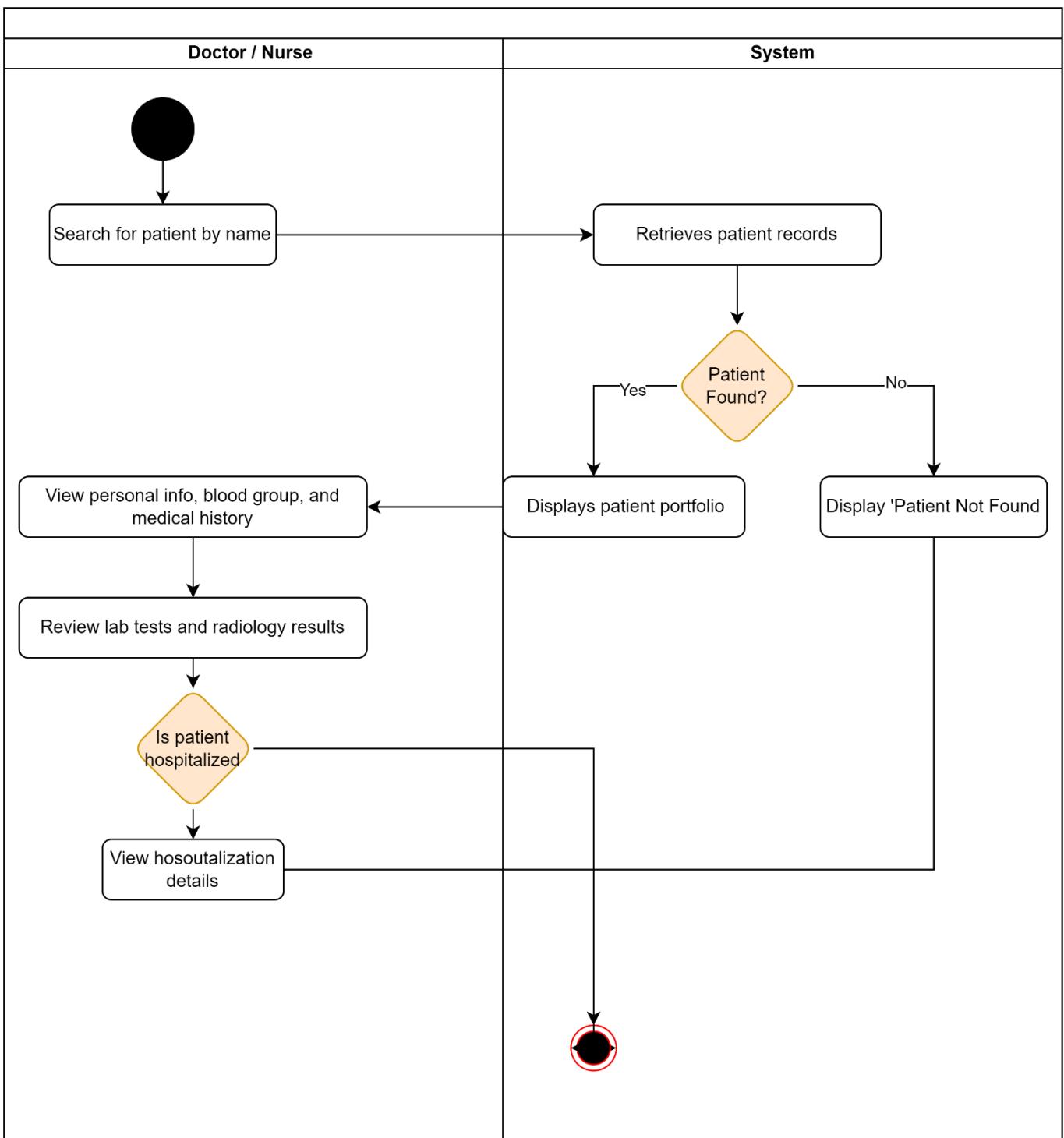
Author: Shpetim Shabanaj AD-05 Billing Component & Payment



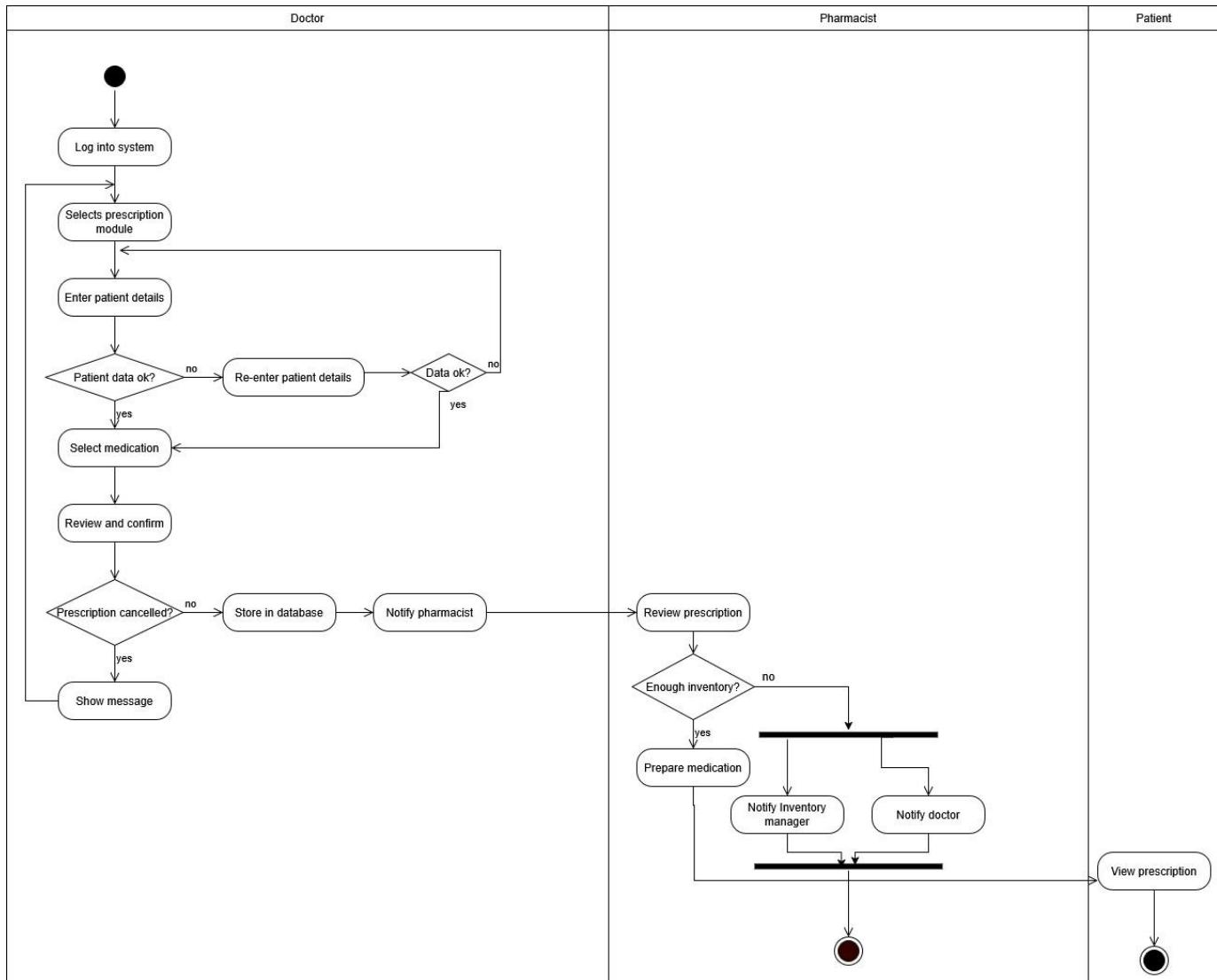
Author: Marin Tartaraj AD-06 Profile Management



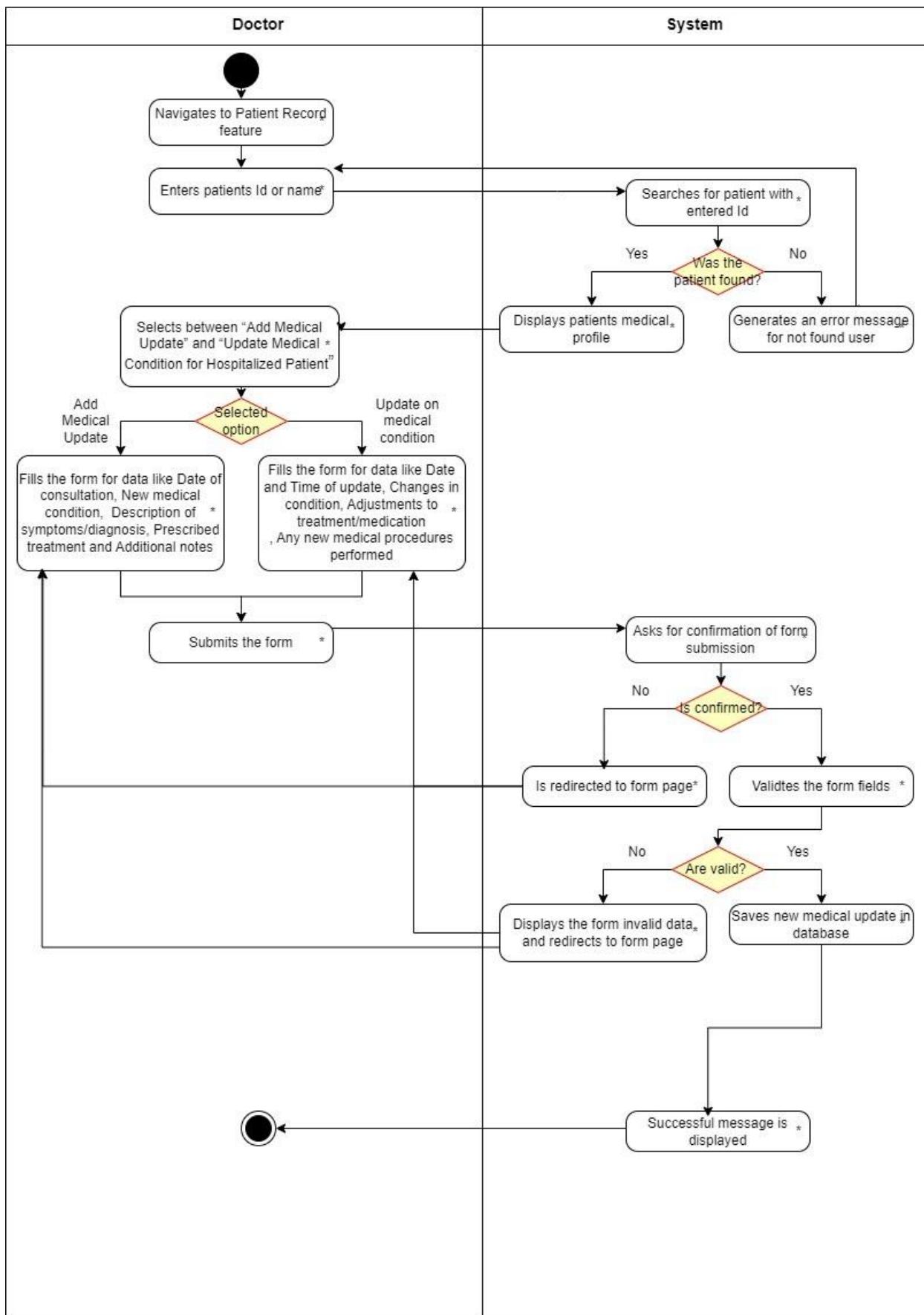
Author: Marin Tartaraj AD-07 Patient Medical Profile



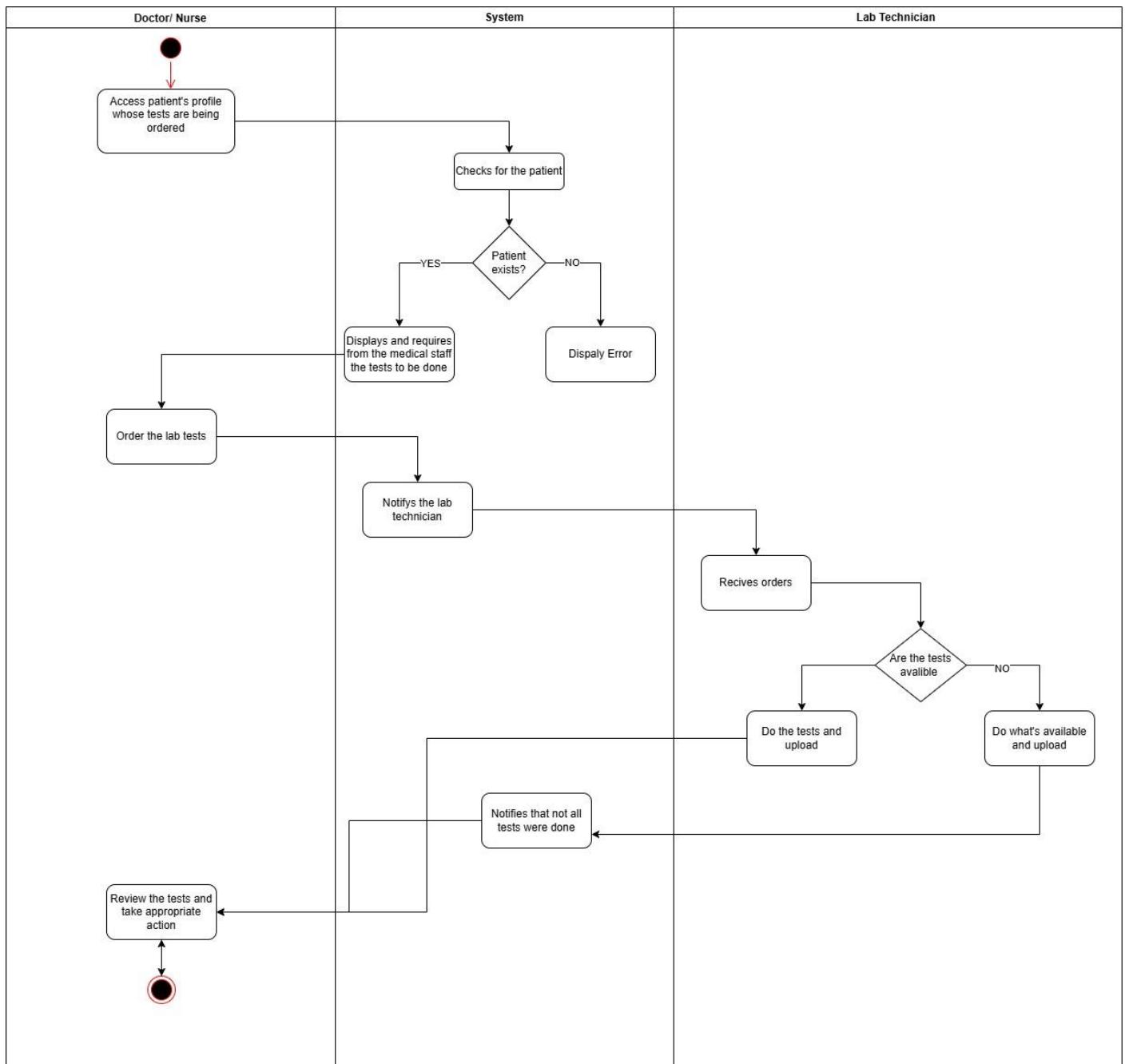
Author: Nikola Rigo AD -08 Medication Prescription & Viewing



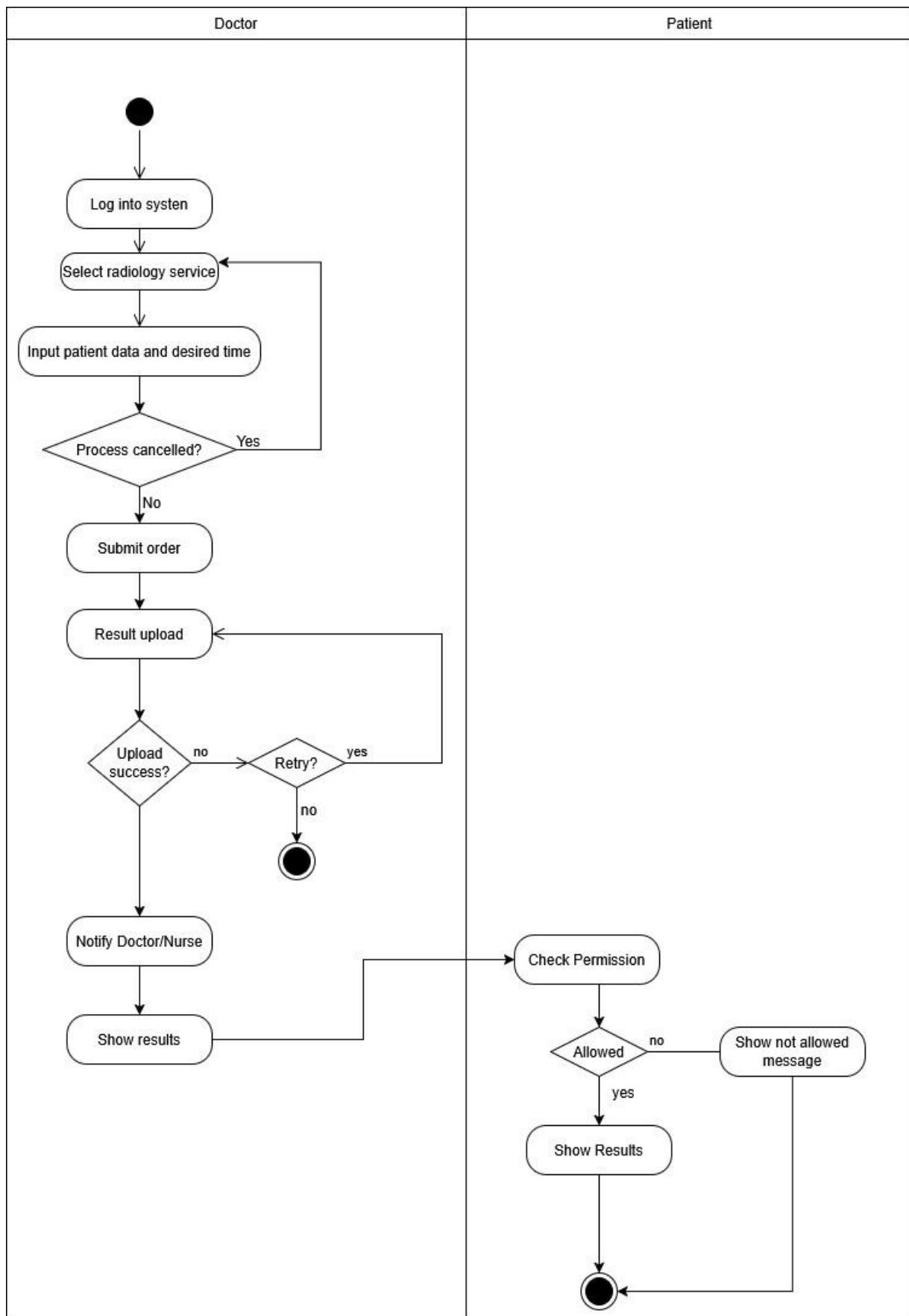
Author: Shpetim Shabanaj AD -09 Electronic Health Records Update



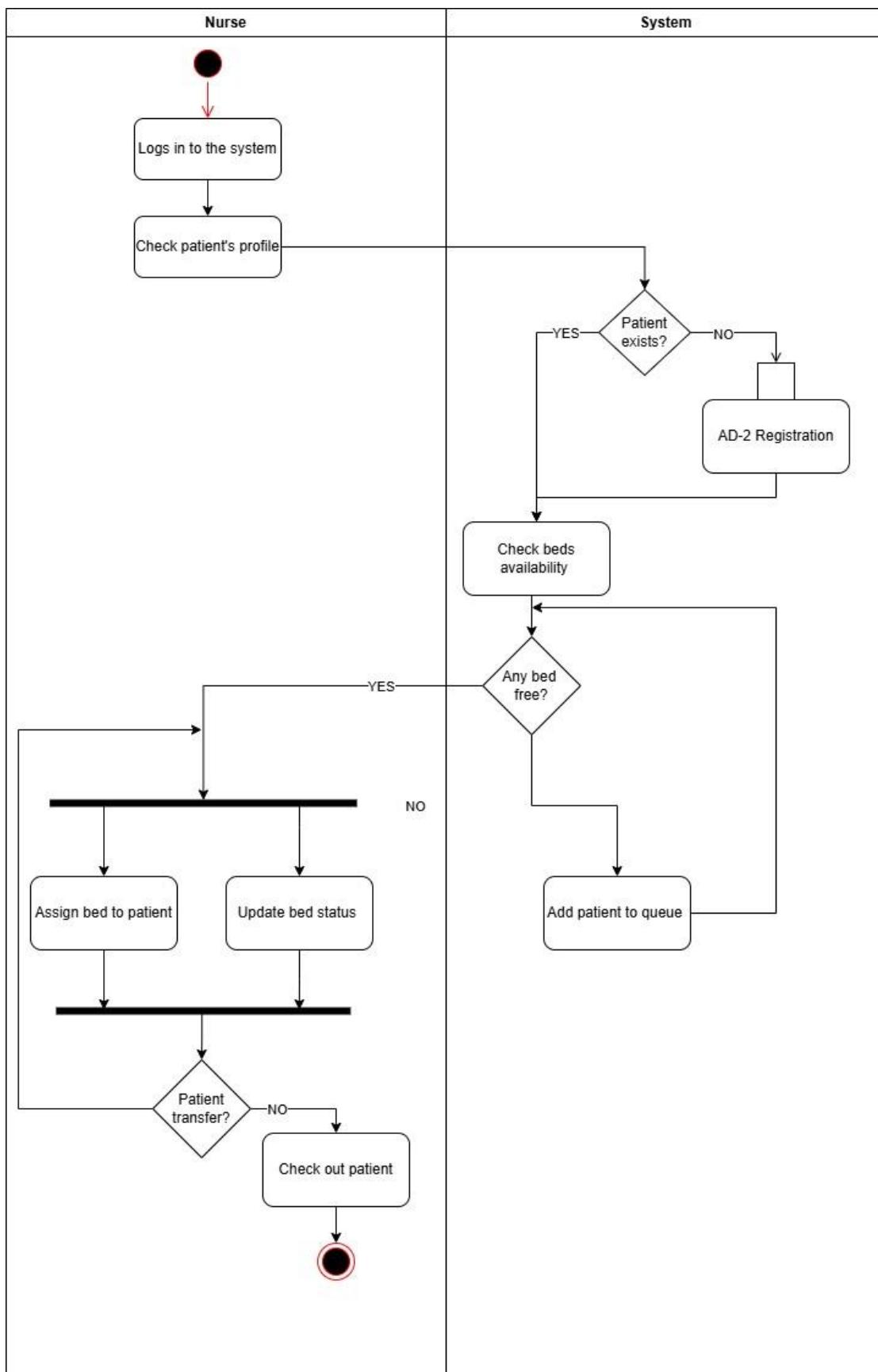
Author: Arjan Muka AD -10 Lab Test Ordering and Result Upload



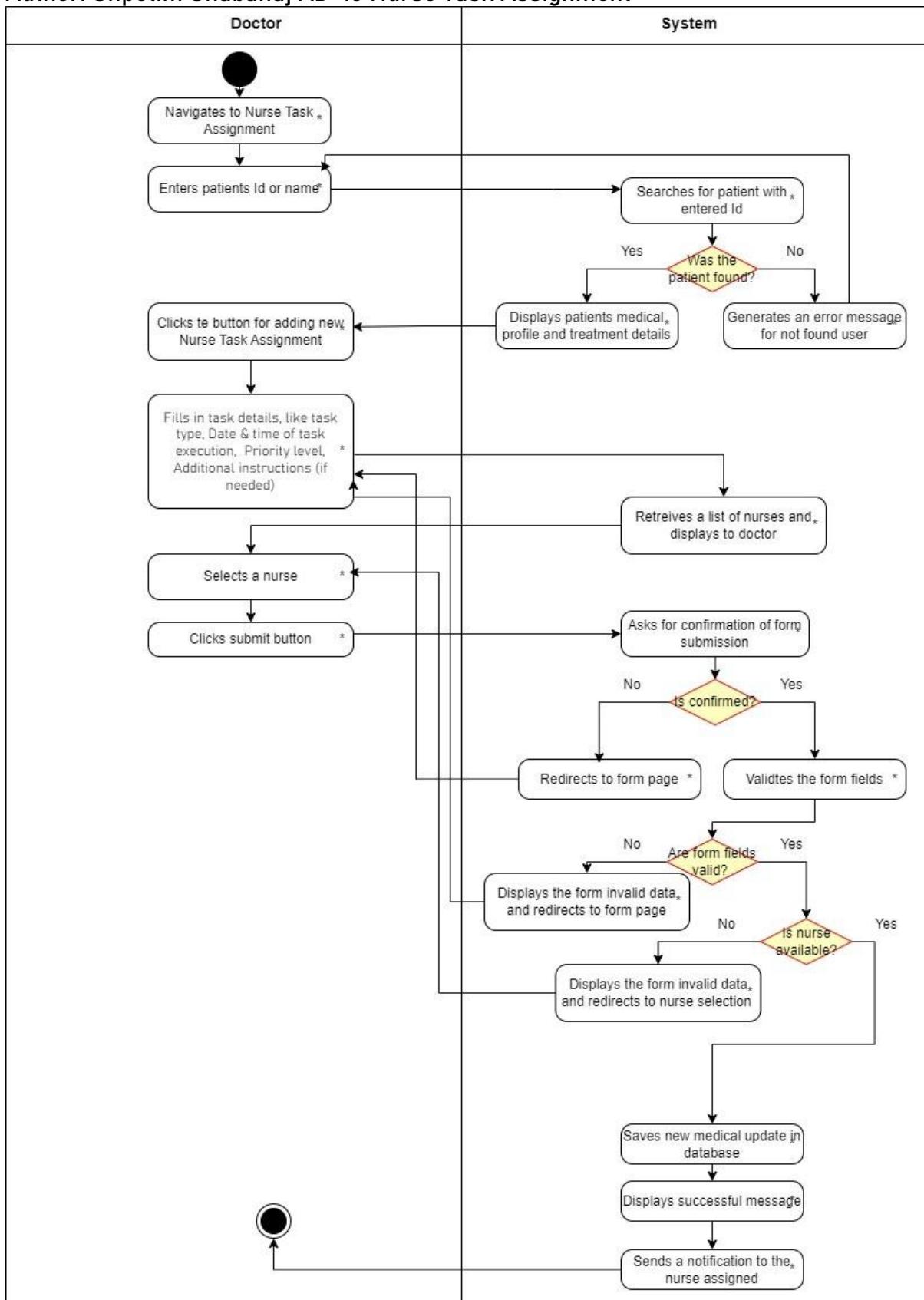
Author: Nikola Rigo AD -11 Radiology & Imaging Ordering and Result Upload



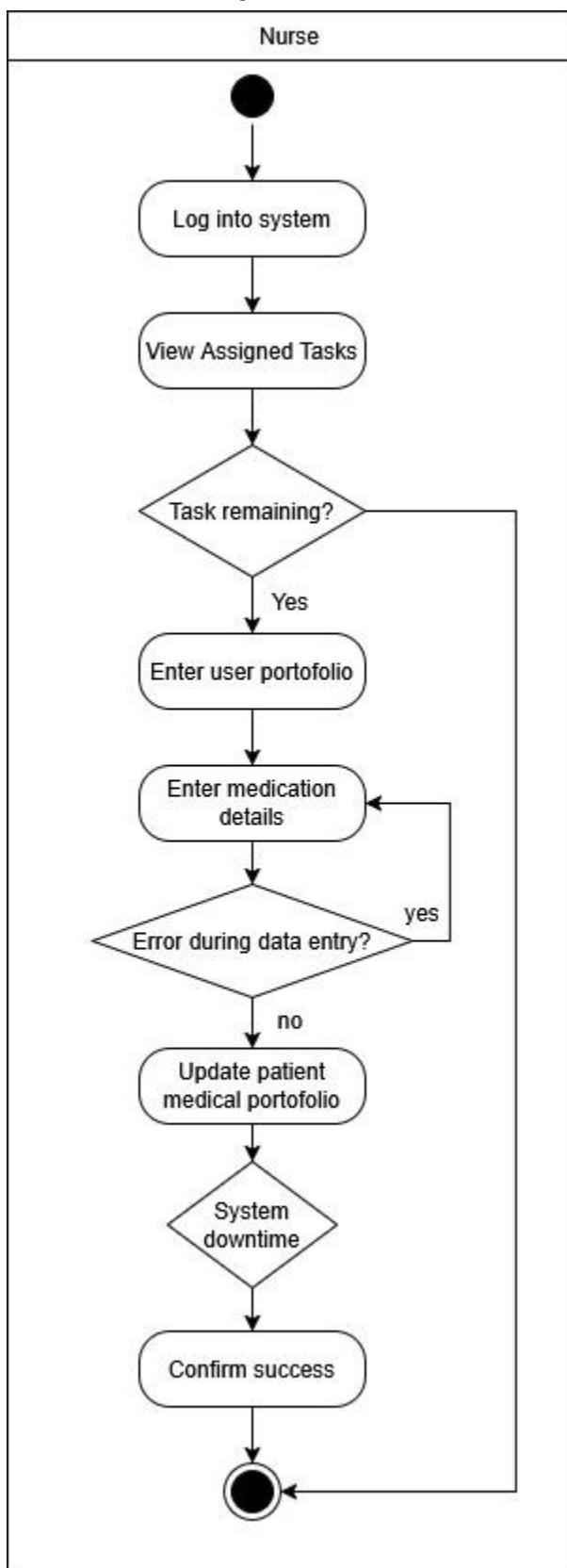
Author: Arjan Muka AD-12 Inpatient & Bed Management



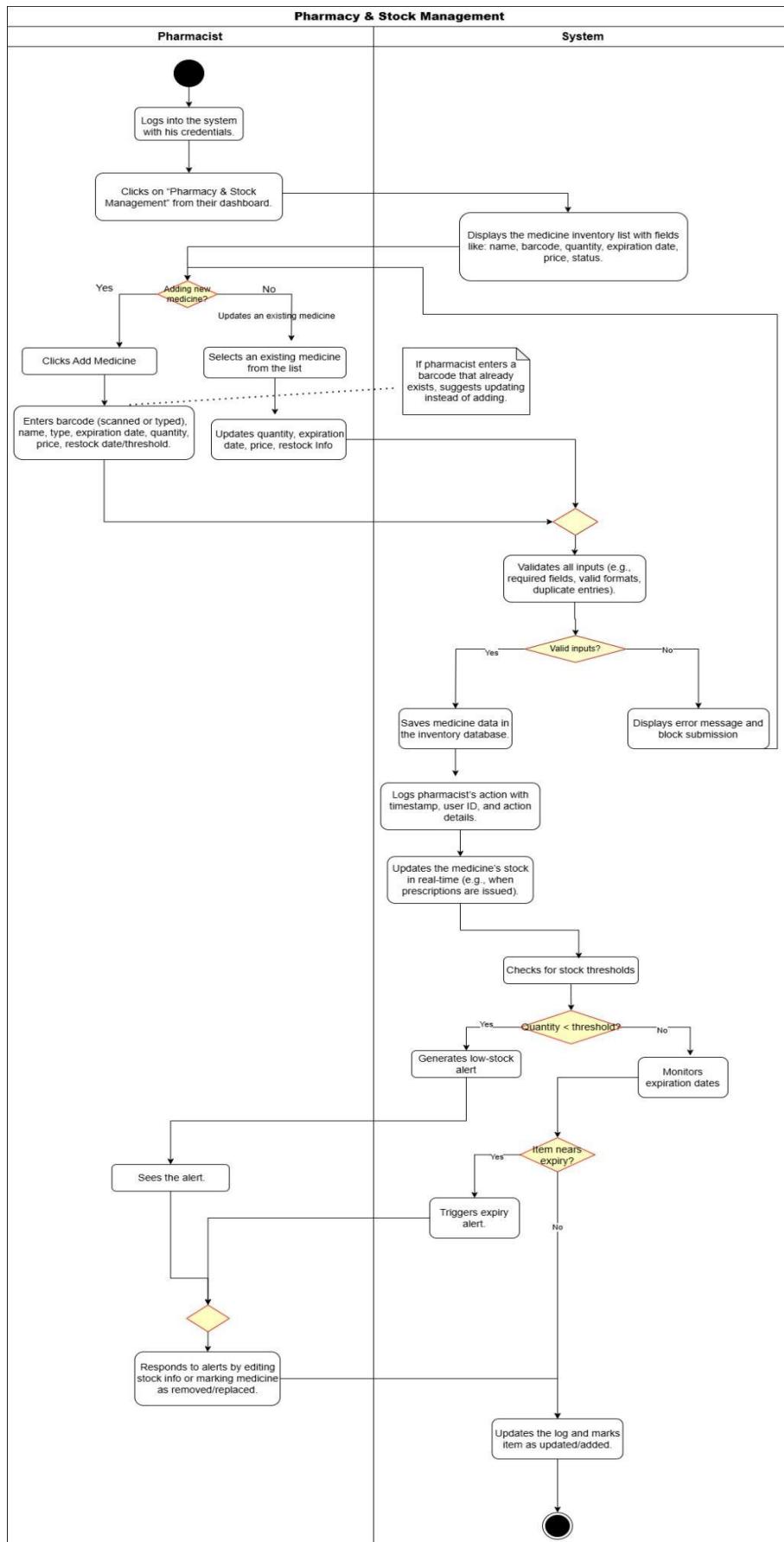
Author: Shpetim Shabanaj AD-13 Nurse Task Assignment



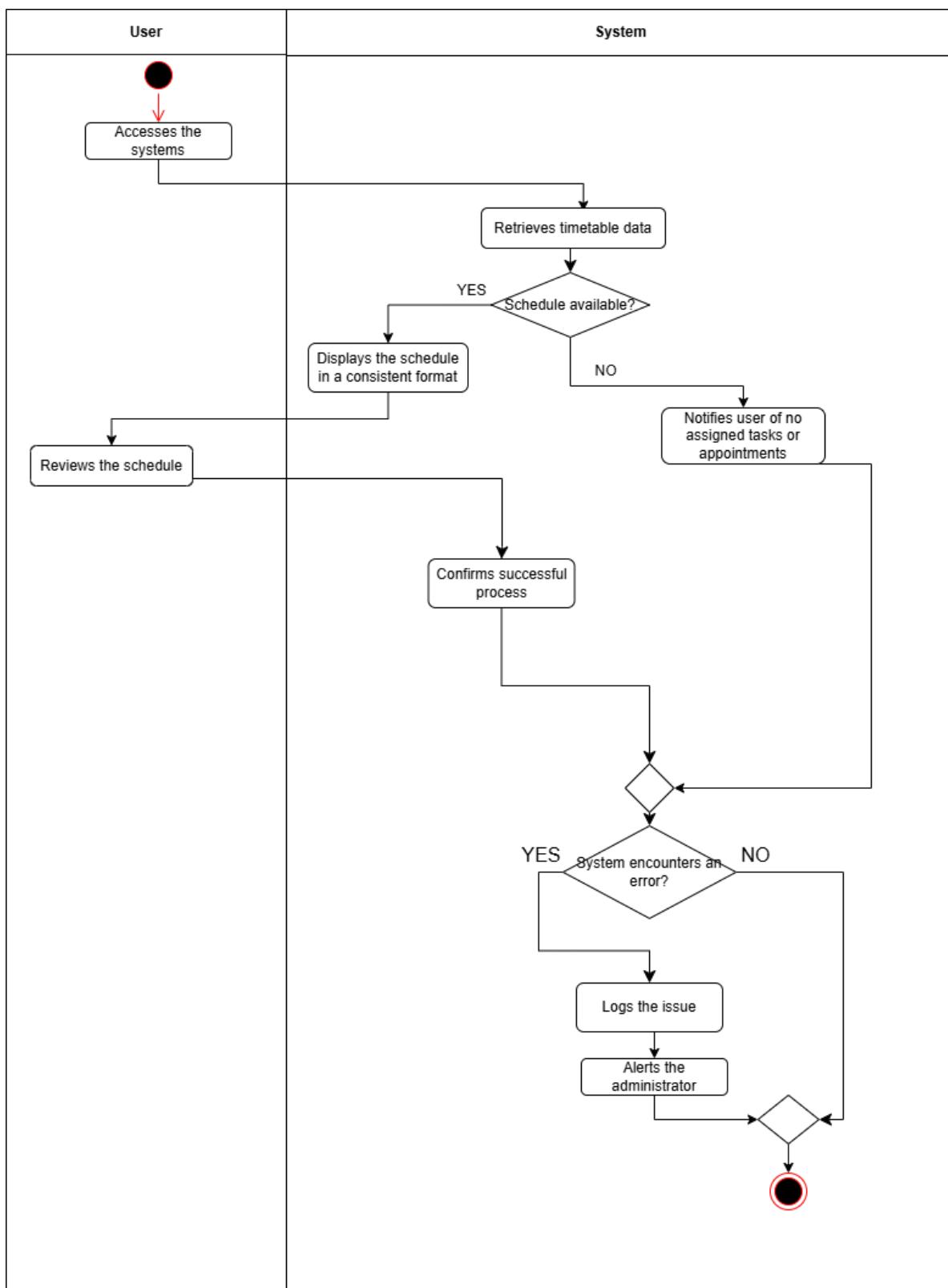
Author: Nikola Rigo AD-14 Medication Processing for patients



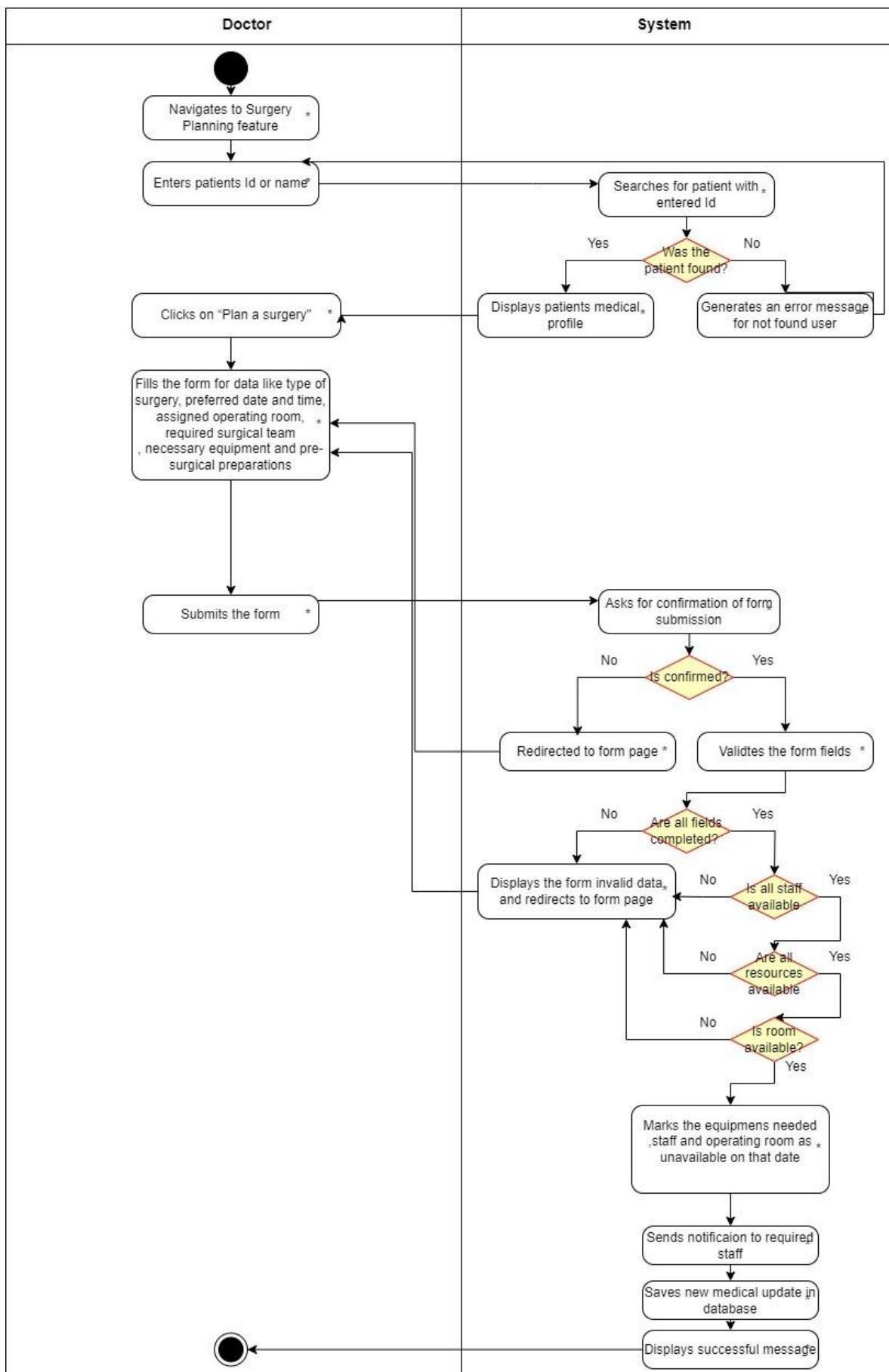
Author: Artjol Zaimi AD-15 Pharmacy & Stock Management



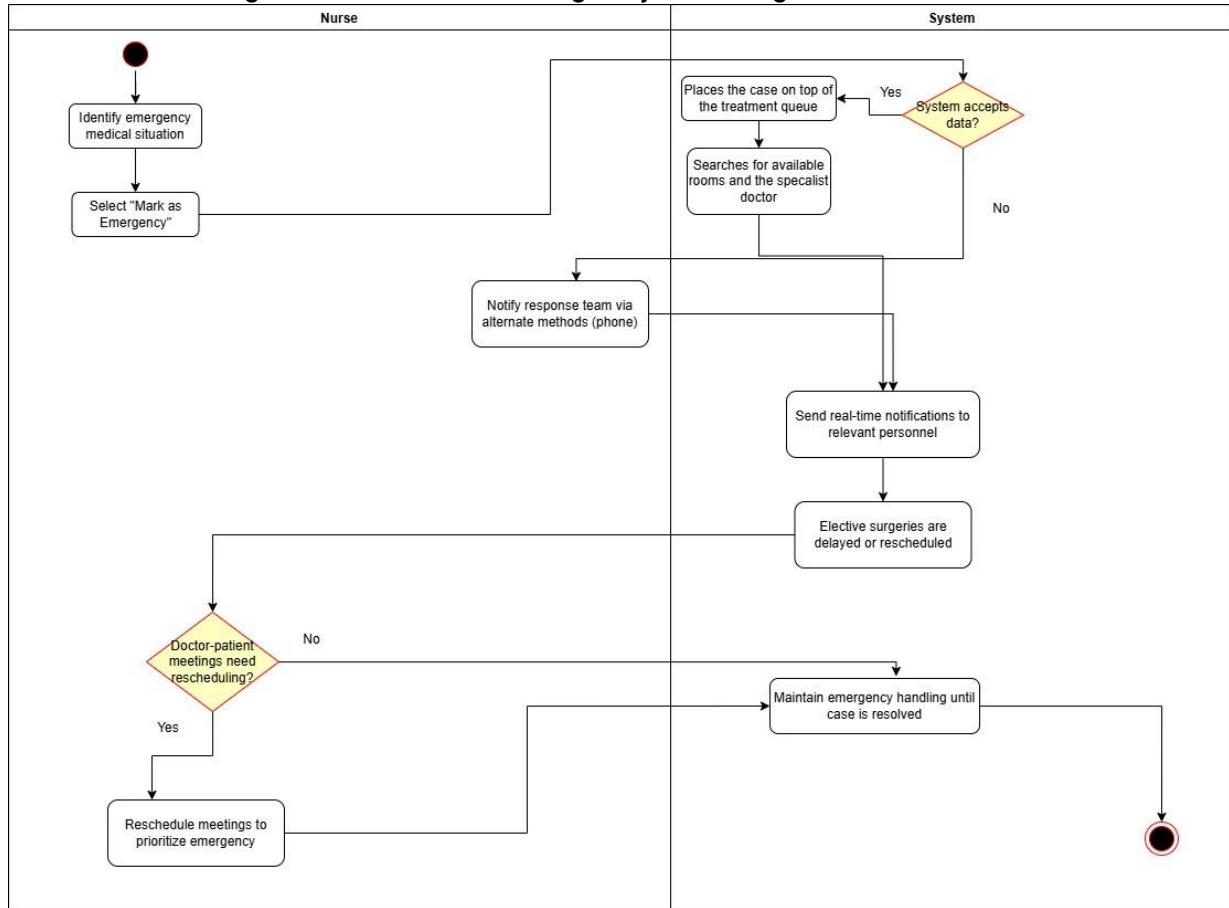
Author: Arlin Bashllari AD-16 Medical Staff Timetable



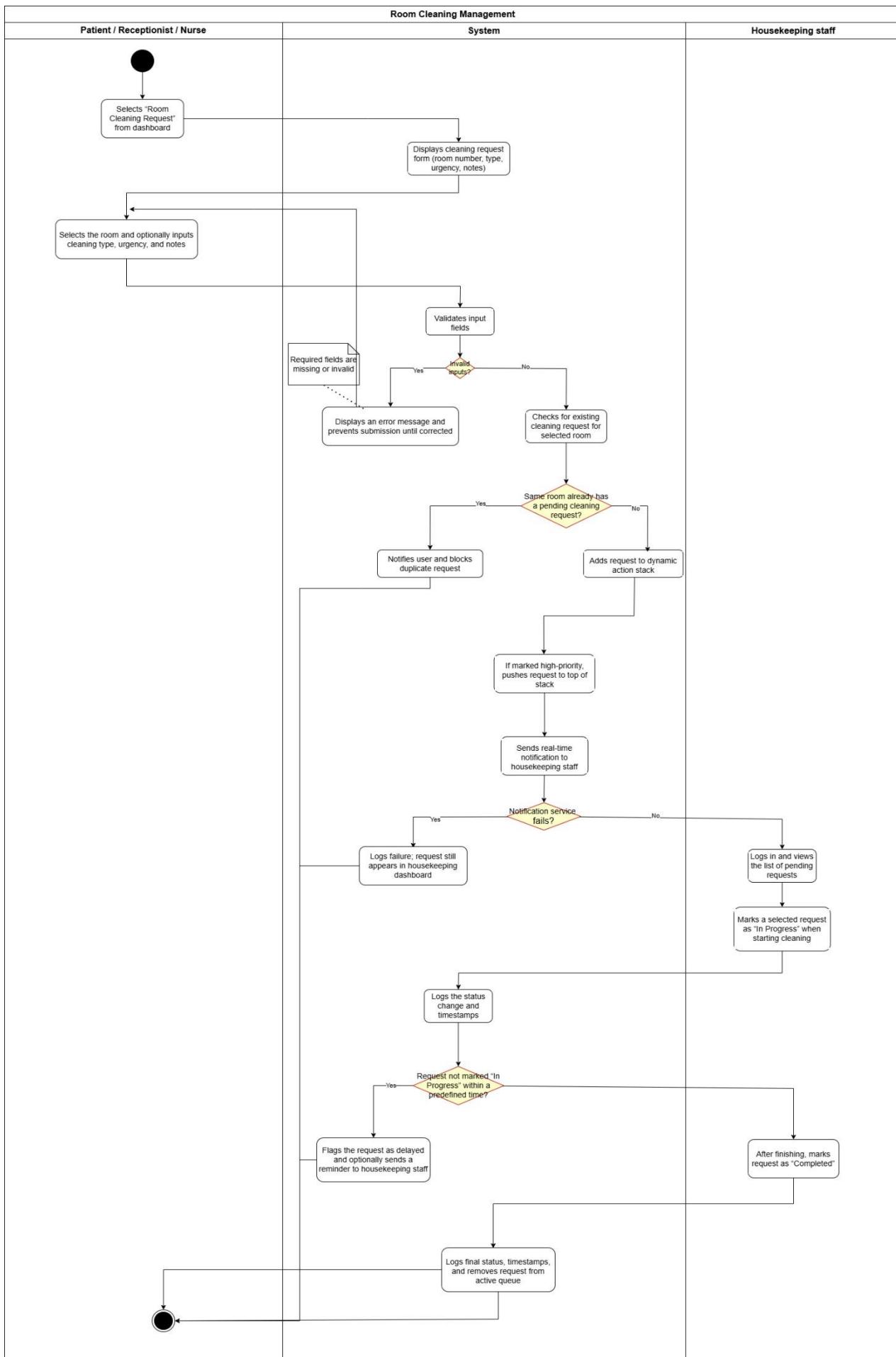
Author: Shpetim Shabanaj AD-17 Surgery Planning



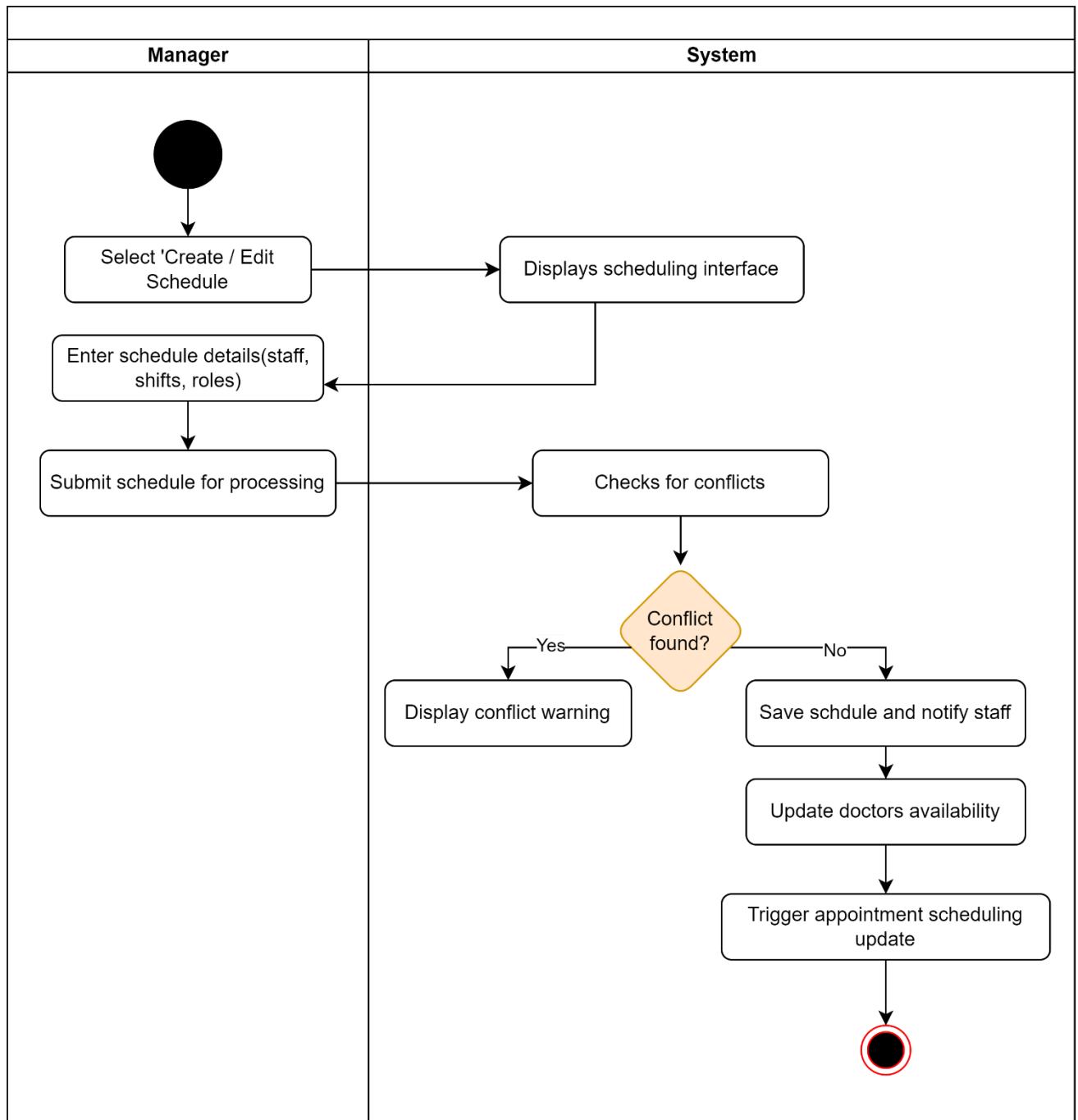
Author: Eglis Braho AD-18 Emergency Handling and Alerts



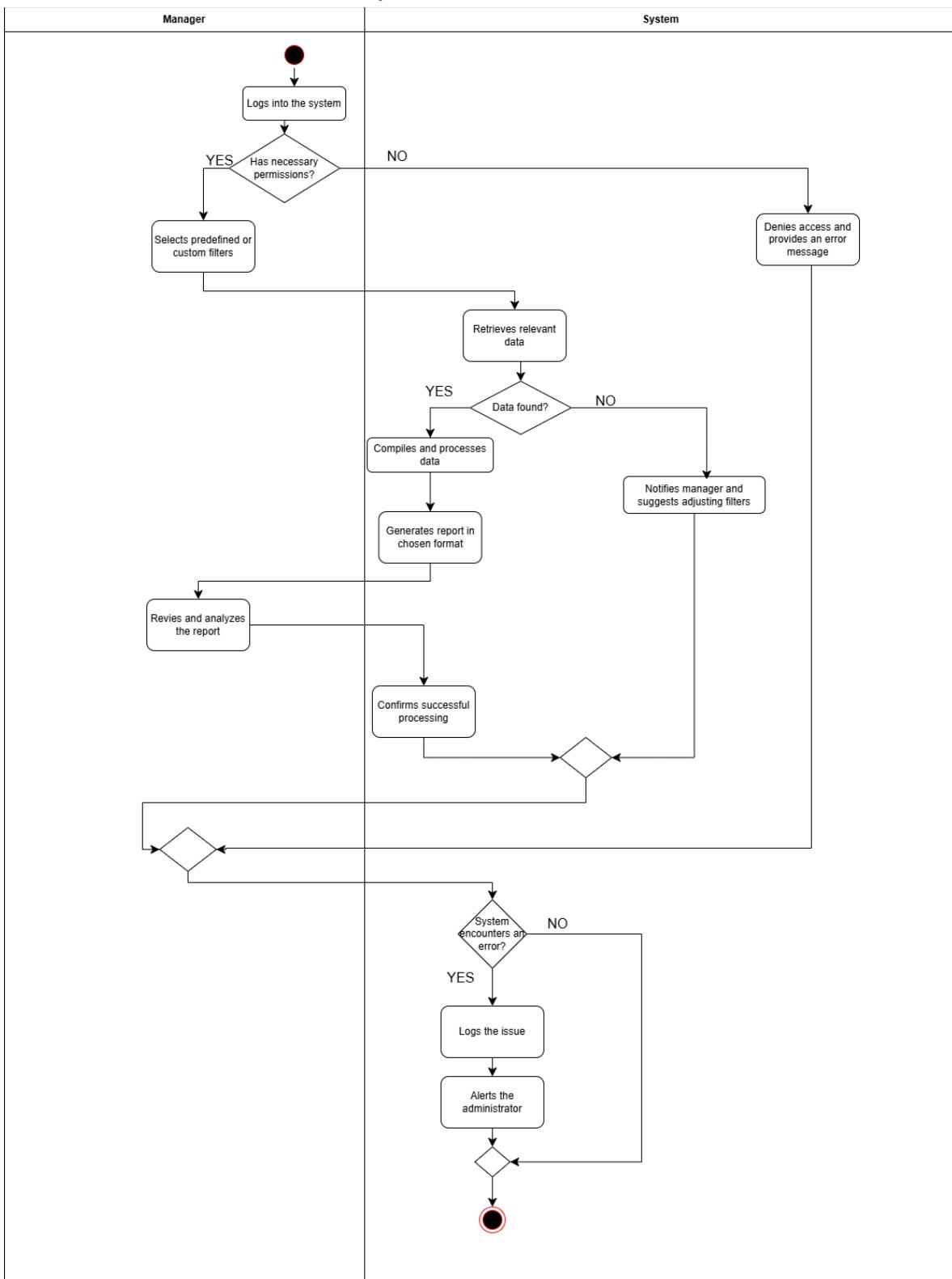
Author: Artjol Zaimi AD-19 Room Cleaning Management



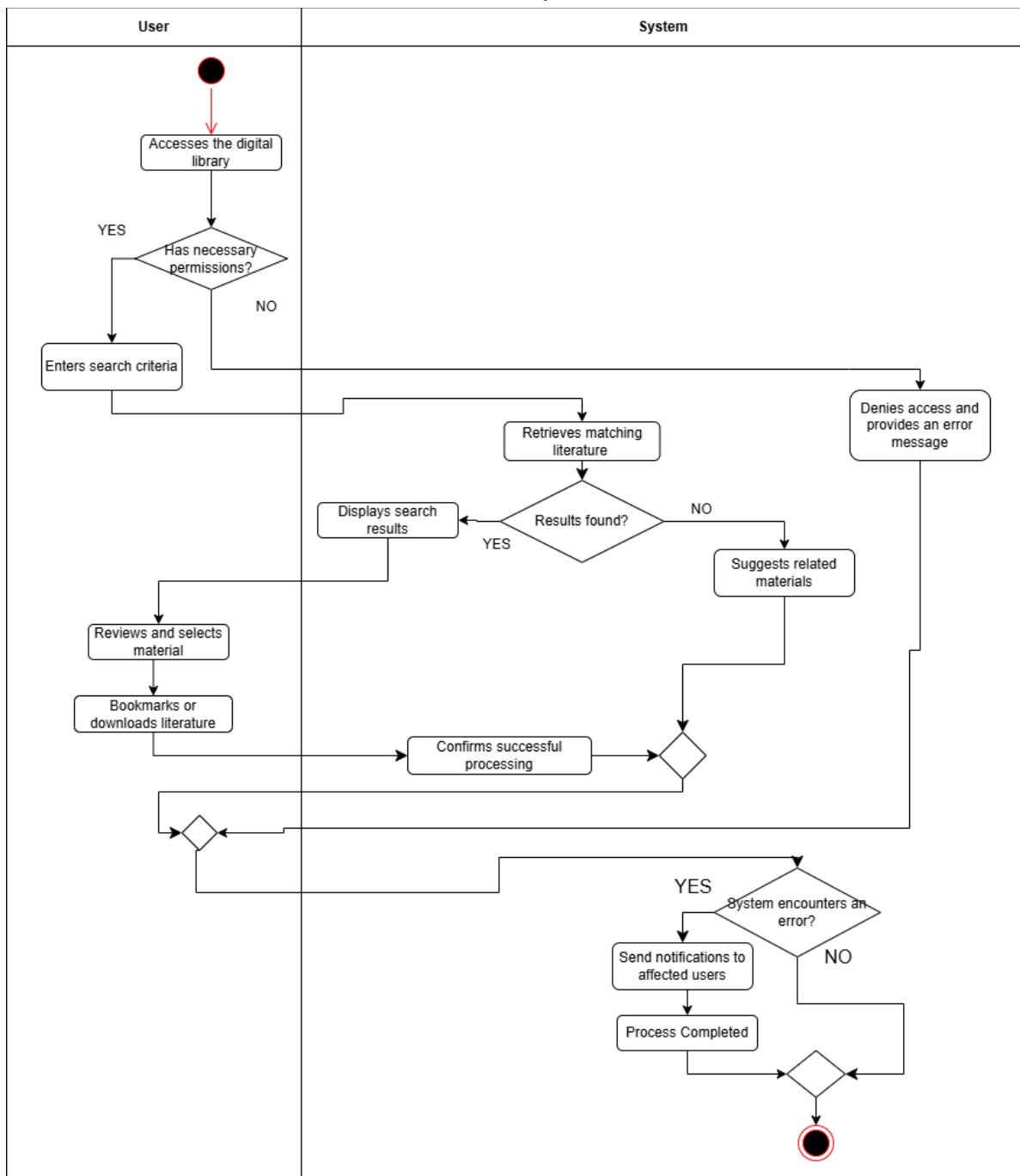
Author: Marin Tartaraj AD-20 Staff Scheduling



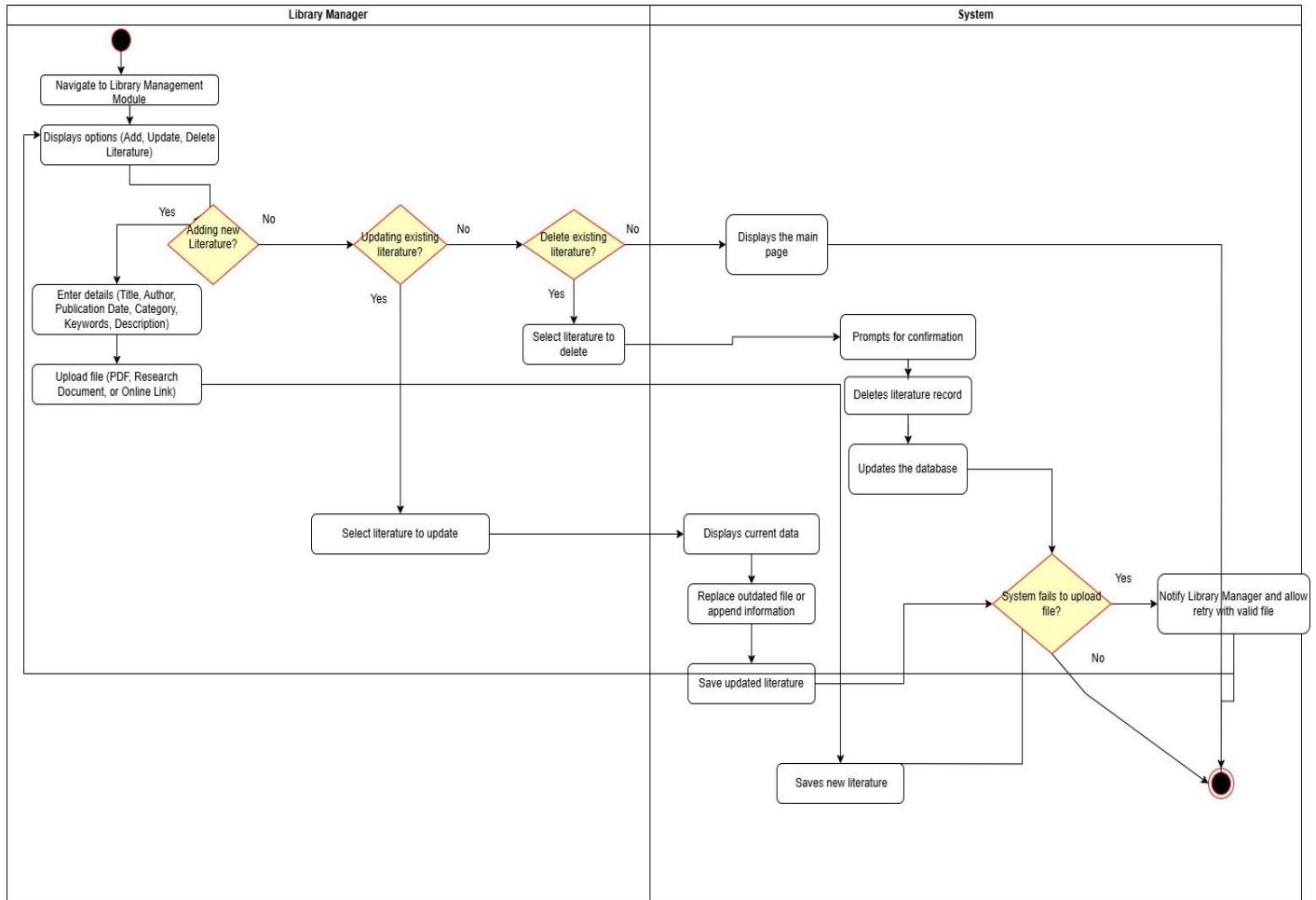
Author: Arlin Bashllari AD-21 Report Generation



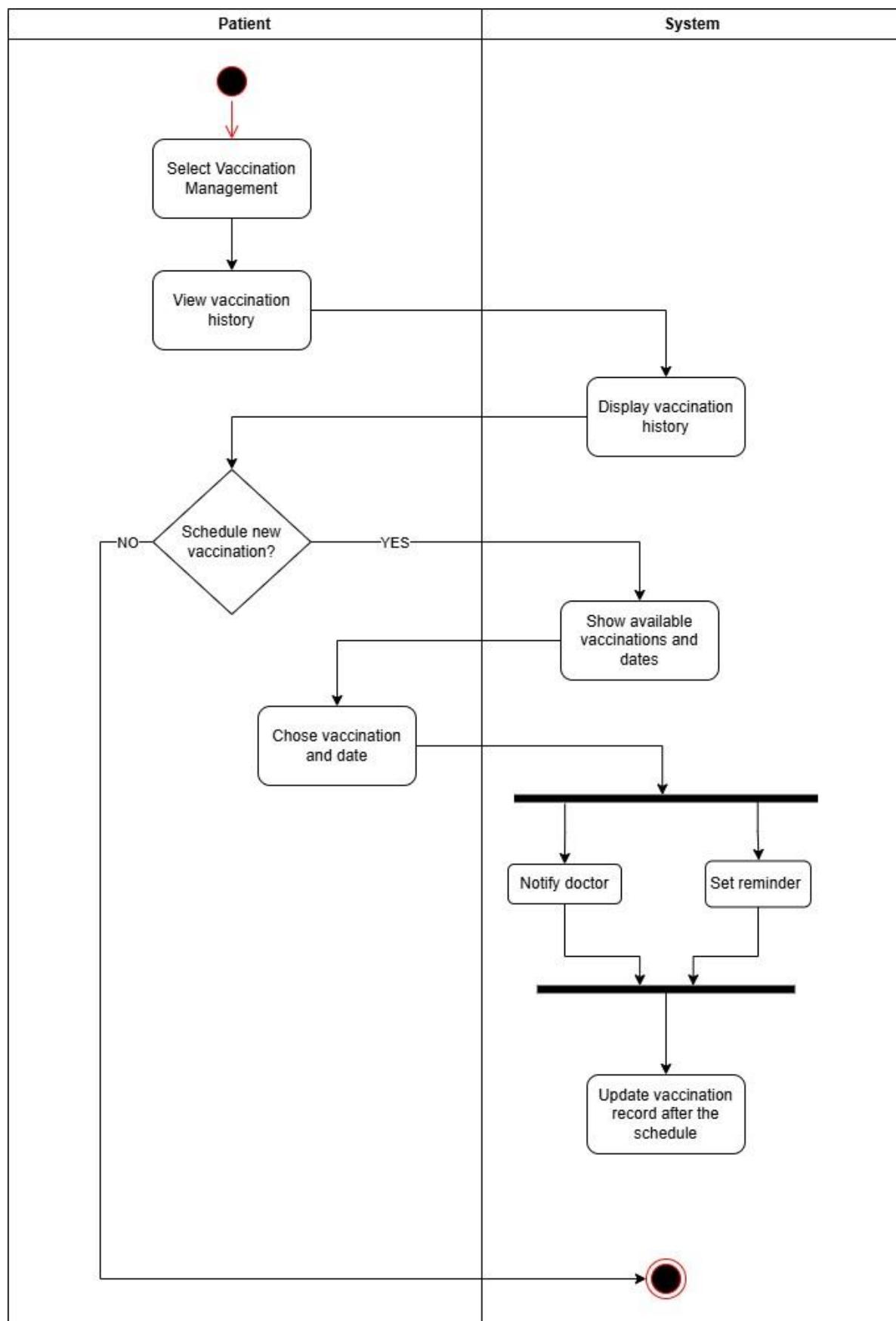
Author: Arlin Bashllari AD-22 Library and Literature Search



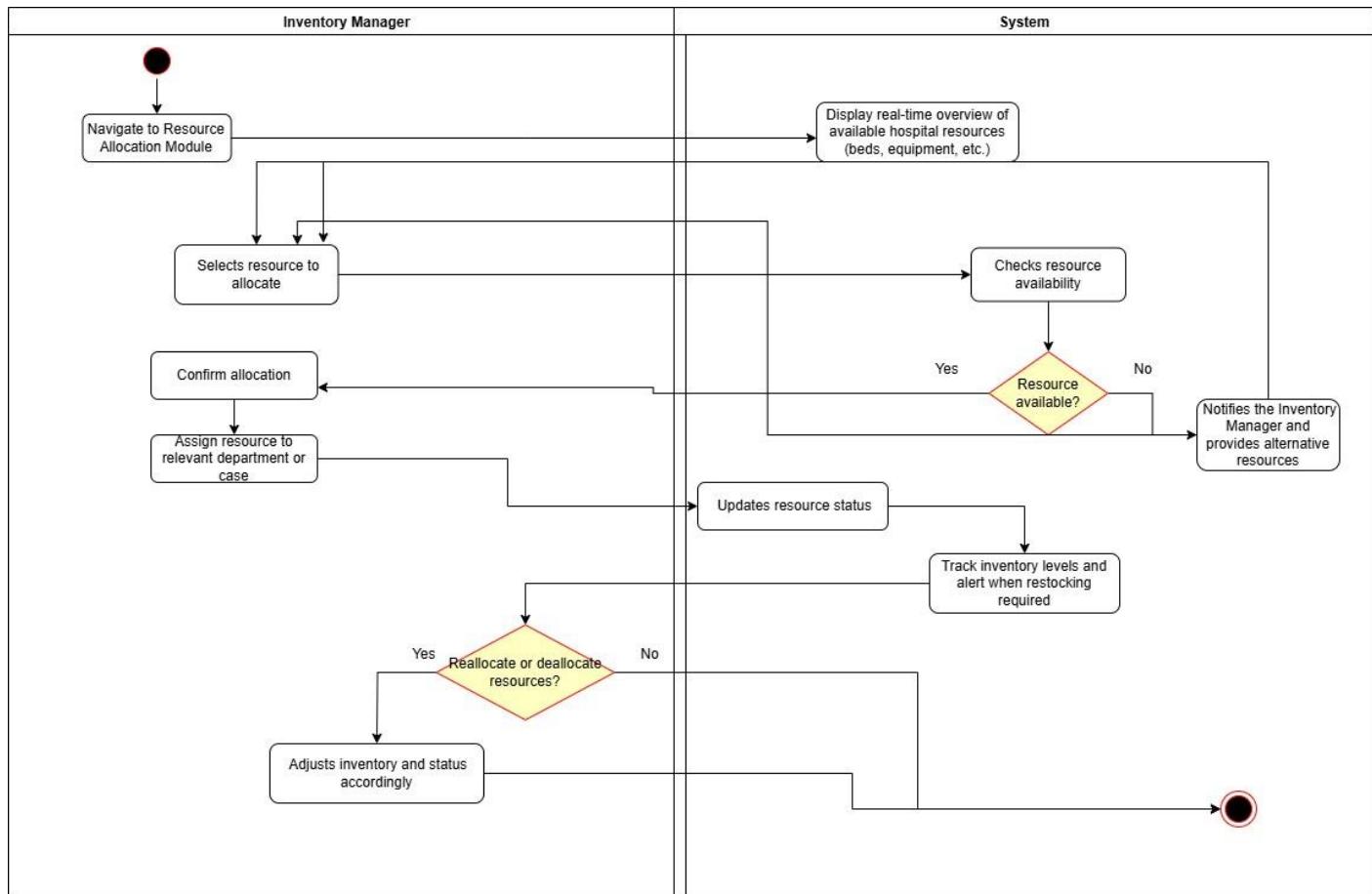
Author: Eglis Braho AD-23 Library and Literature Management



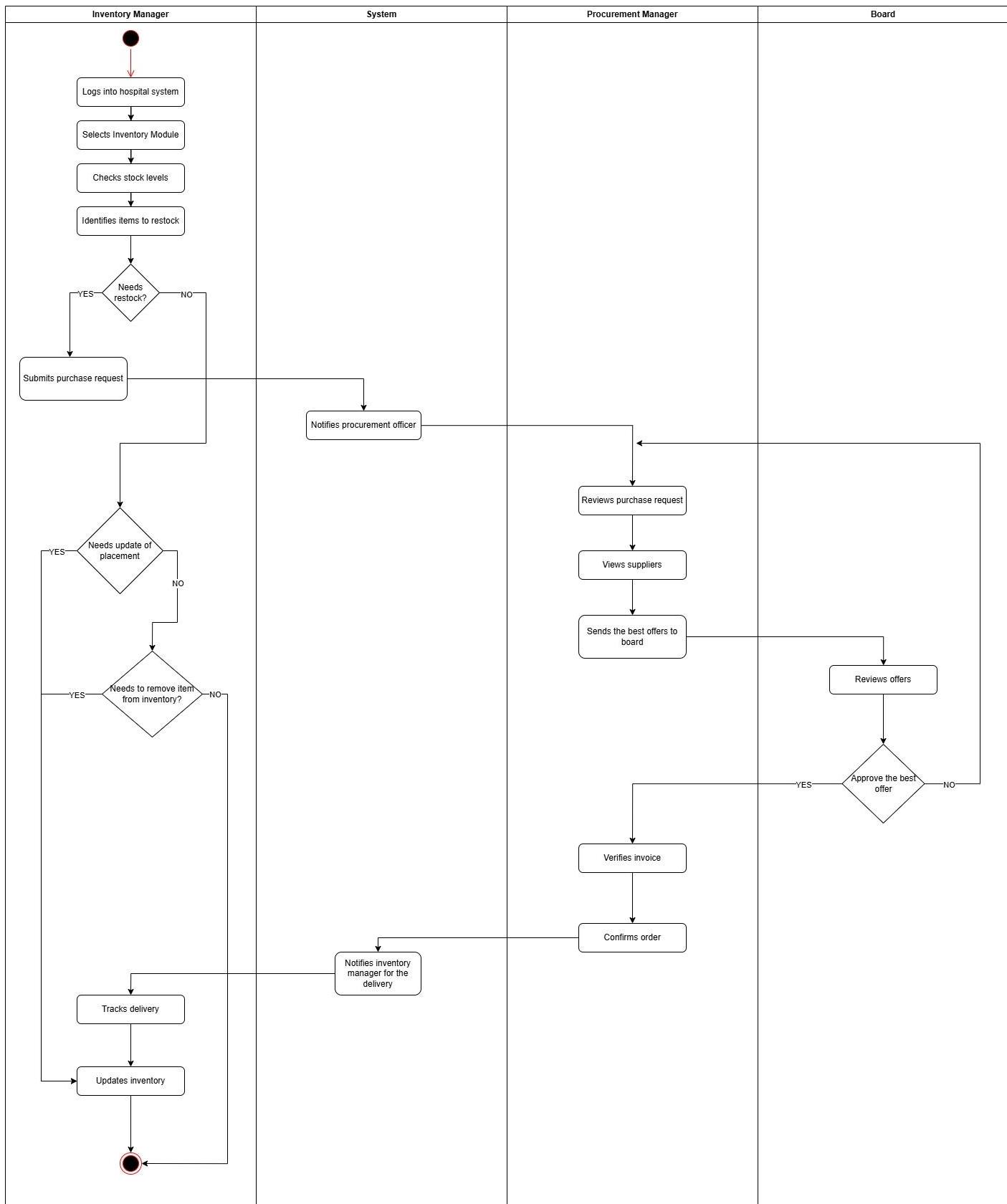
Author: Arjan Muka AD-24 Vaccination Management



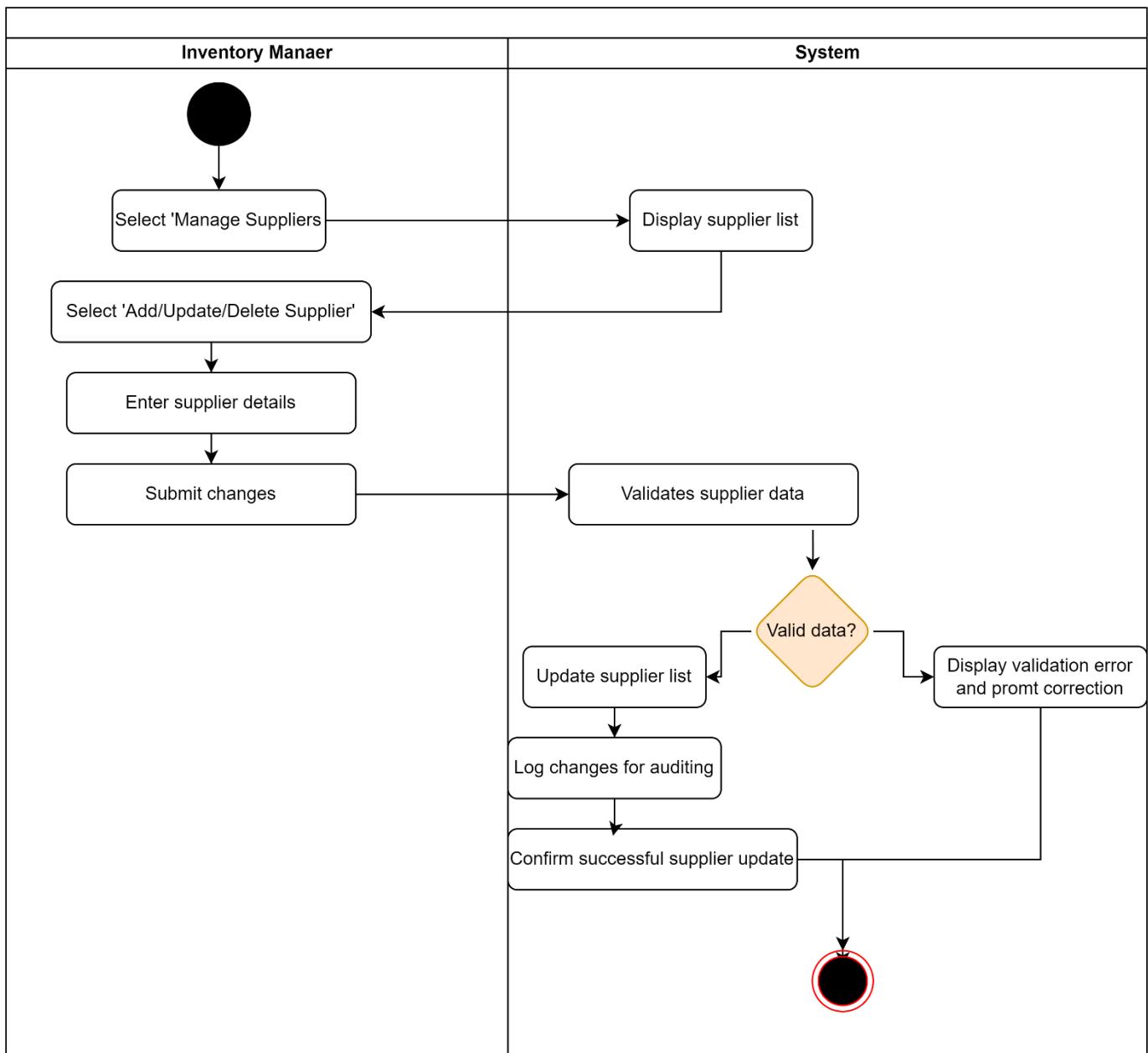
Author: Eglis Braho AD-25
Resource Allocation



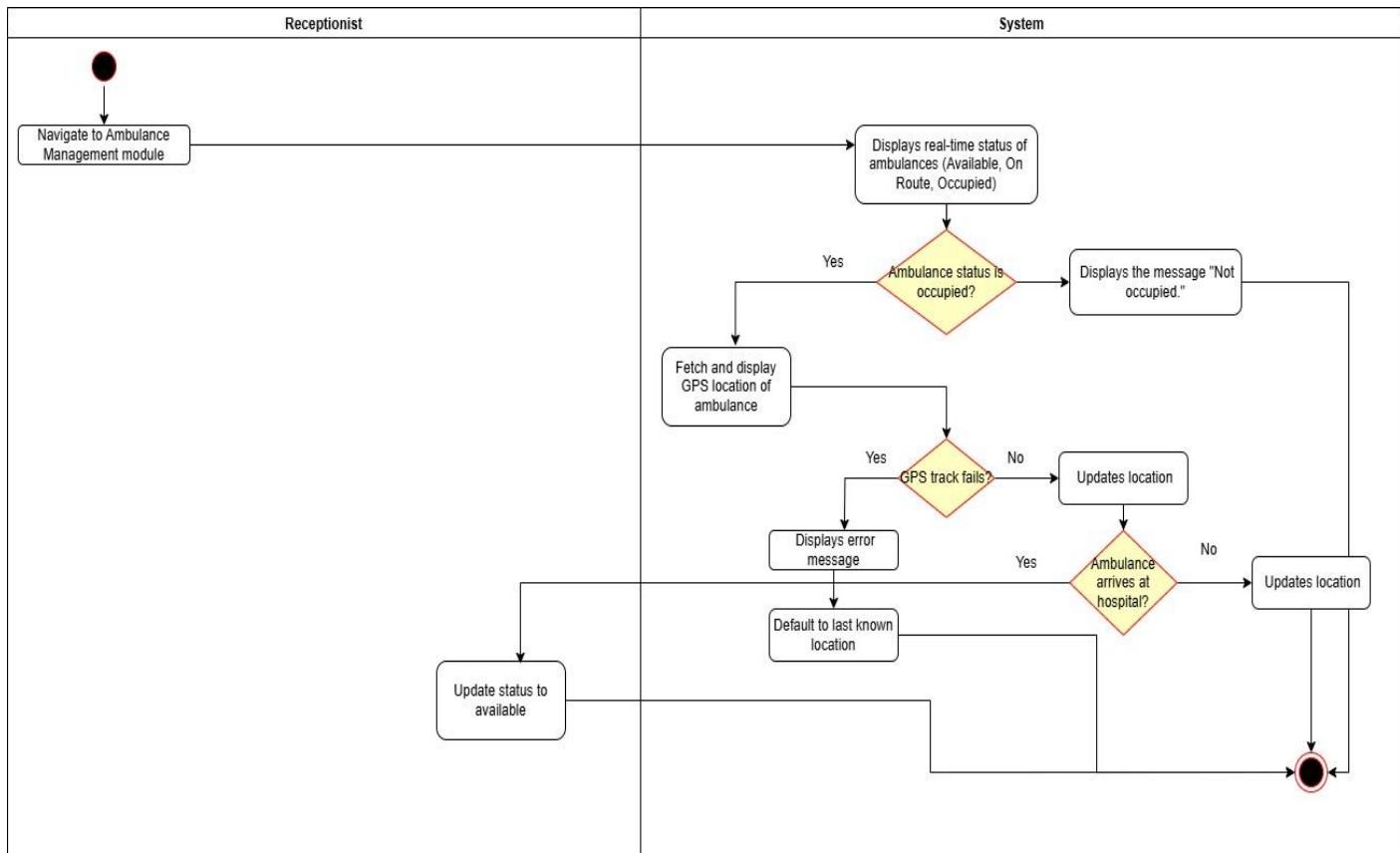
Author: Nikola Rigo/ Arjan Muka AD-26 Inventory Item & Procurement



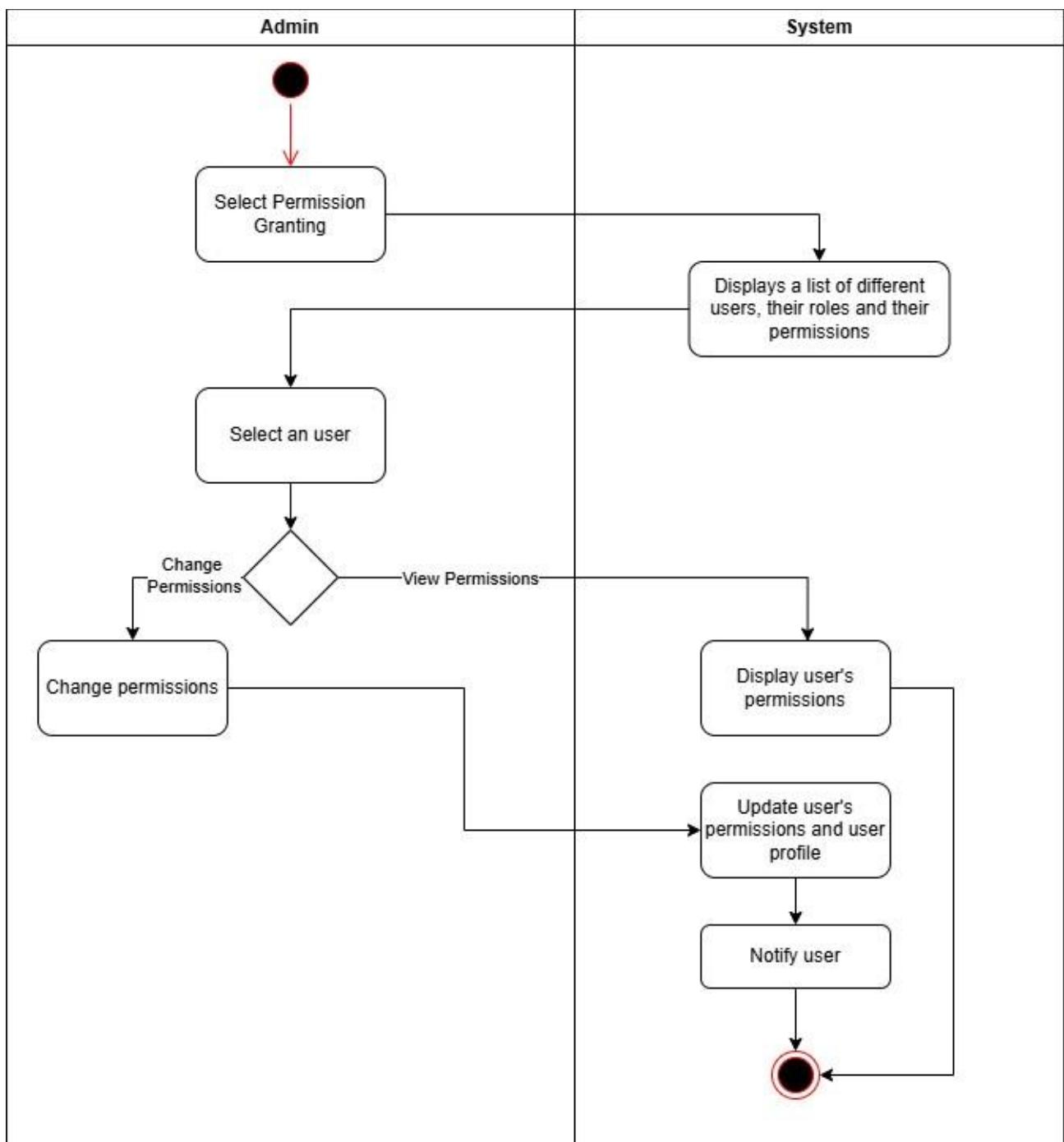
Author: Marin Tartaraj AD-27
Supplier Management



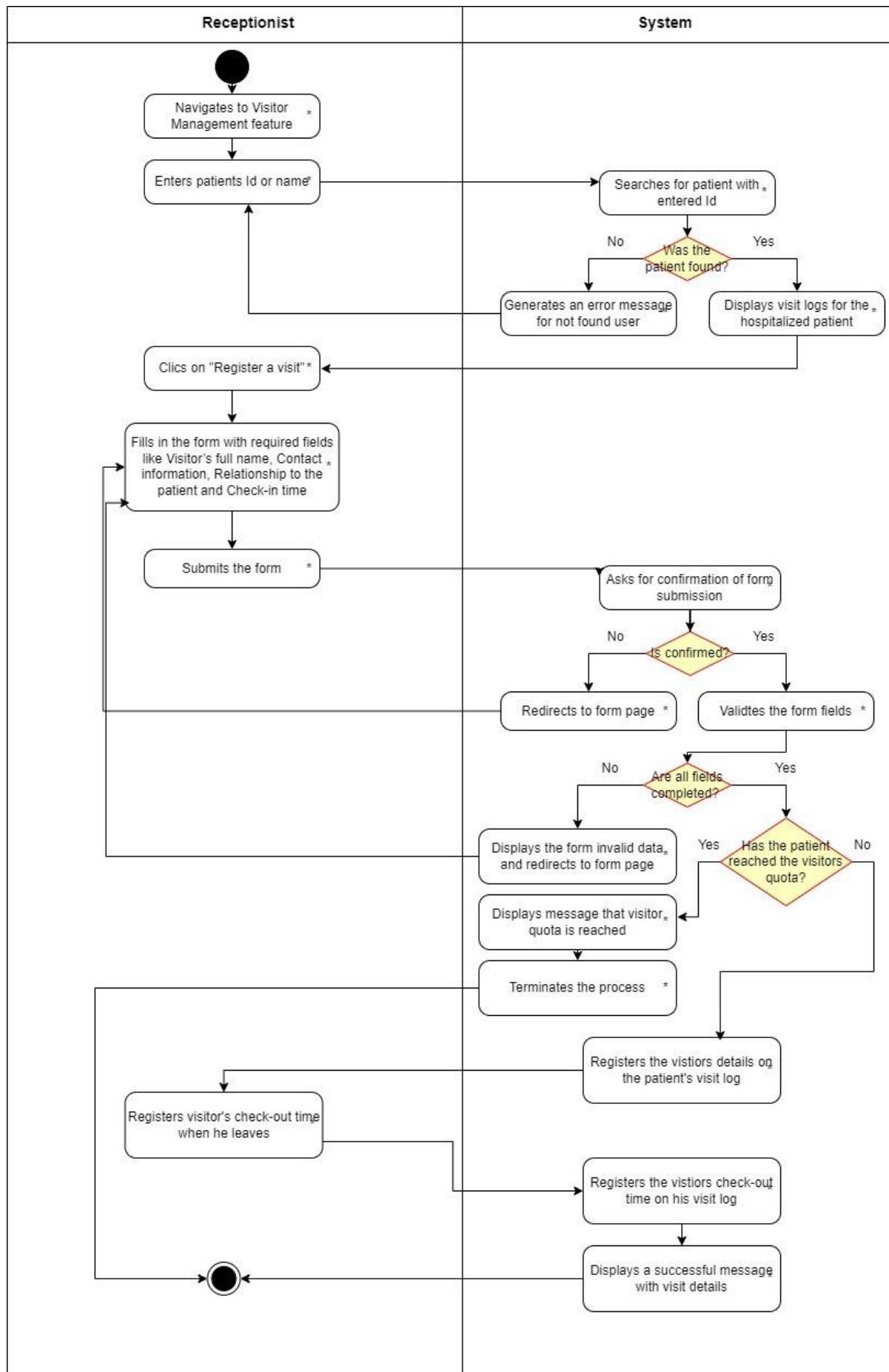
Author: Eglis Braho AD-28
Ambulance Management



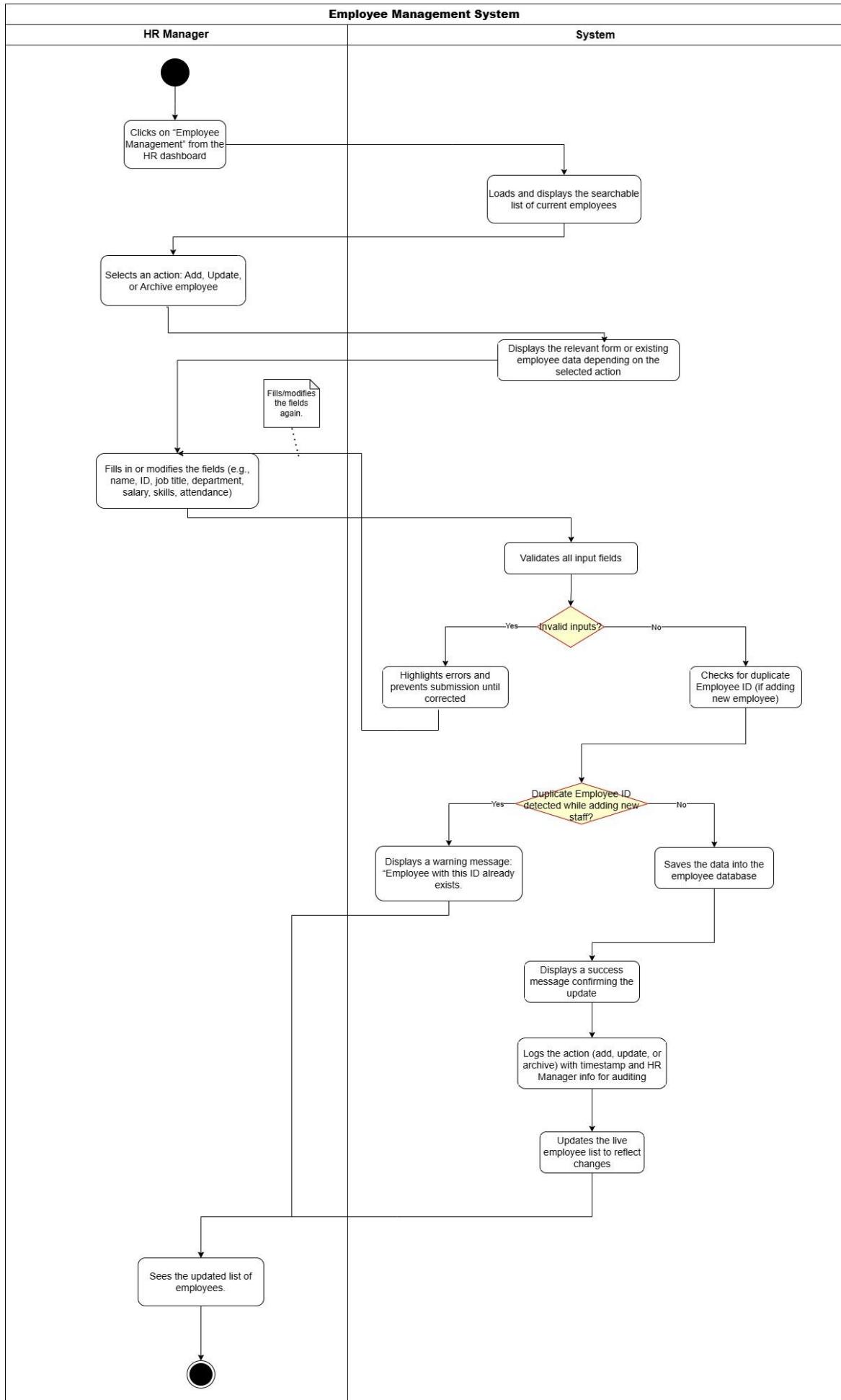
Author: Arjan Muka AD-29 Permission Granting



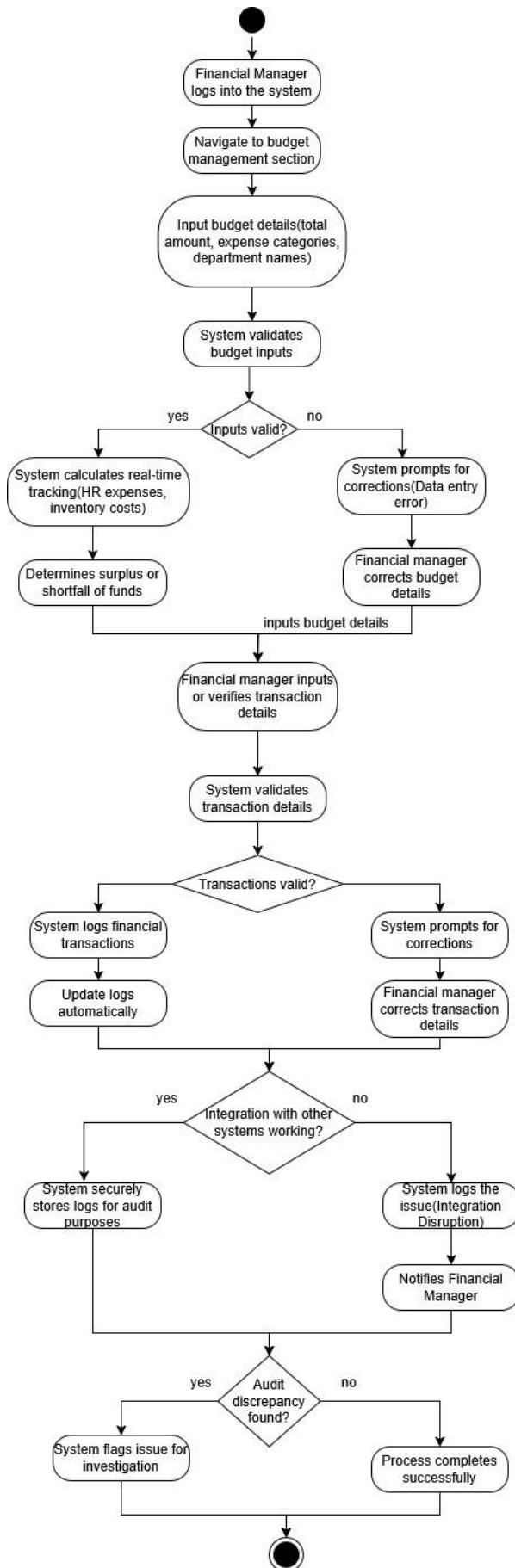
Author: Shpetim Shabanaj AD-30 Visitor Management



Author: Artjol Zaimi AD-31 Employee Management System

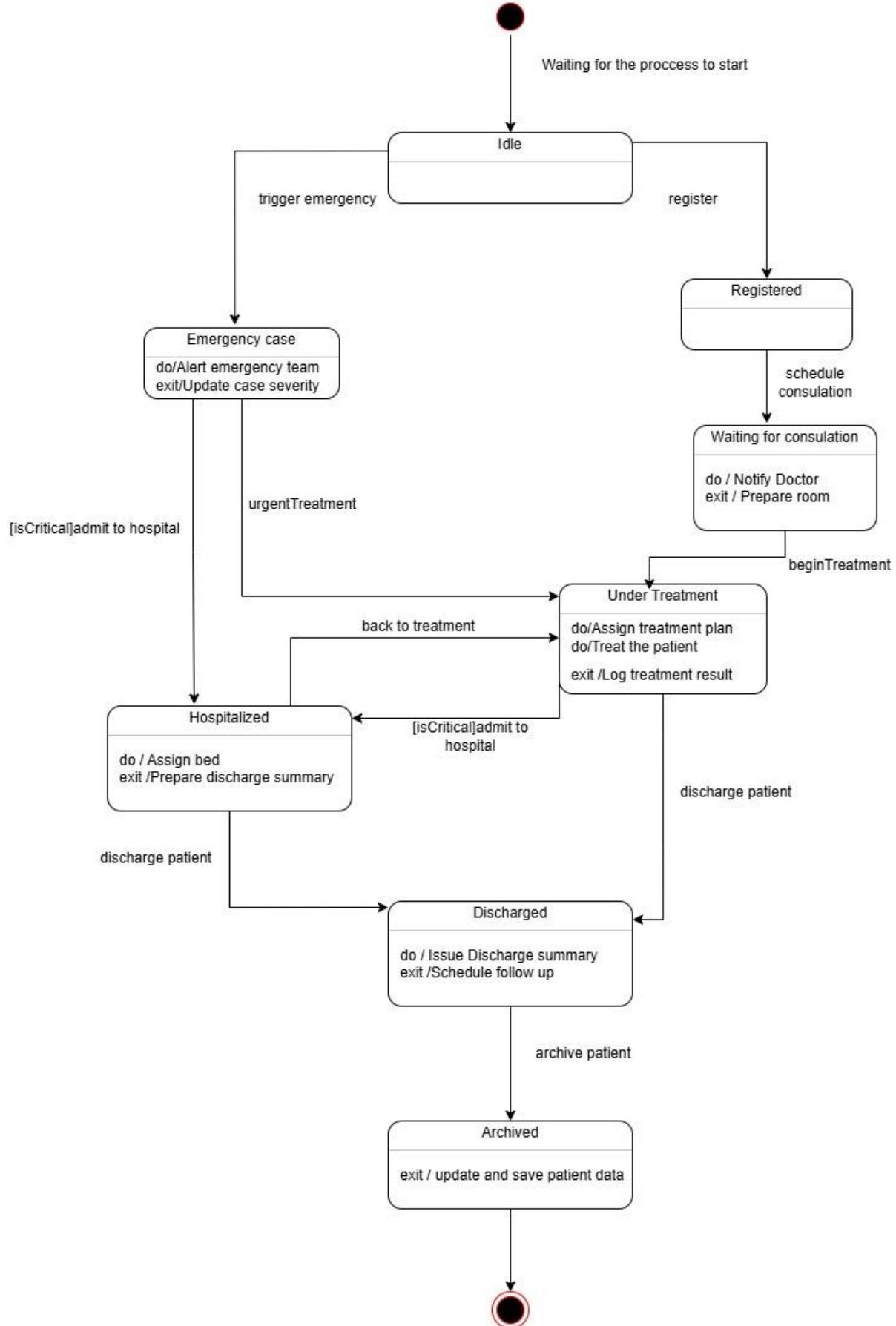


Author: Nikola Rigo AD-32 Financial Management

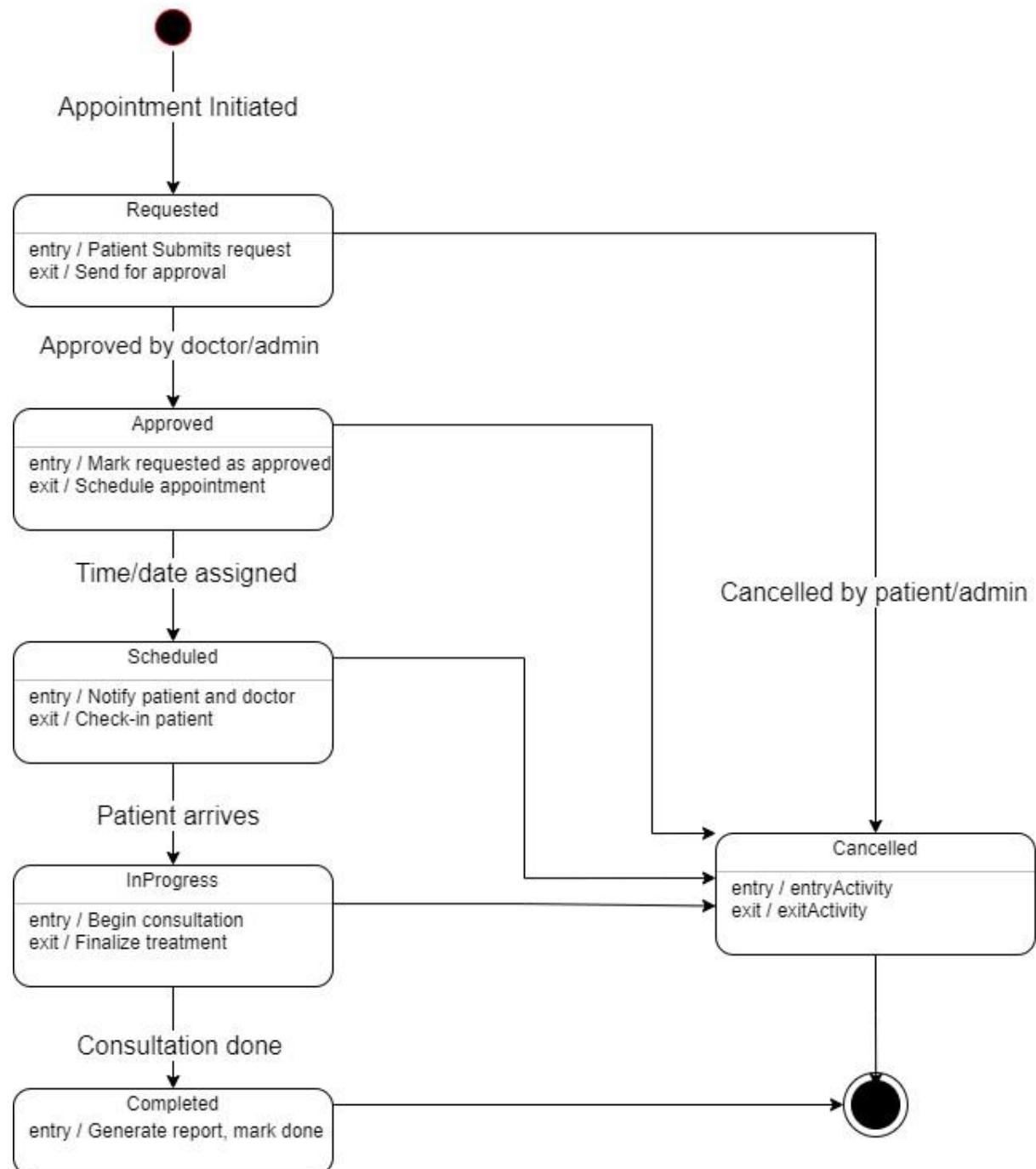


State Diagrams

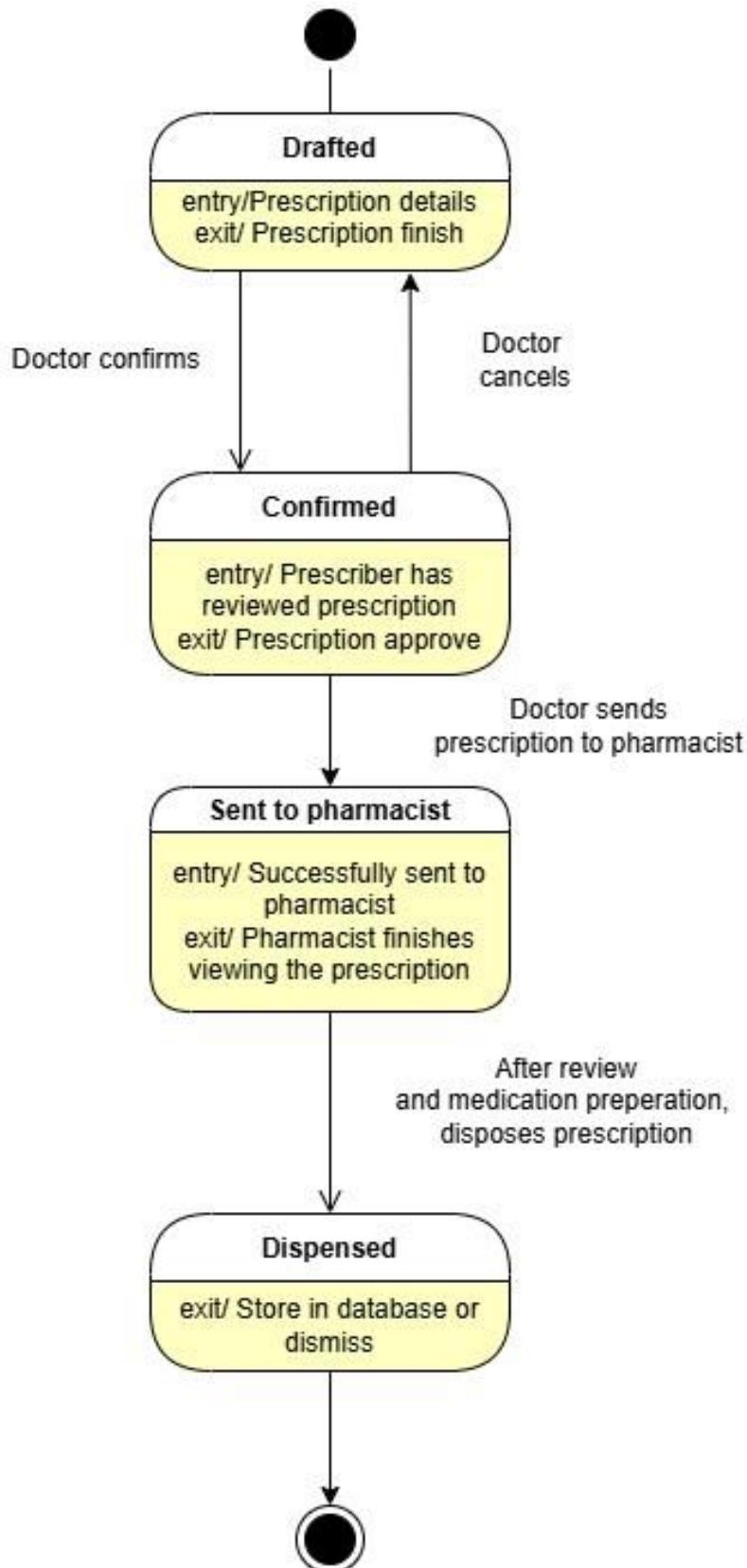
State Diagram 1: Patient State Diagram --- Eglis Braho



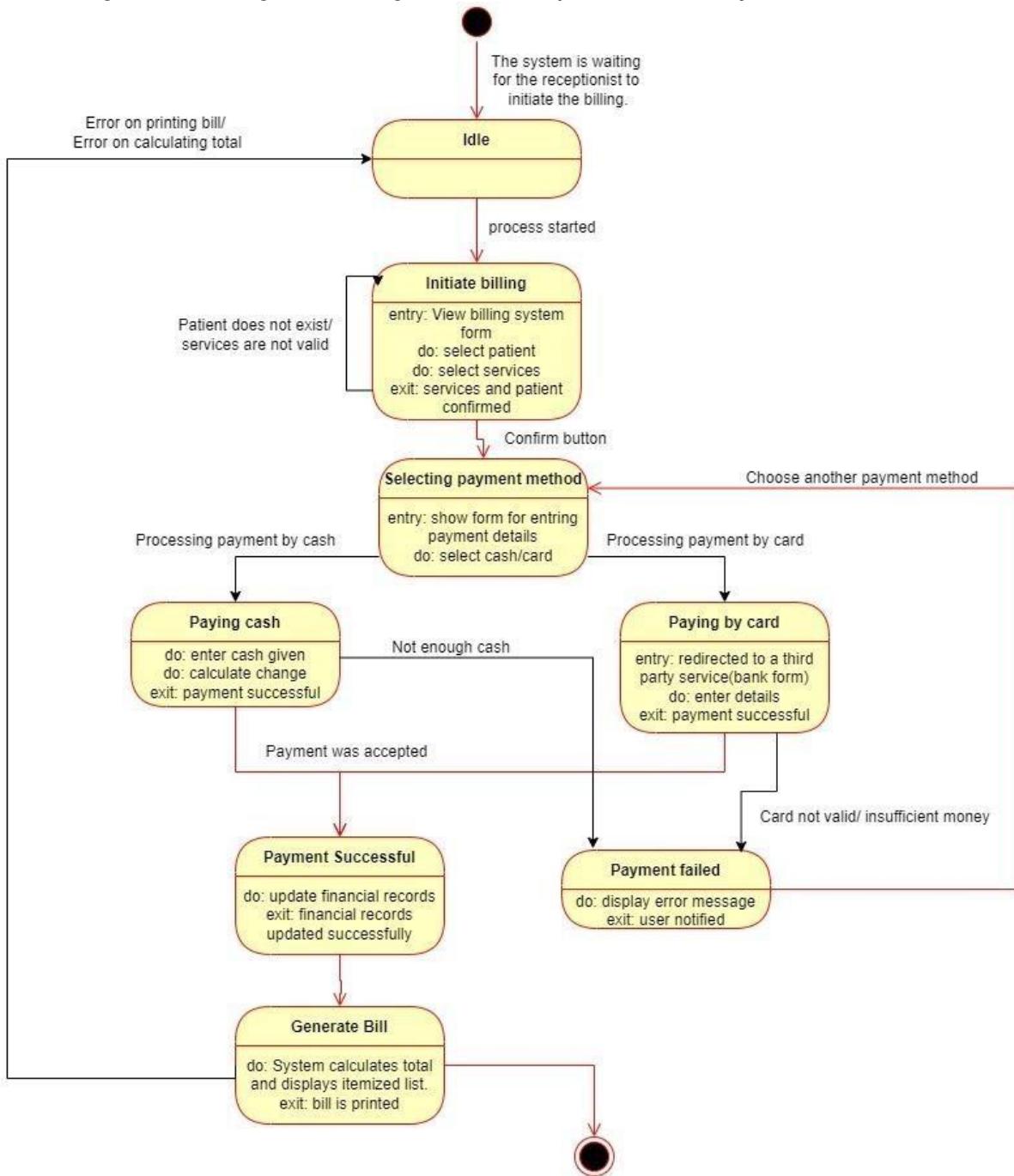
State Diagram 2: Appointment State Diagram --- Marin Tartaraj



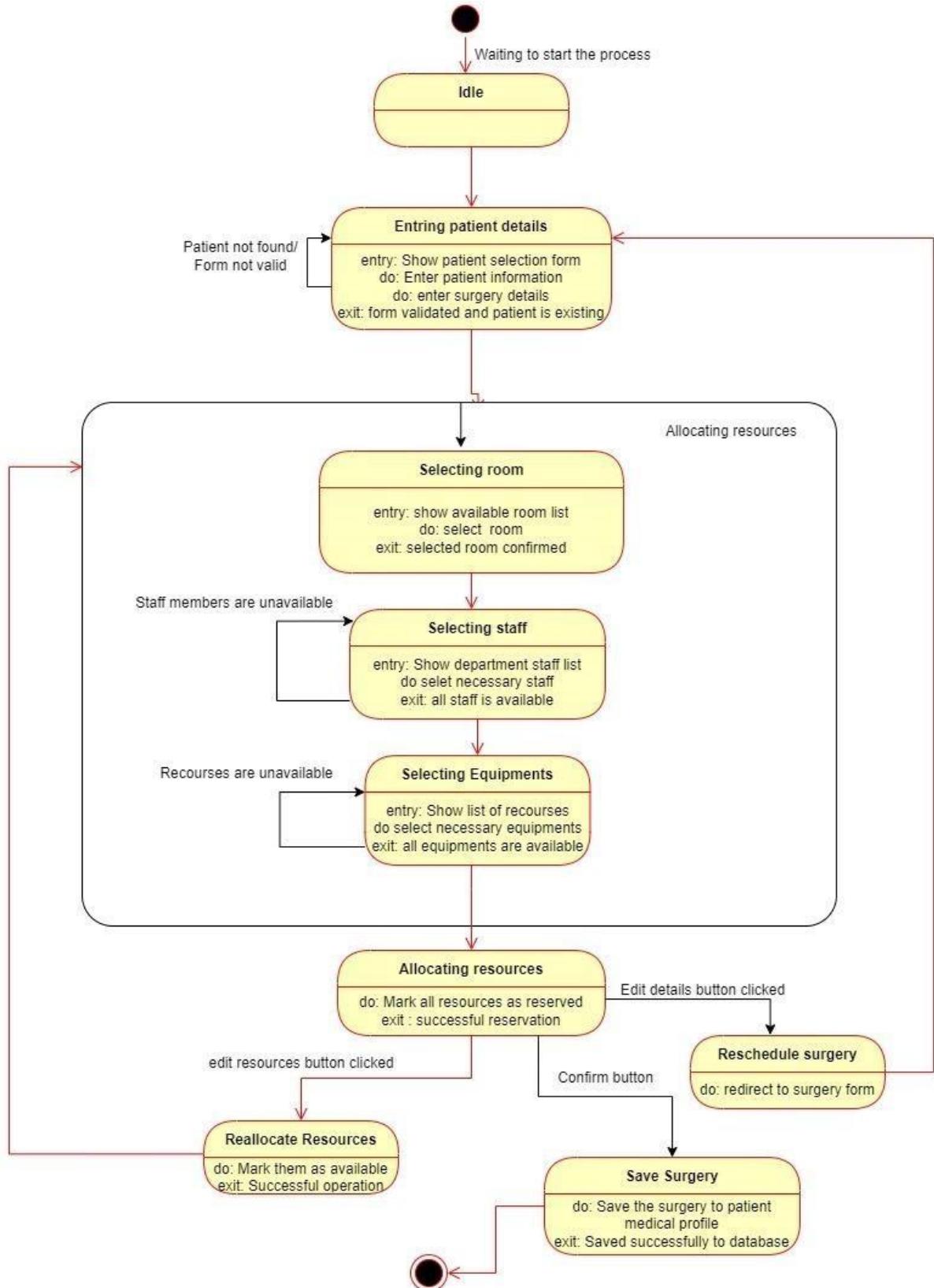
State diagram 3: Models a prescription progresses --- Nikola Rigo



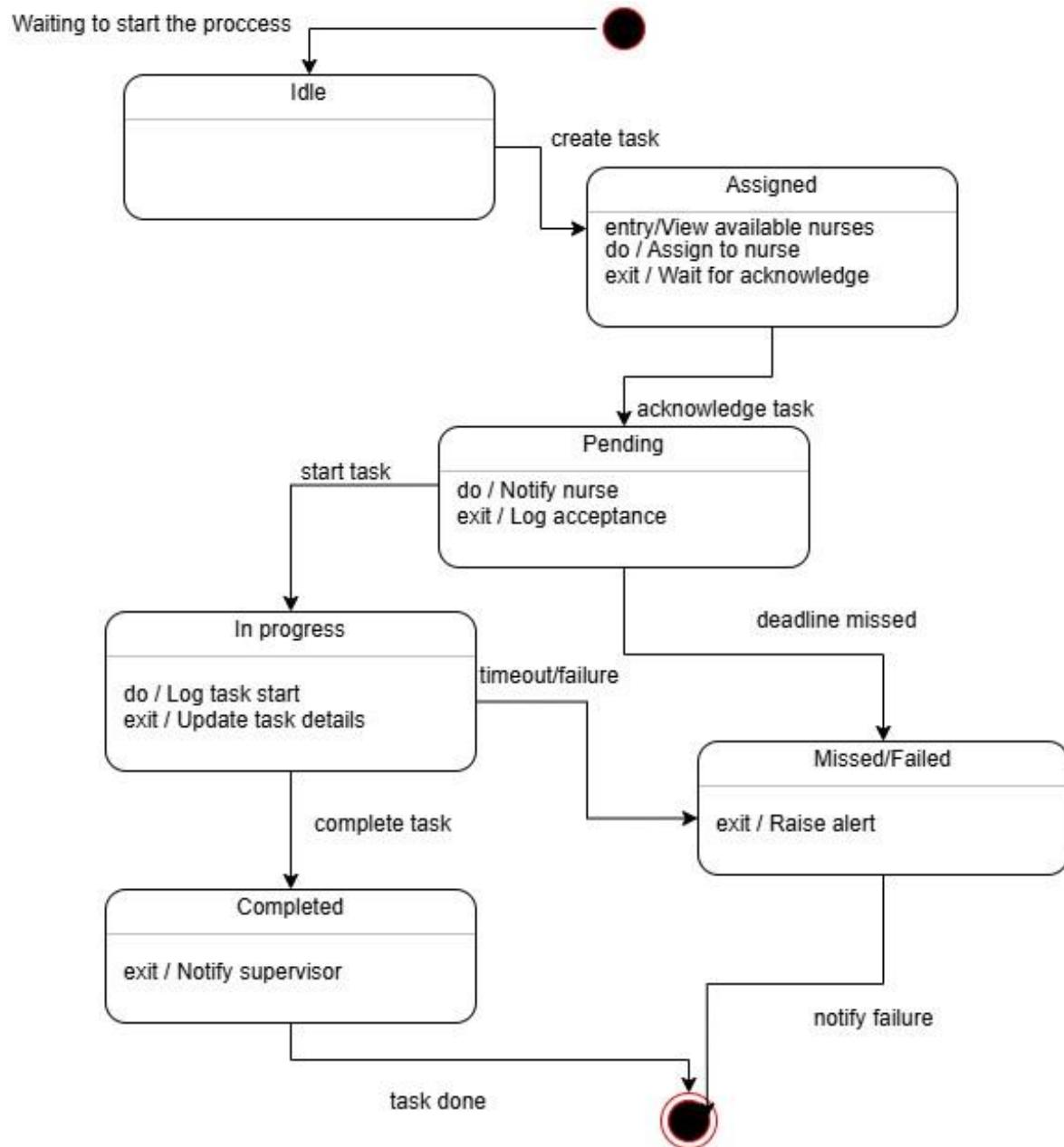
State Diagram 4: Billing State Diagram --- Shpetim Shabanaj



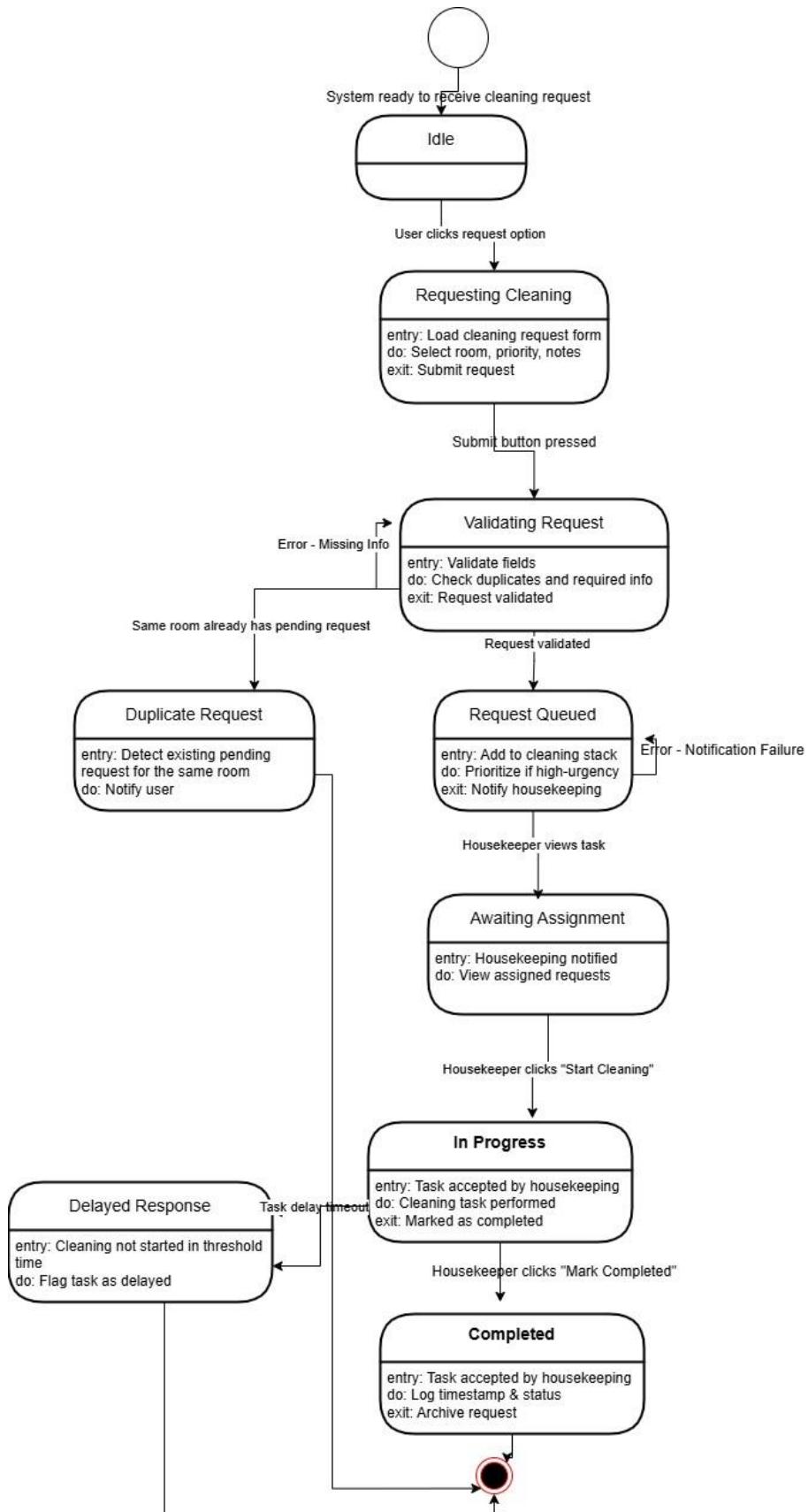
State Diagram 5:Surgery State Diagram --- Shpetim Shabanaj



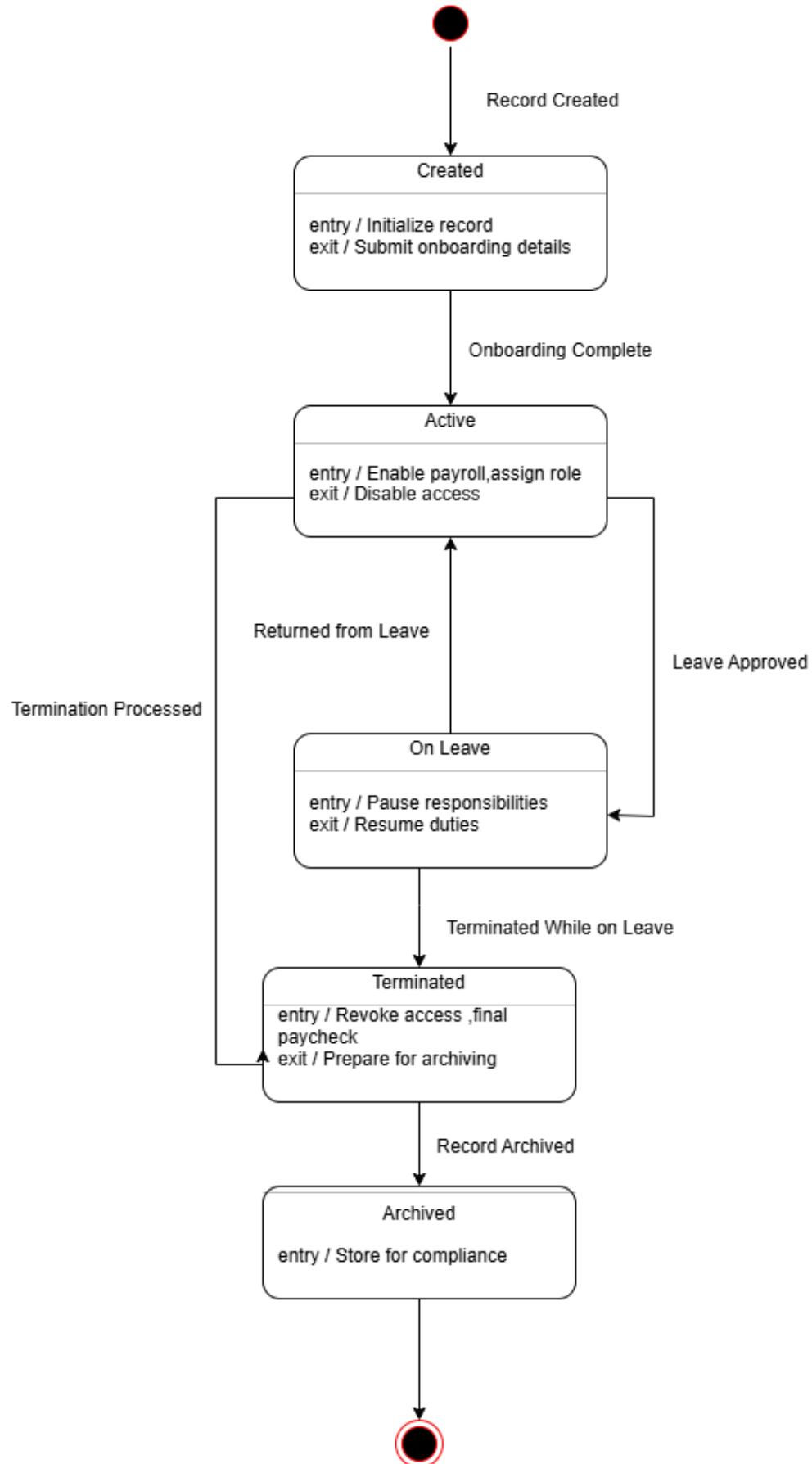
State Diagram 6:Nurse Task State Diagram --- Eglis Braho



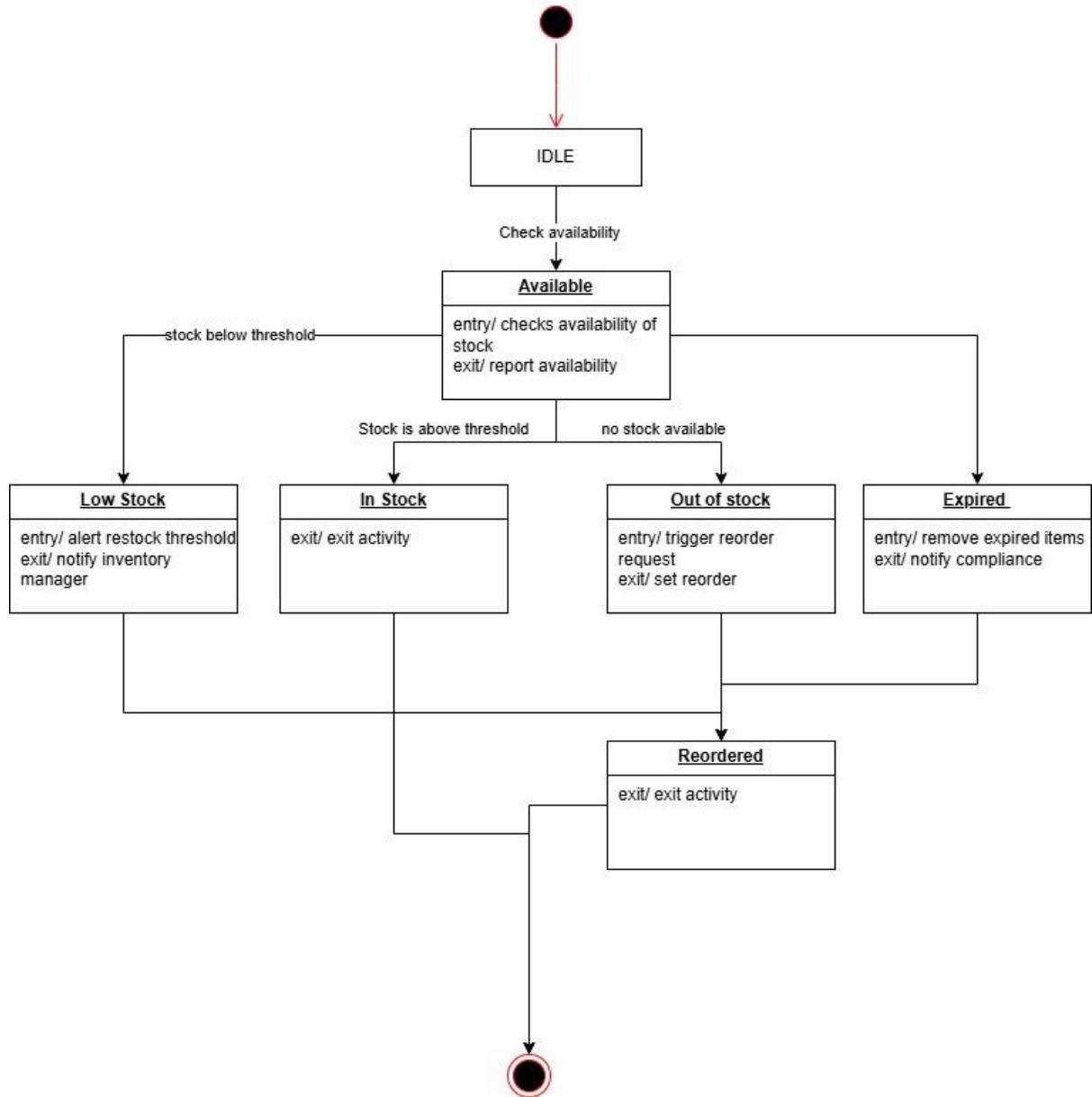
State Diagram 7: Room Cleaning Request State Diagram --- Artjol Zaimi



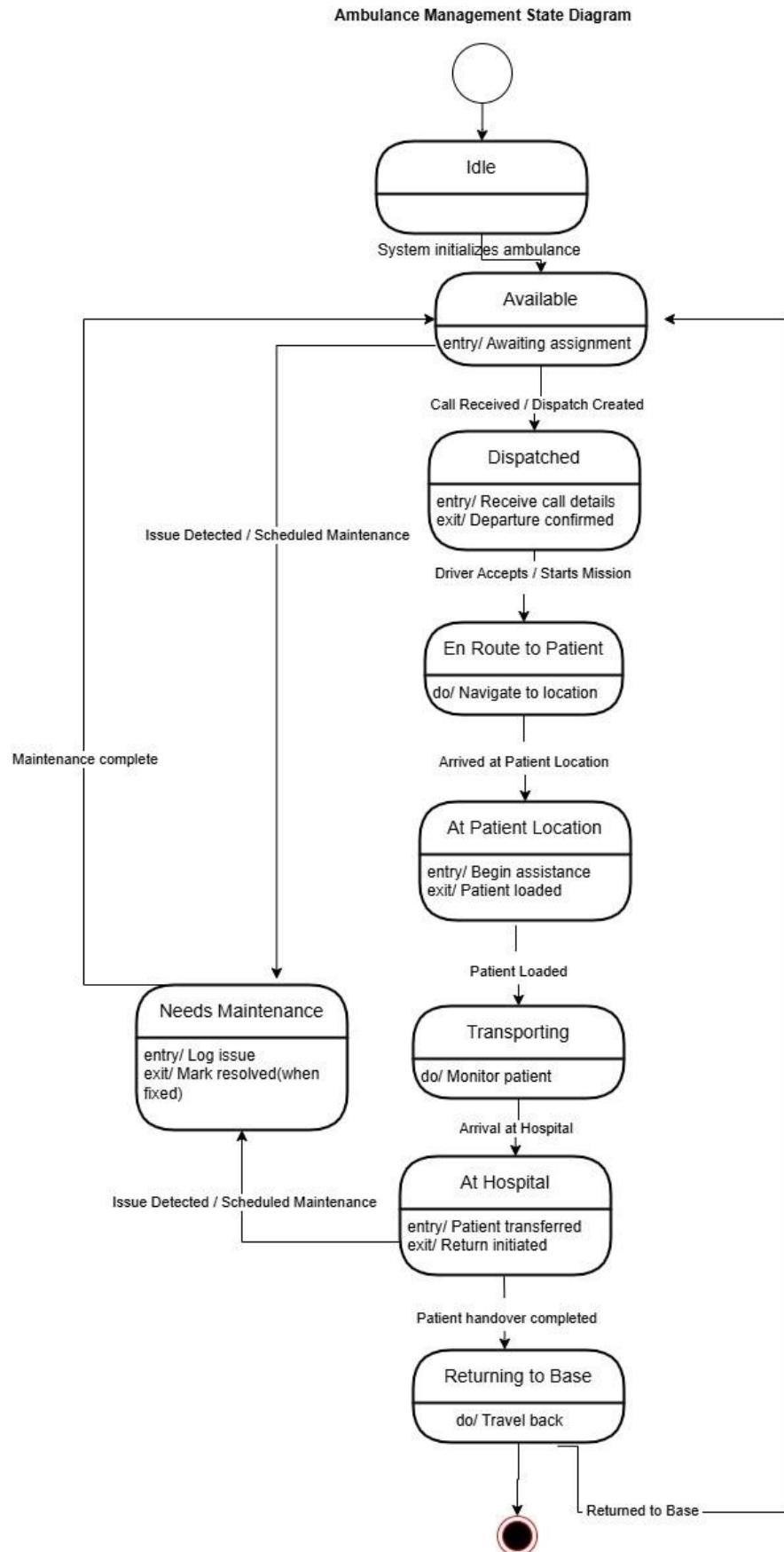
State Diagram 8:HR diagram --- Arlin Bashllari



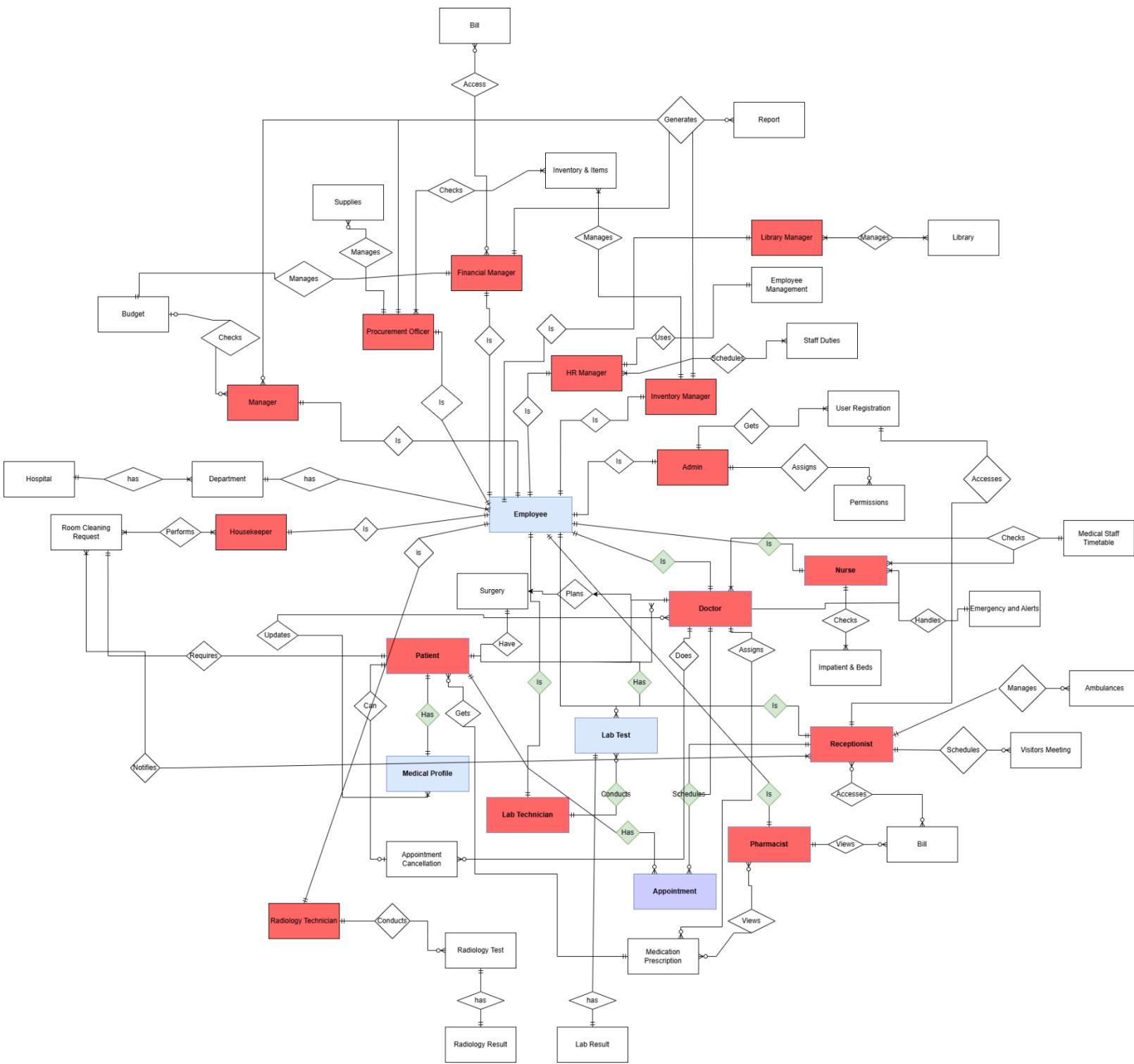
State Diagram 9: Inventory Item State Diagram --- Arjan Muka



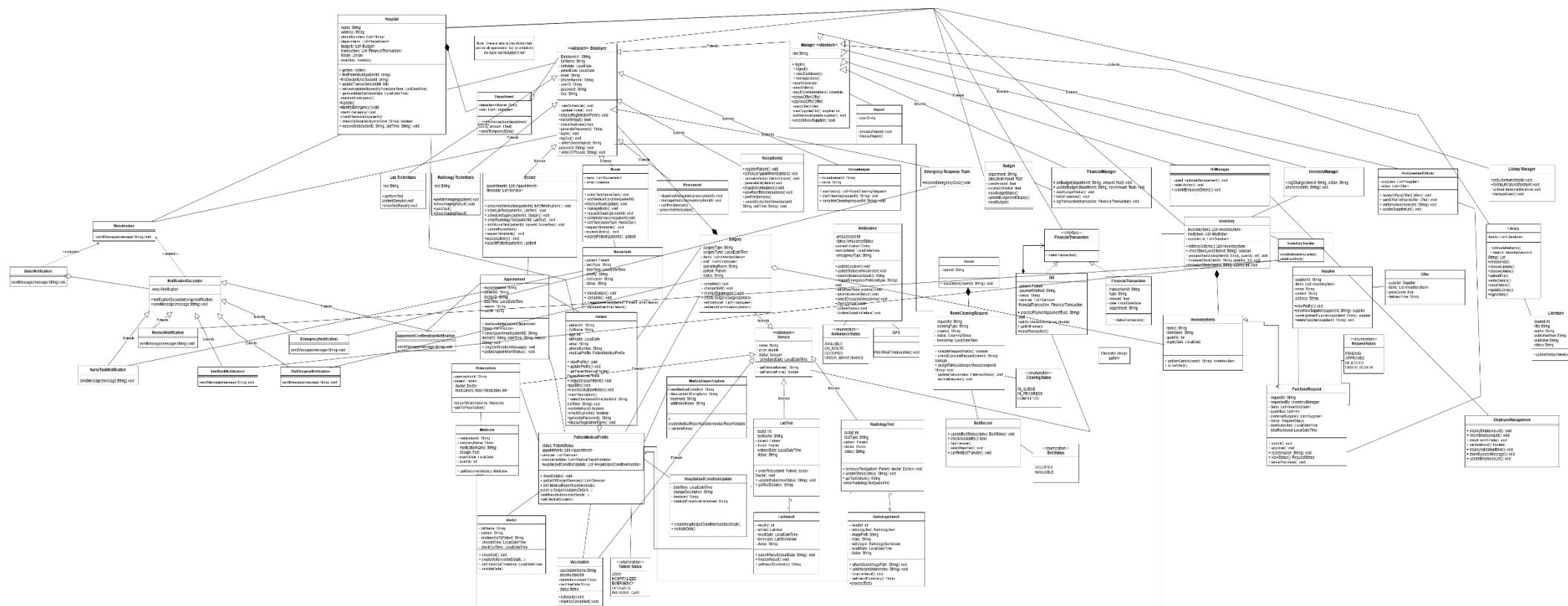
State Diagram 10: Ambulance Management State Diagram --- Artjol Zaimi



ERD

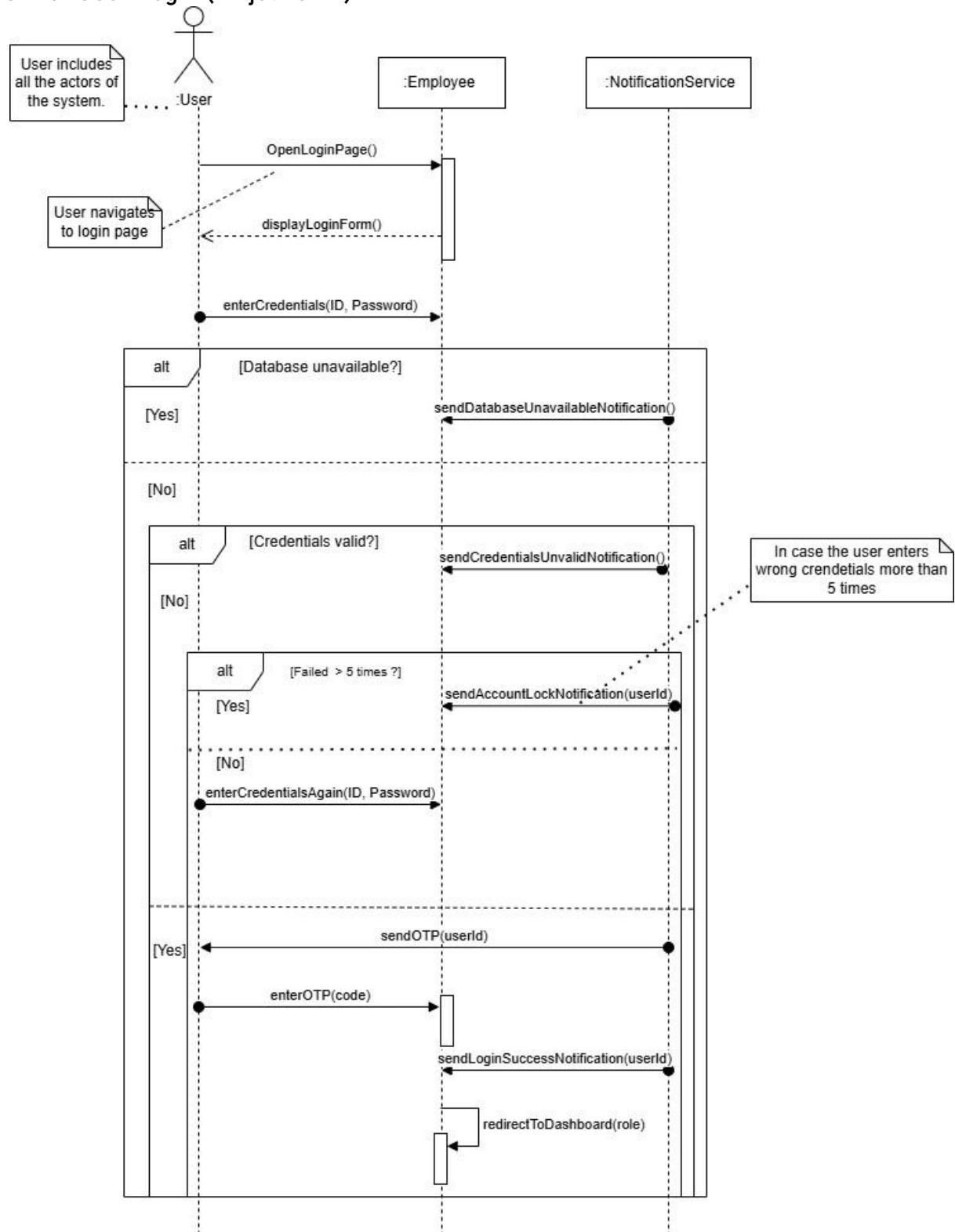


CLASS DIAGRAM

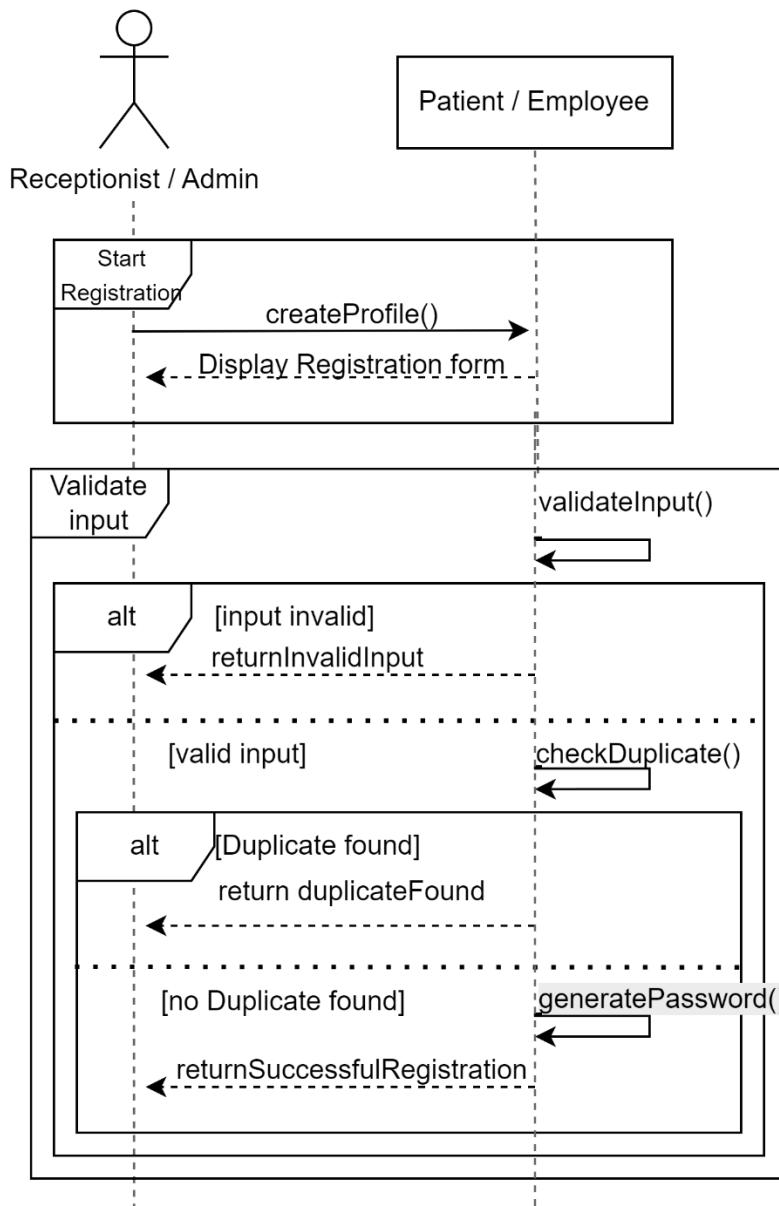


Sequence Diagrams

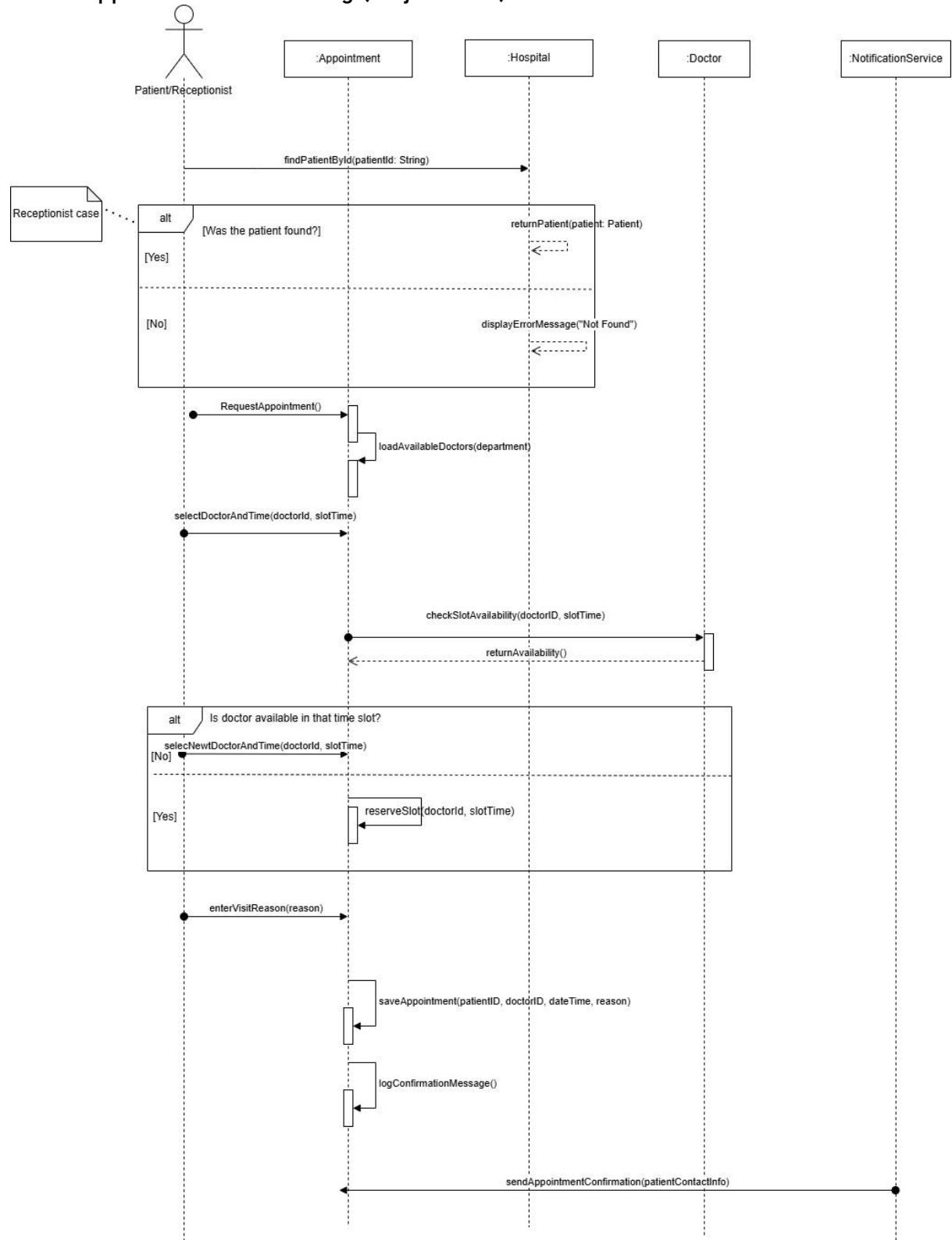
SD-01 User Login (Artjol Zaimi)



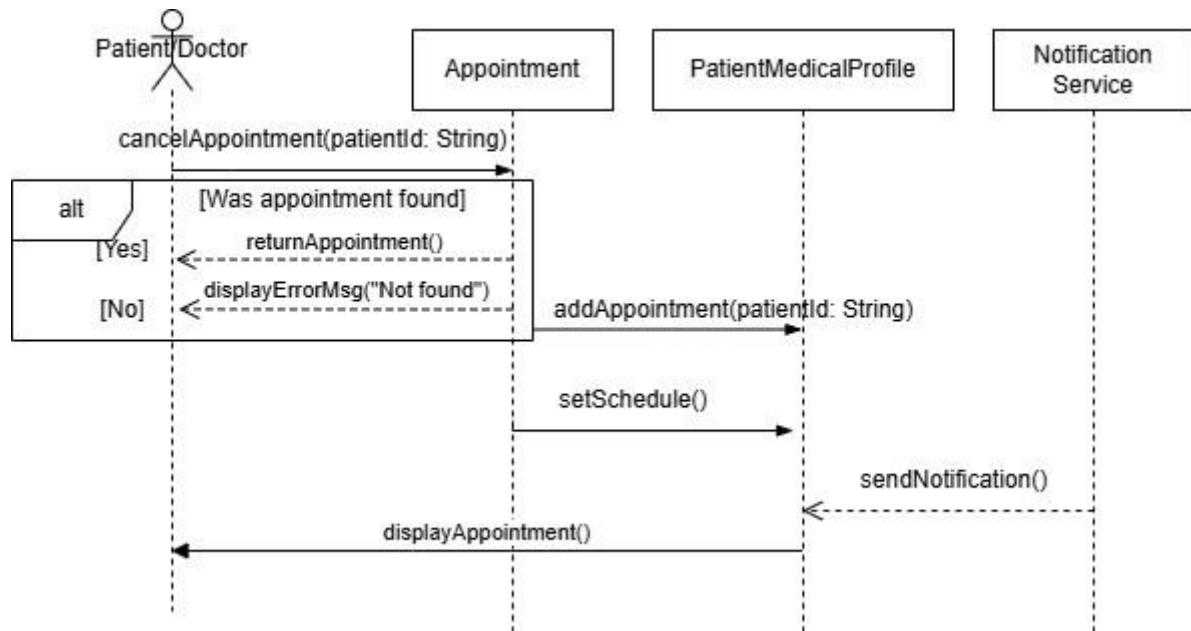
SD-02 Registration (Marin Tartaraj)



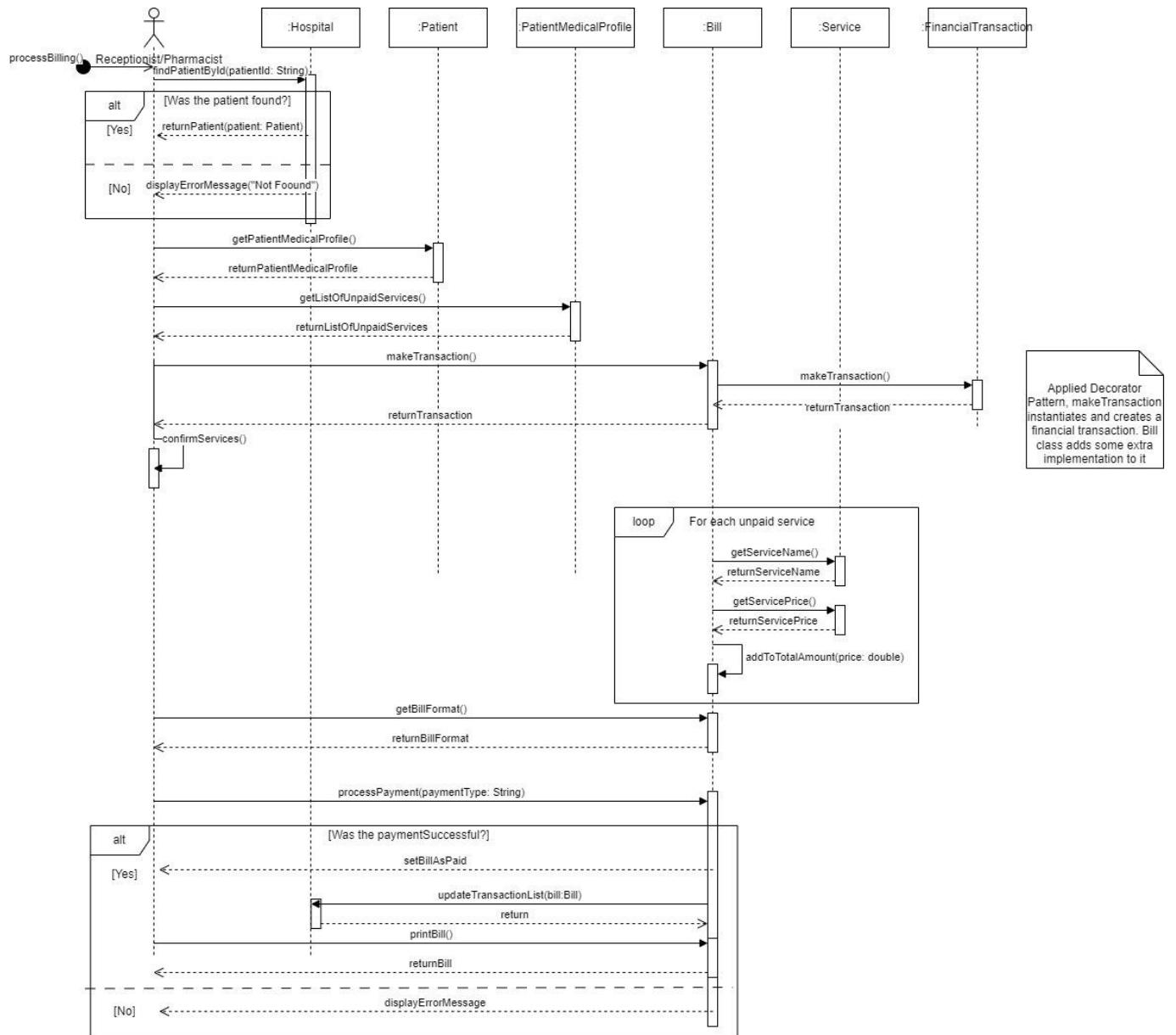
SD-03 Appointment Scheduling (Artjol Zaimi)



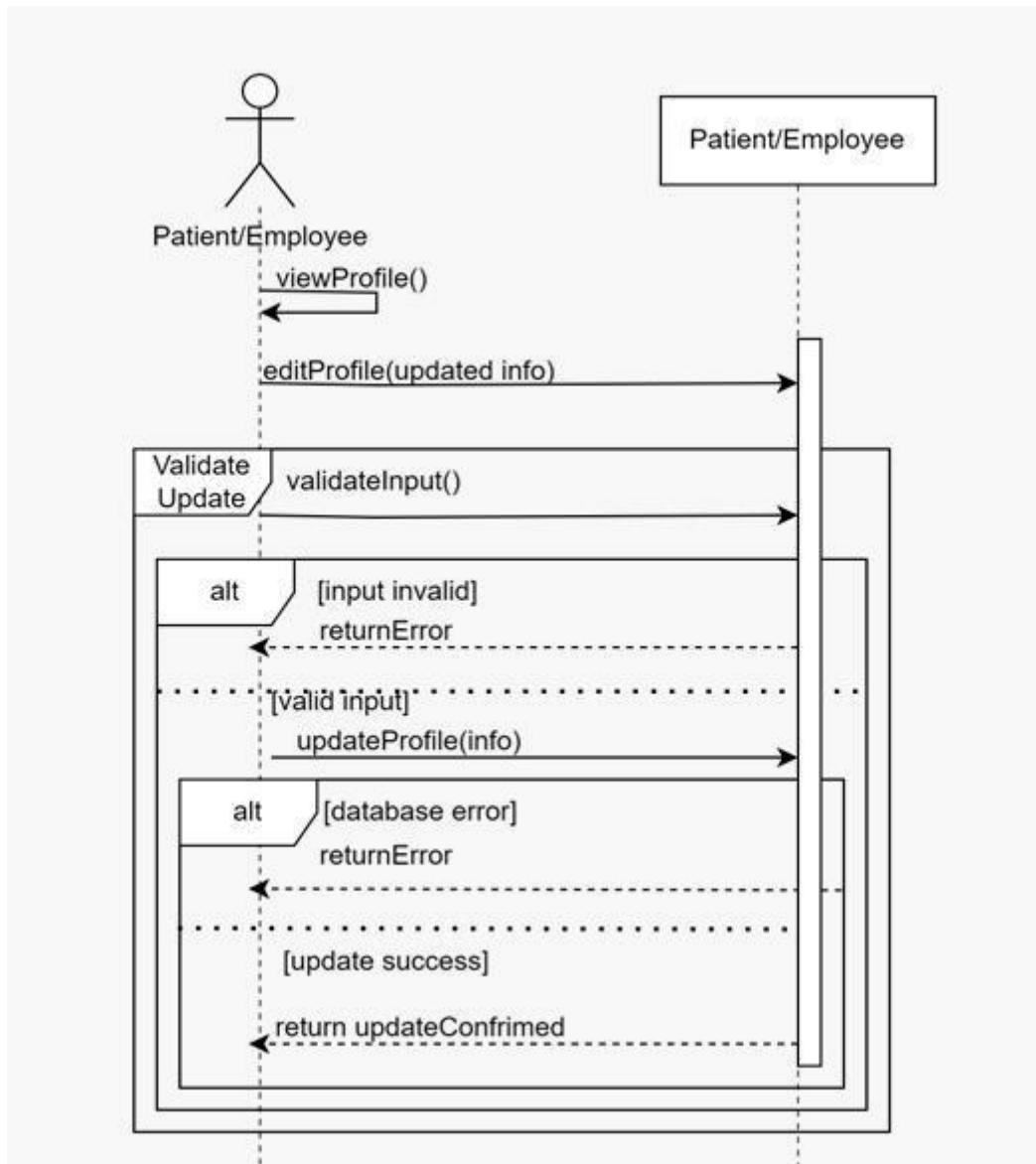
04 Appointment Cancellation (Arlin Bashllari)



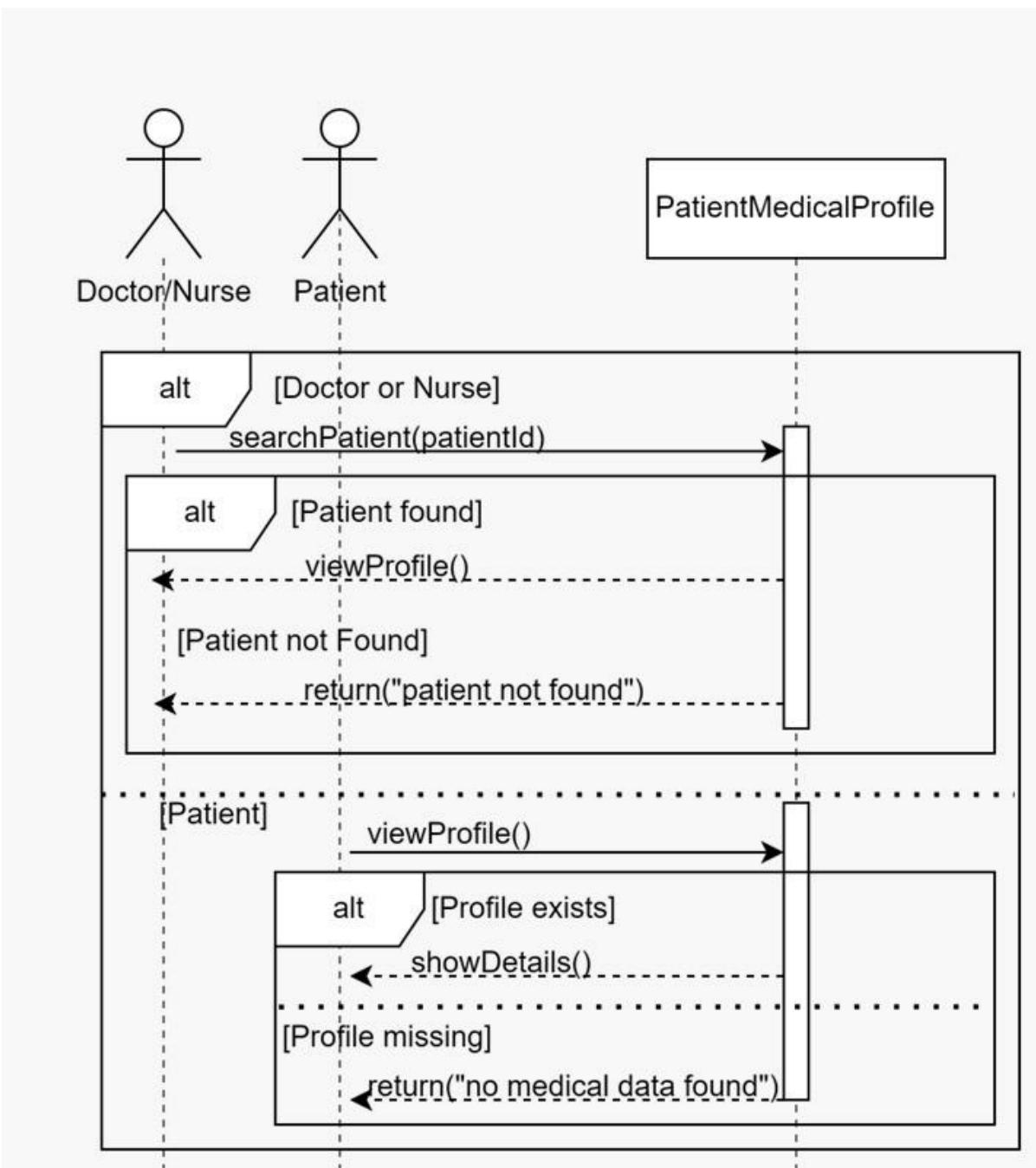
05 Billing Component & Payment Processing (Shpetim Shabanaj)



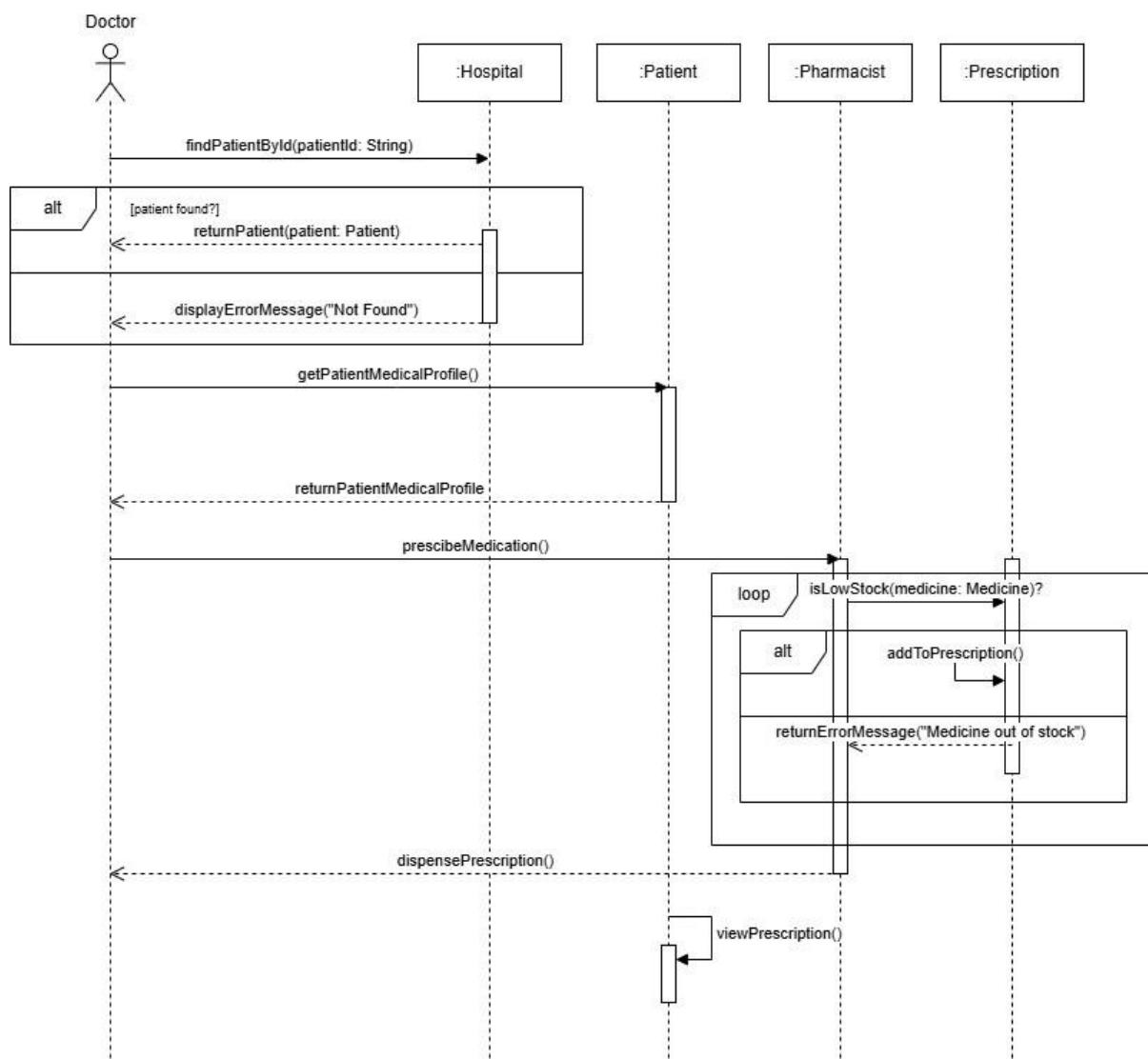
06 Profile Management (Marin Tartara)



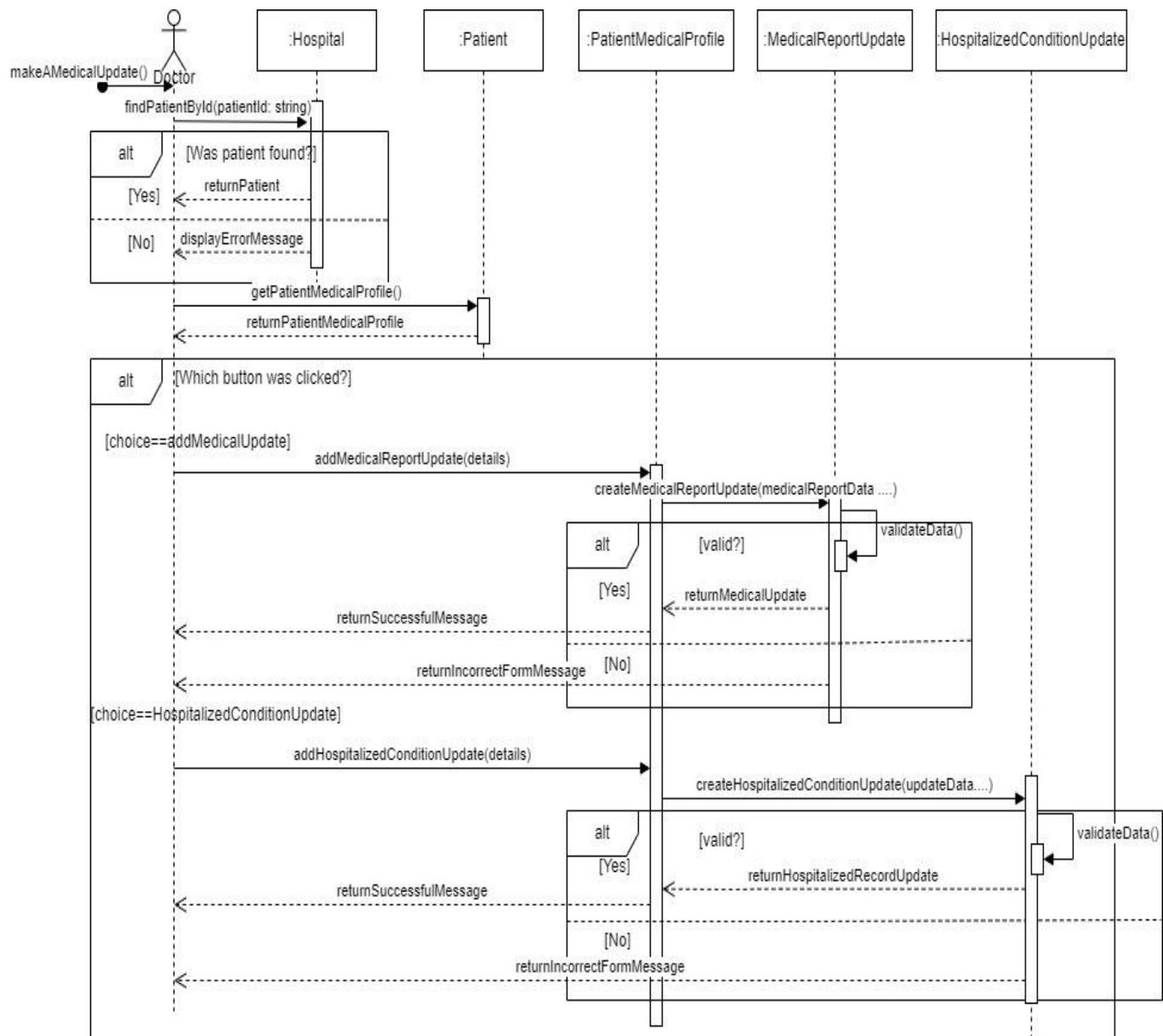
07 Patient Medical Profile()



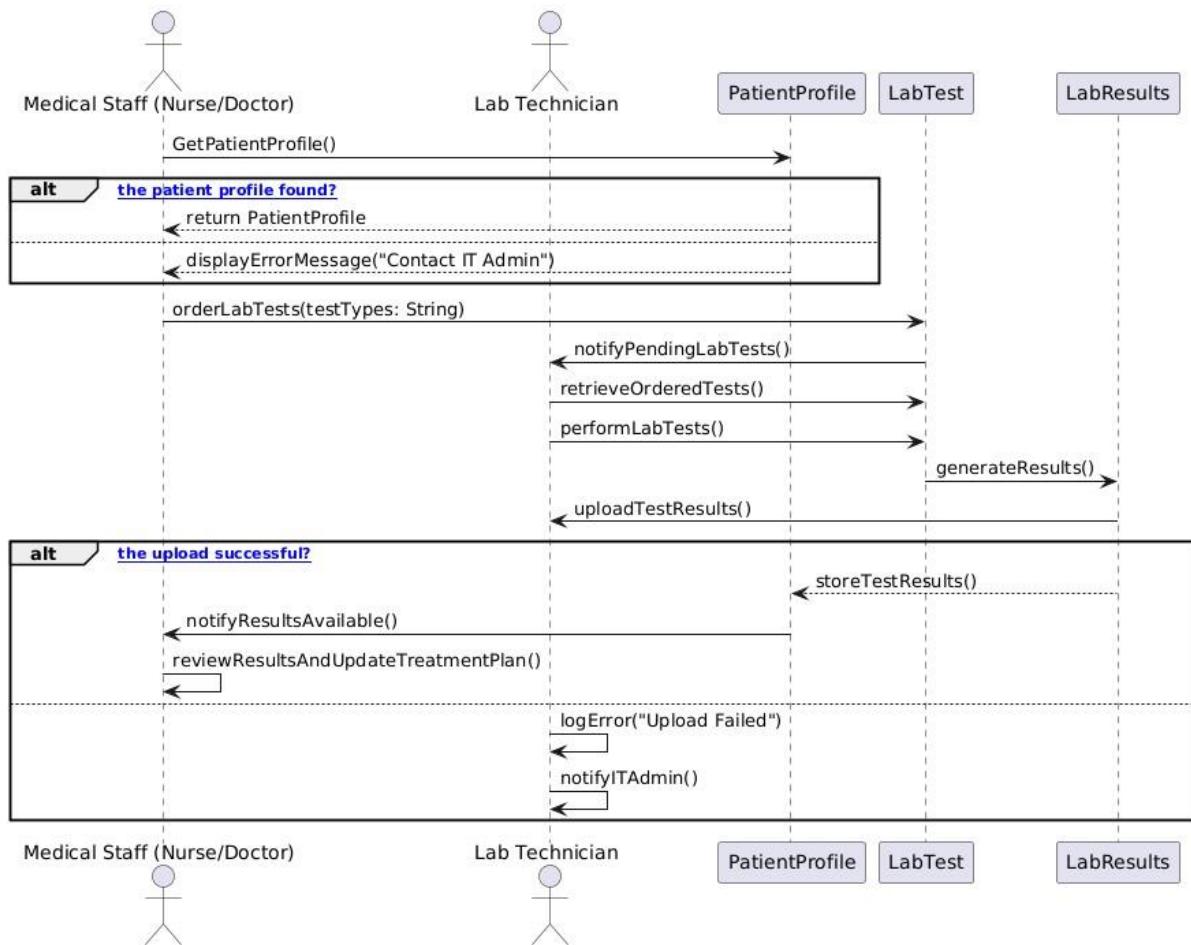
08 Medication Prescription & Viewing(Nikola Rigo)



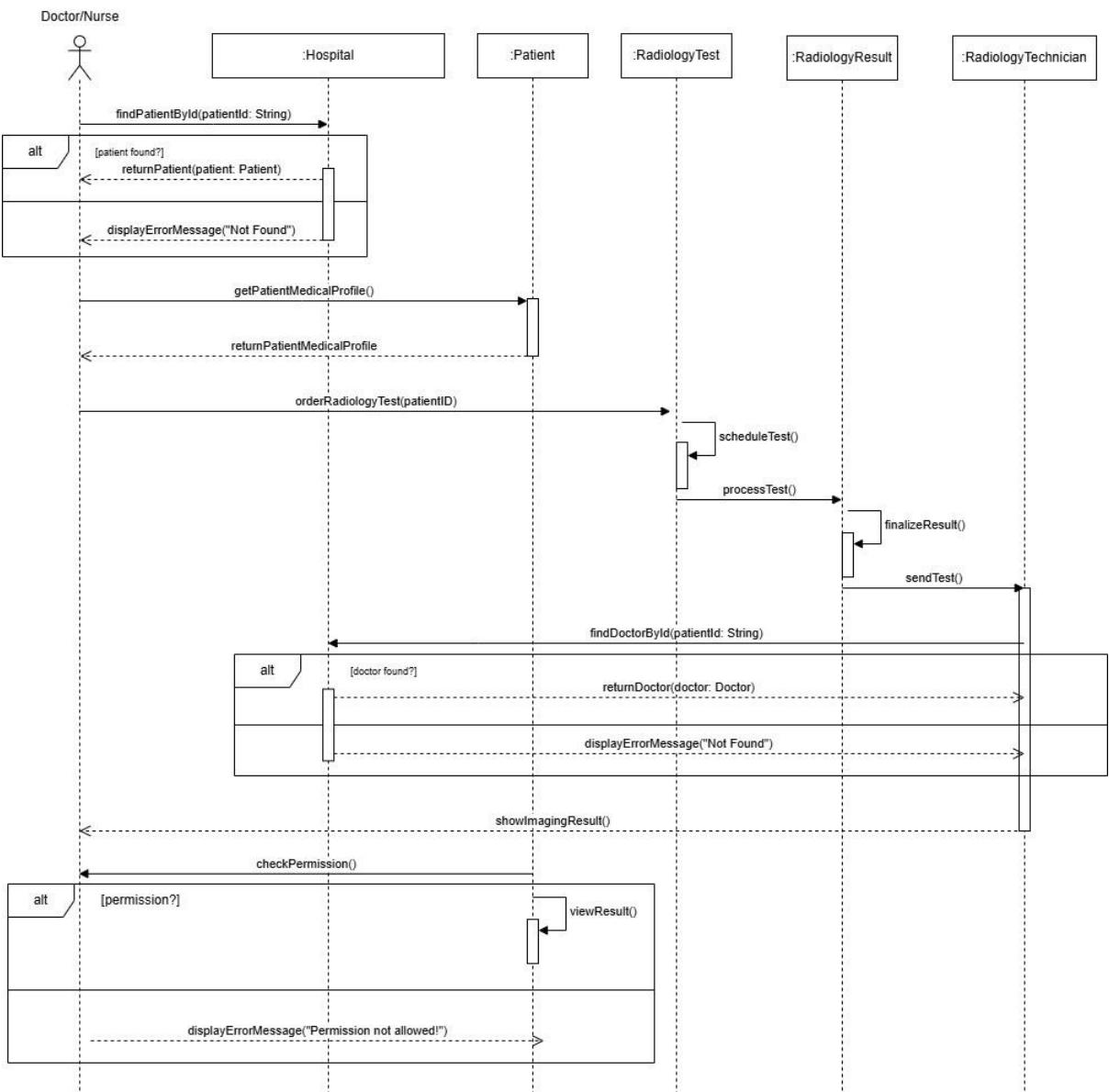
09 Electronic Health Records Update (Shpetim Shabana)



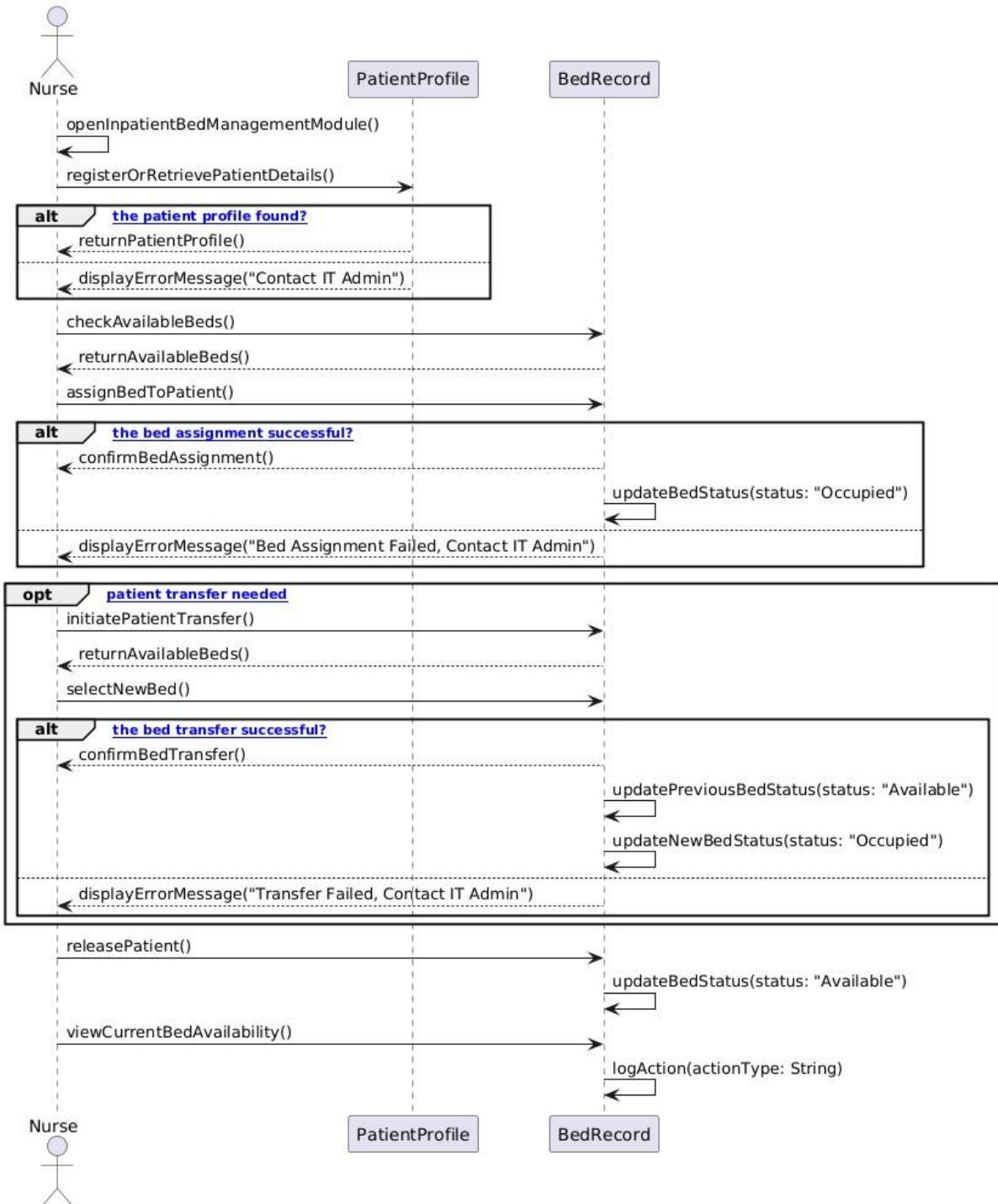
10 Lab Test Ordering and Result Upload (Arjan Muka)



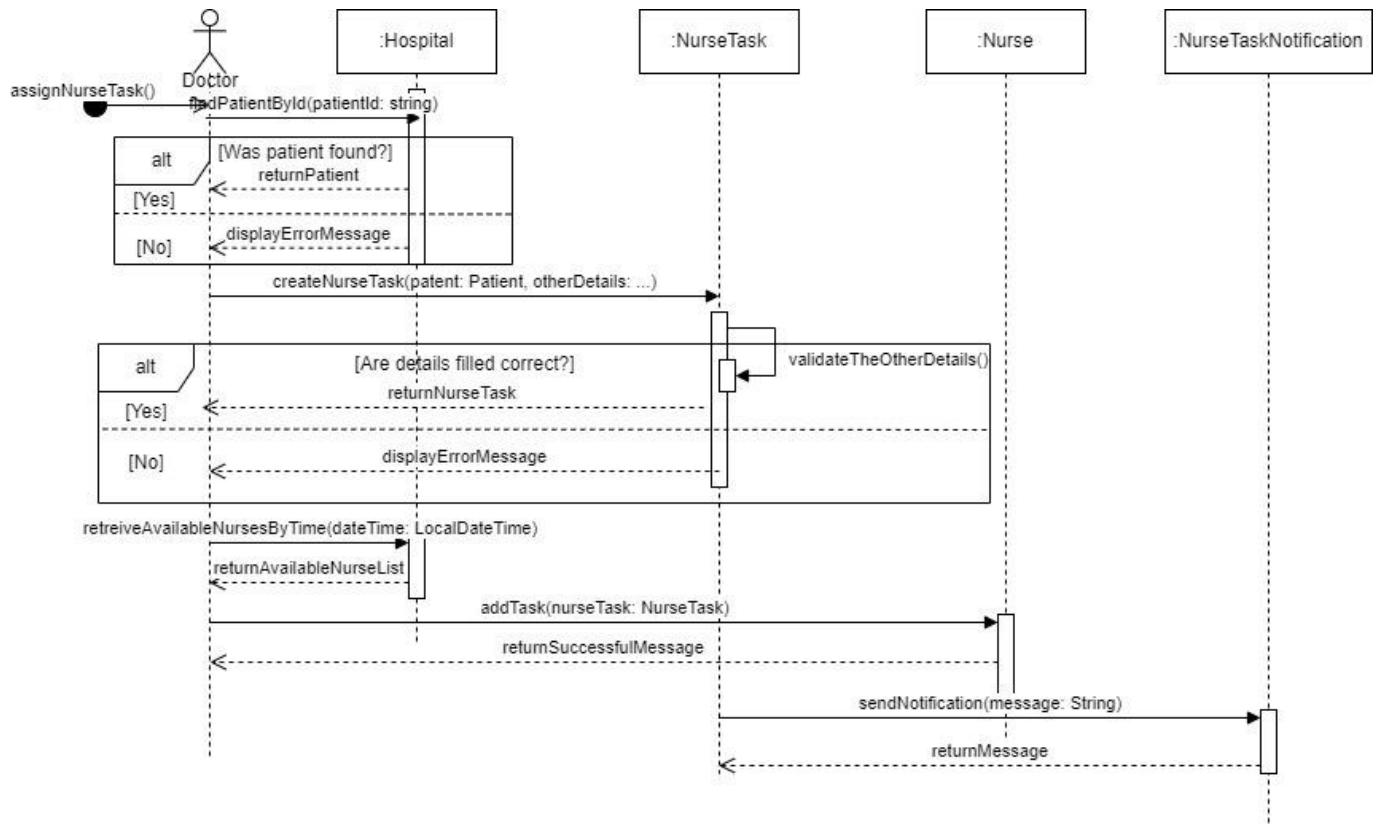
11 Radiology & Imaging Ordering and Result Upload(Nikola Rigo)



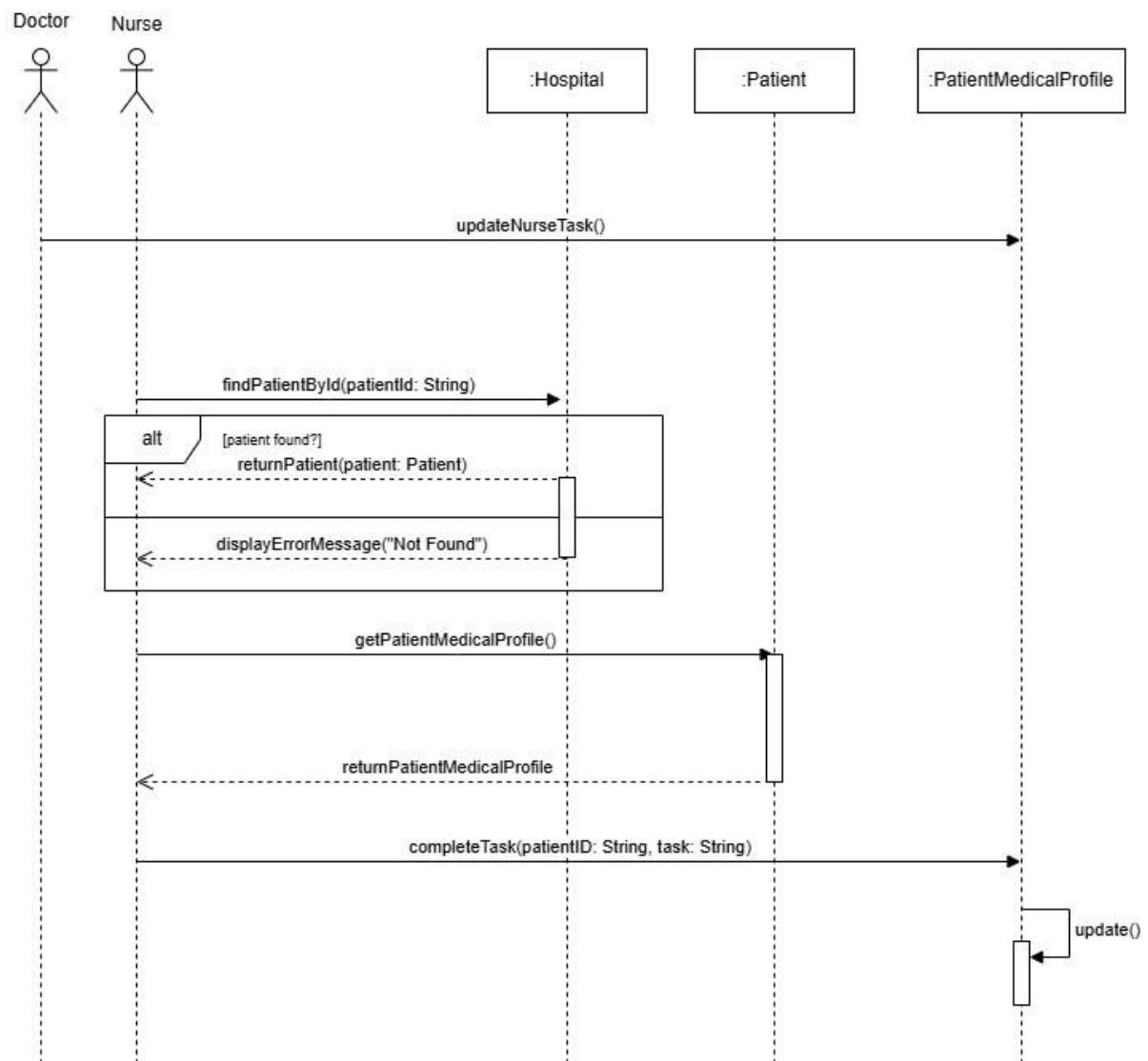
12 Inpatient & Bed Management (Arjan Muka)



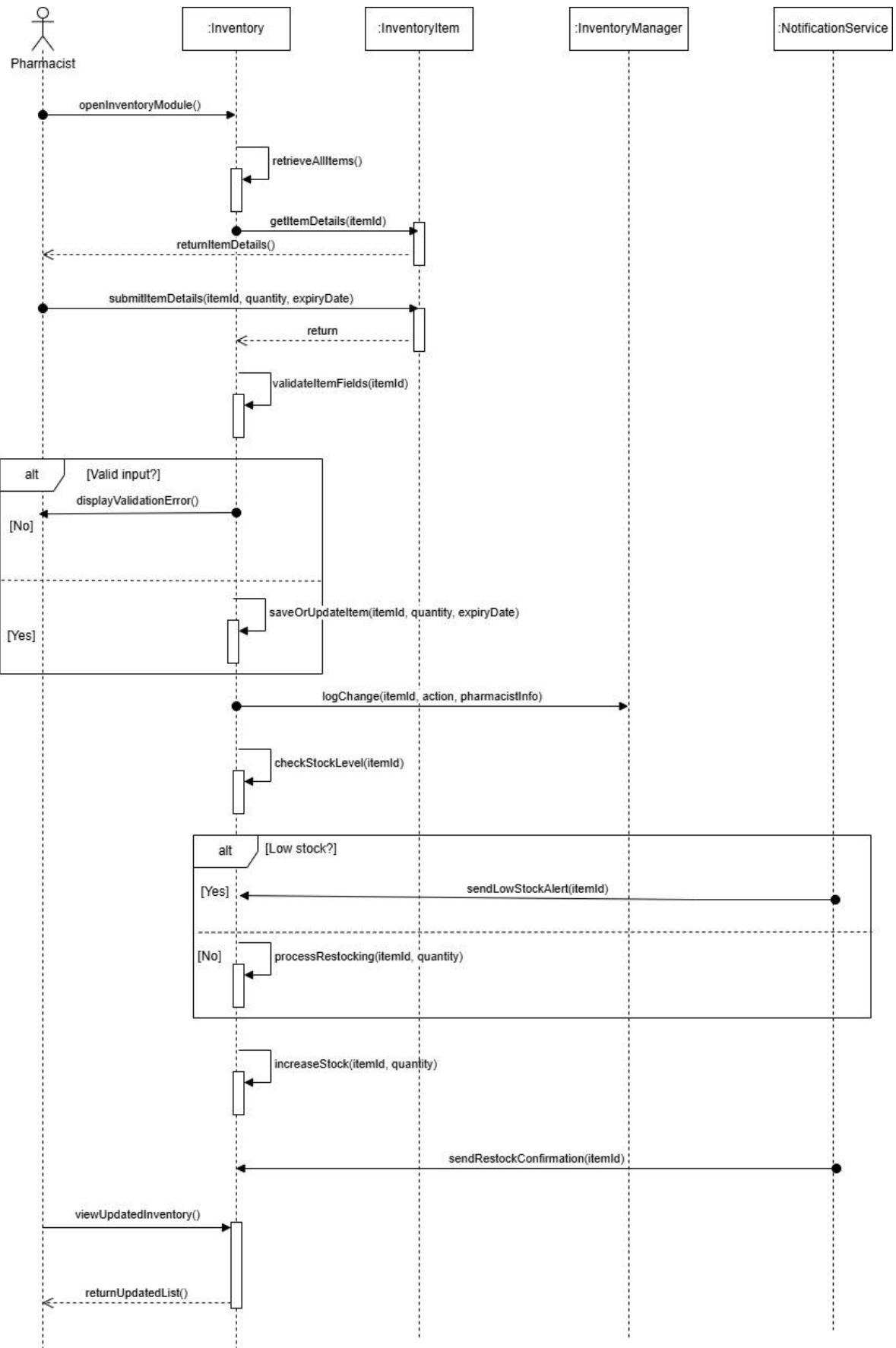
13 Nurse Task Assignment (Shpetim Shabanaj)



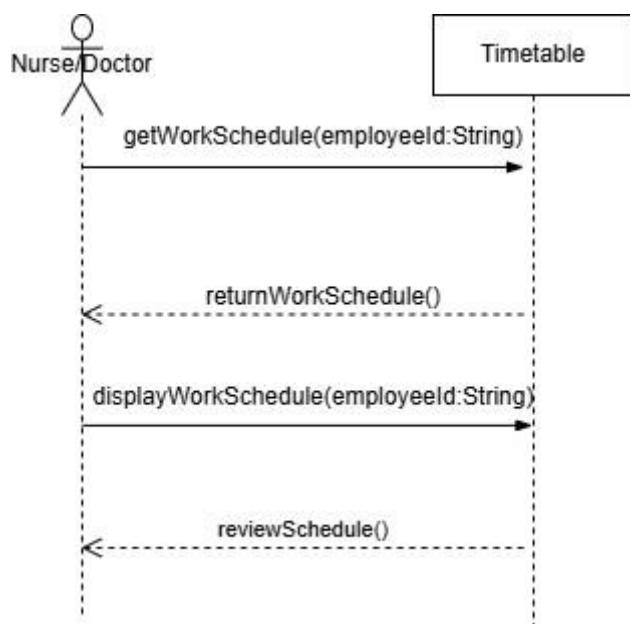
14 Medication Processing for patients (Nikola Rigo)



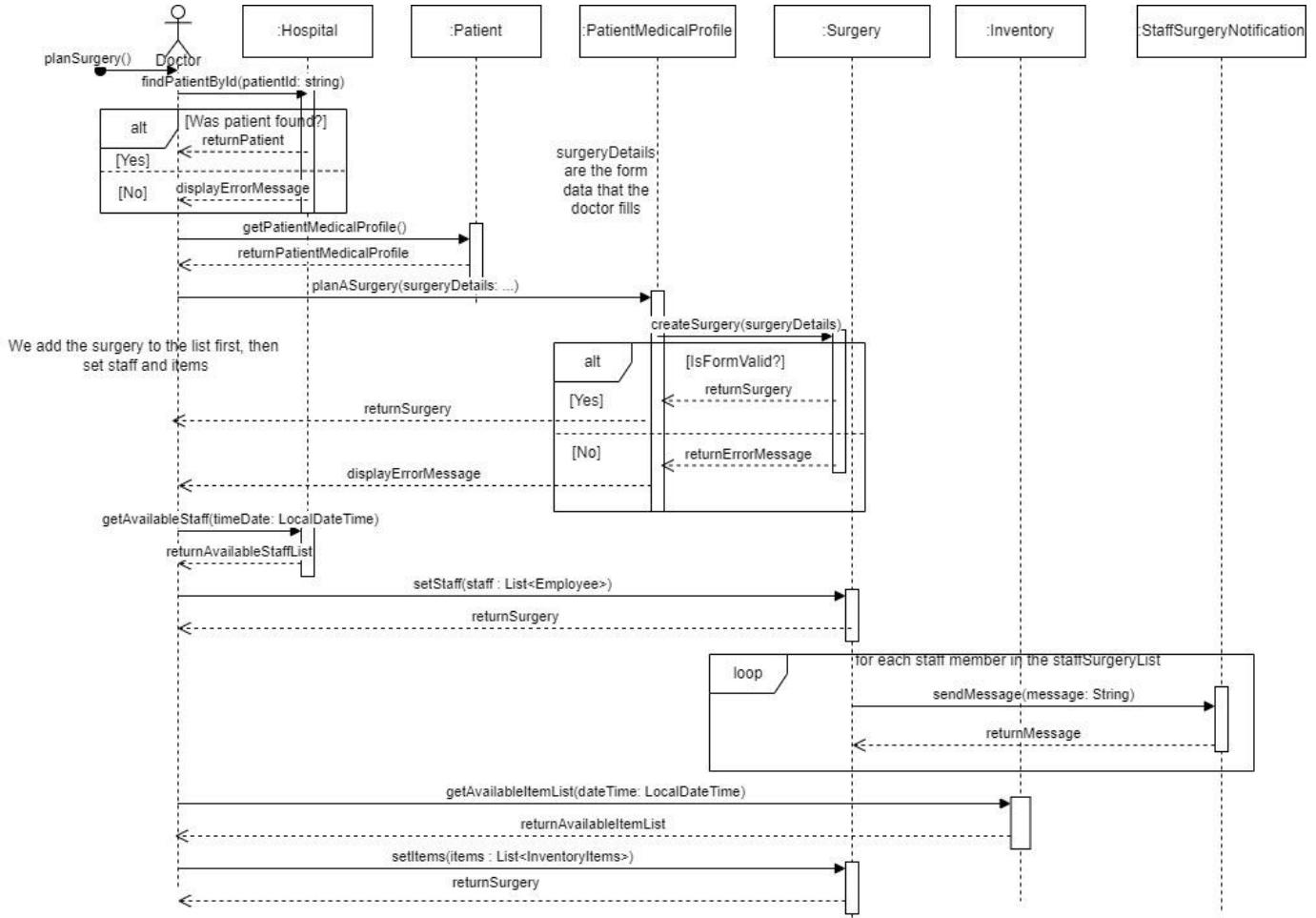
15 Pharmacy & Stock Management (Artjol Zaimi)



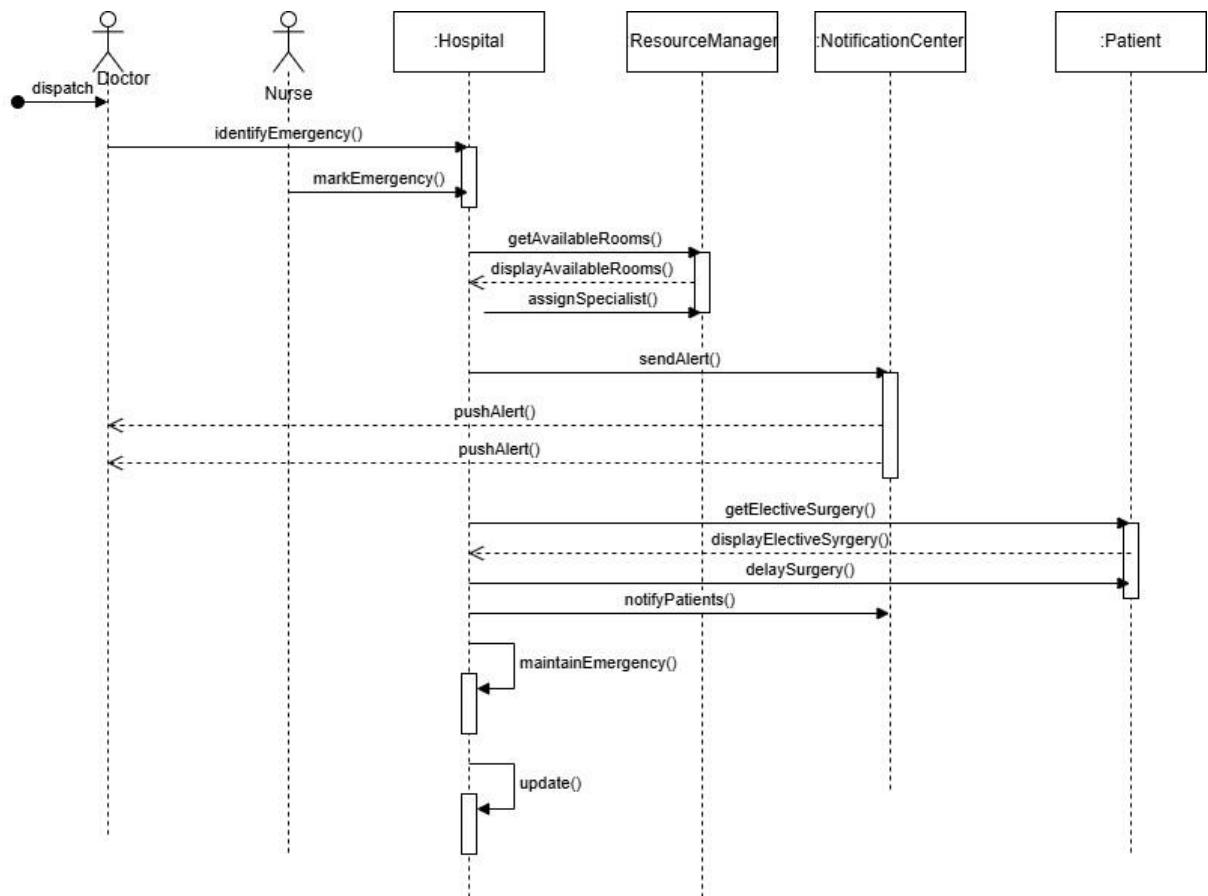
16 Medical Staff Timetable (Arlin Bashllari)



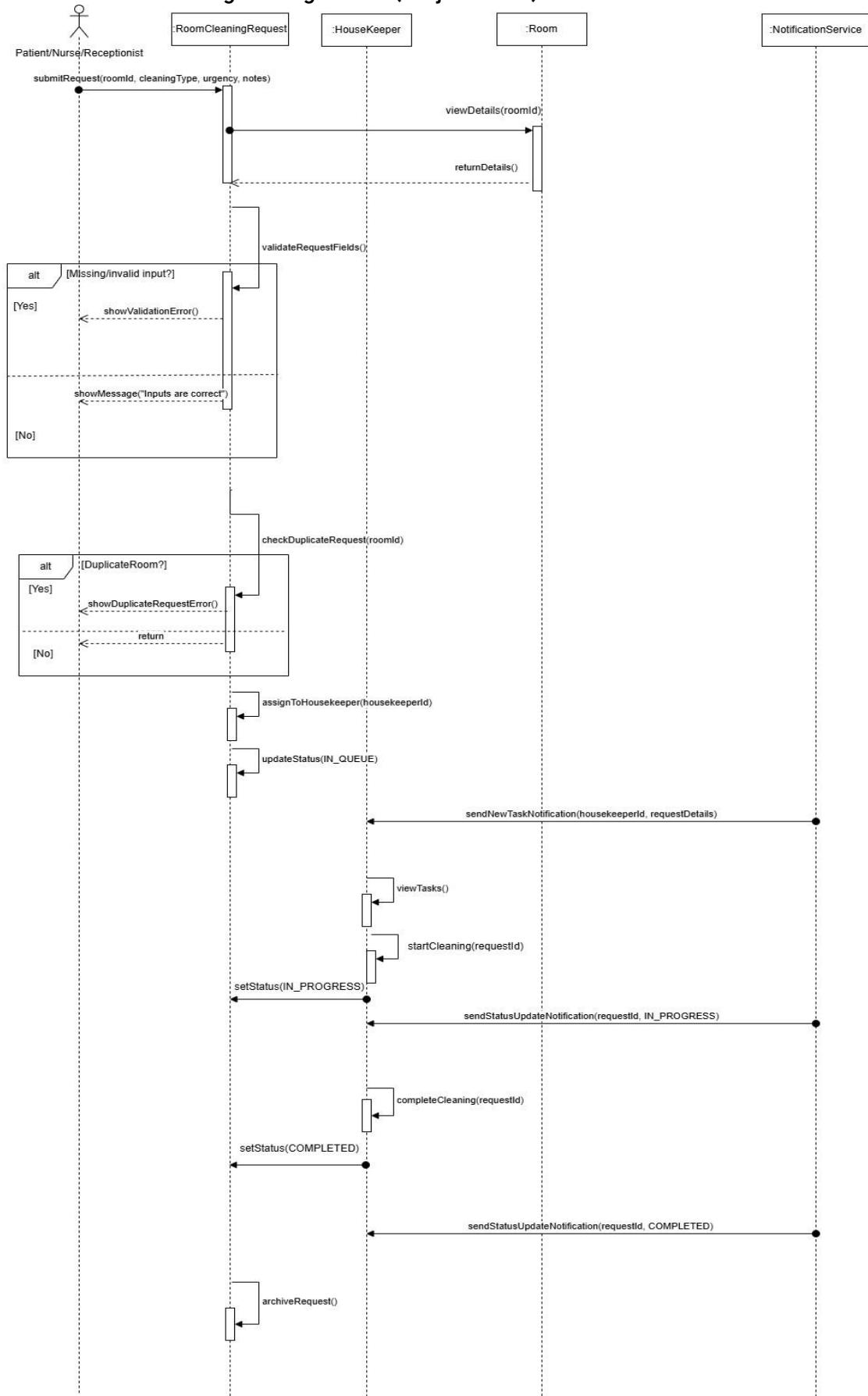
17 Surgery Planning (Shpetim Shabana)



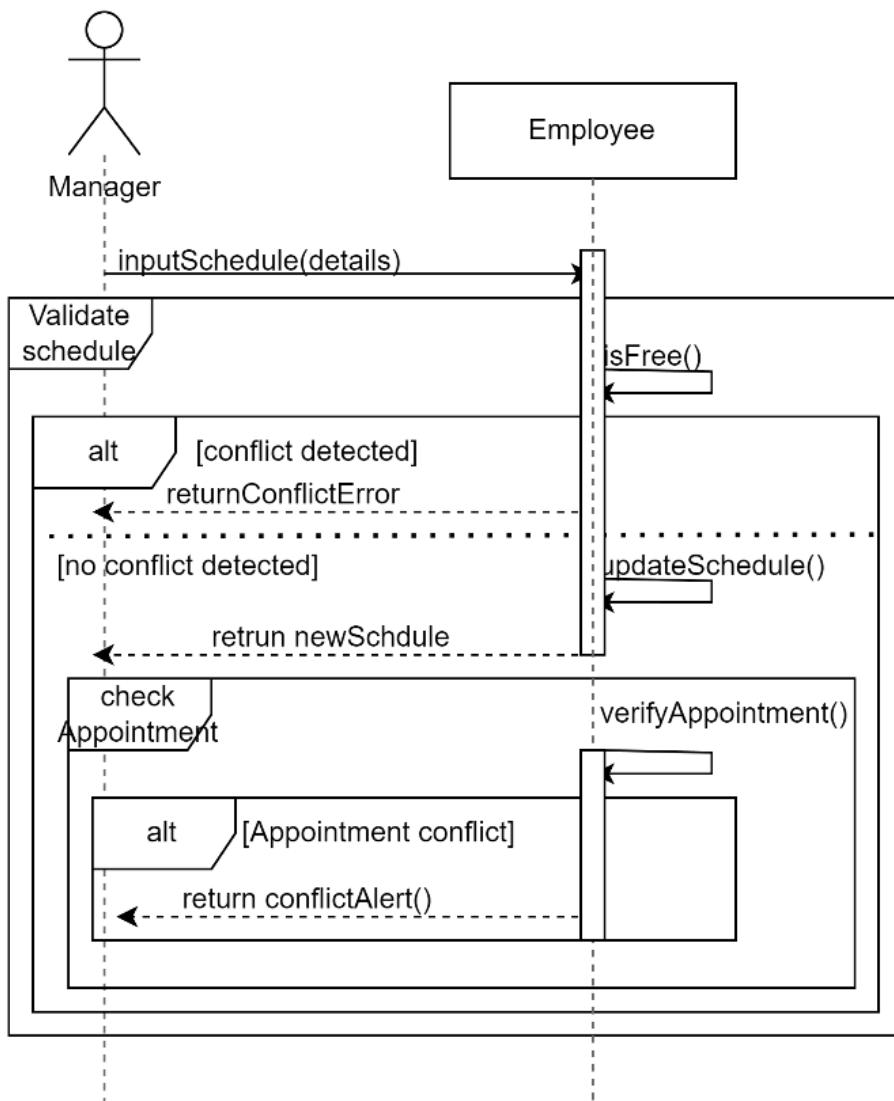
18 Emergency Handling and Alerts (Eglis Braho)



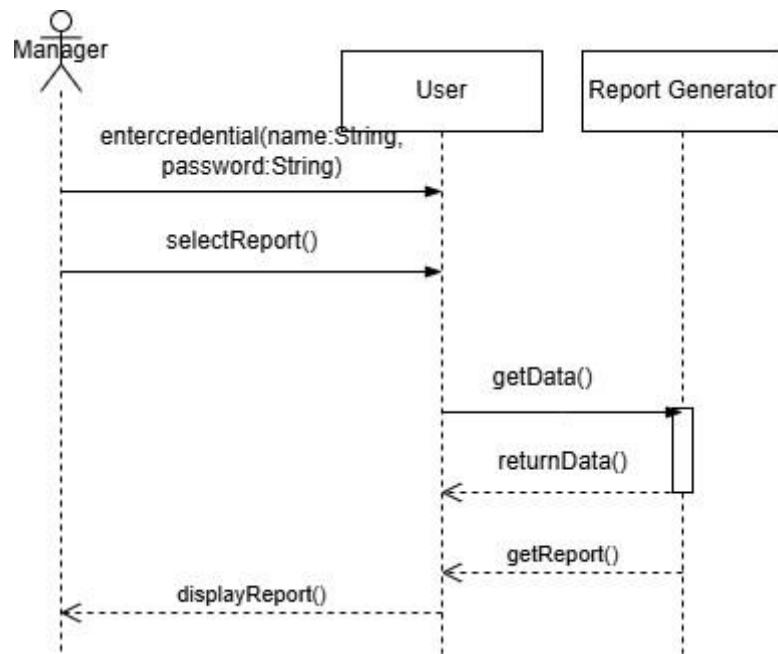
SD-19 Room Cleaning Management (Artjol Zaimi)



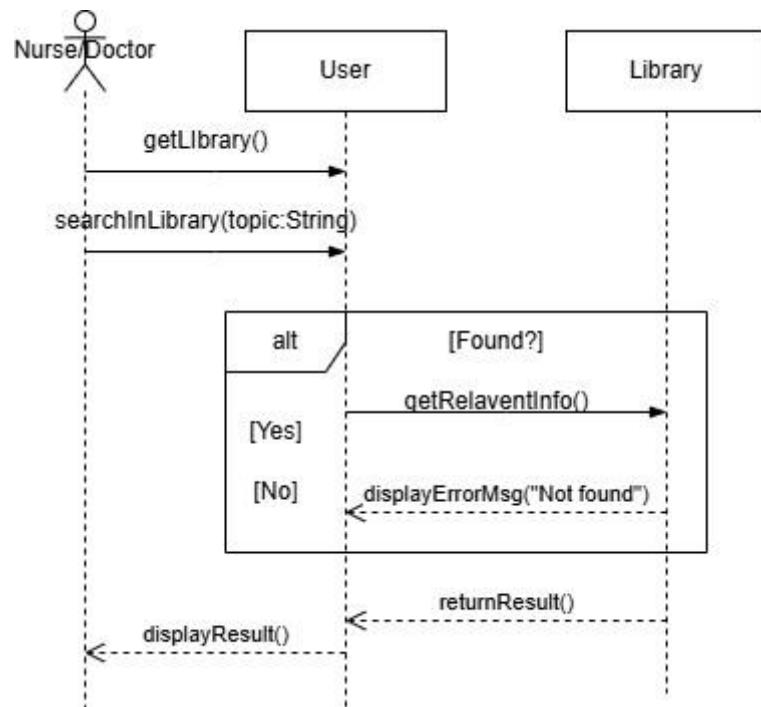
20 Staff Scheduling (Marin Tartaraj)



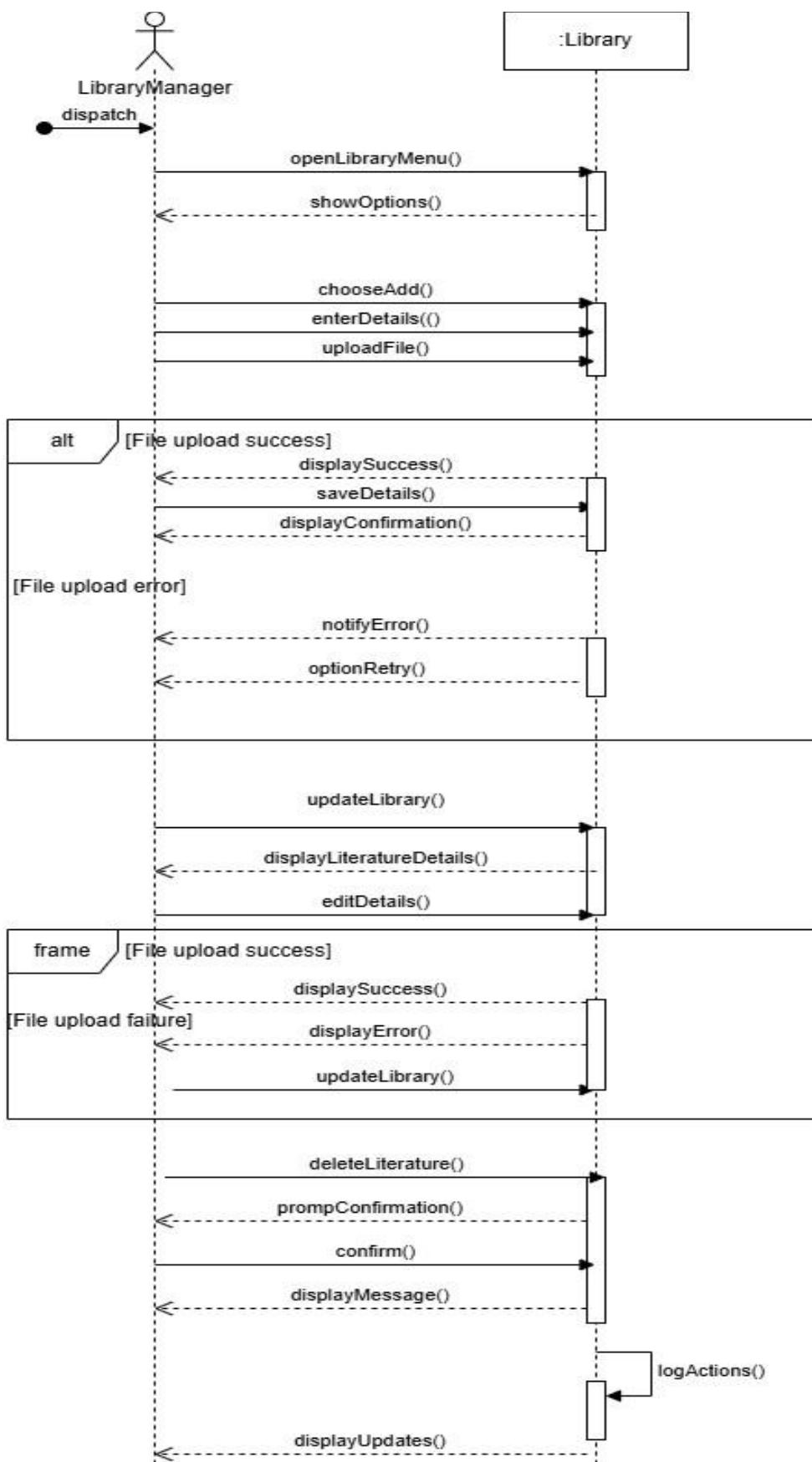
21 Report Generation (Arlin Bashllari)

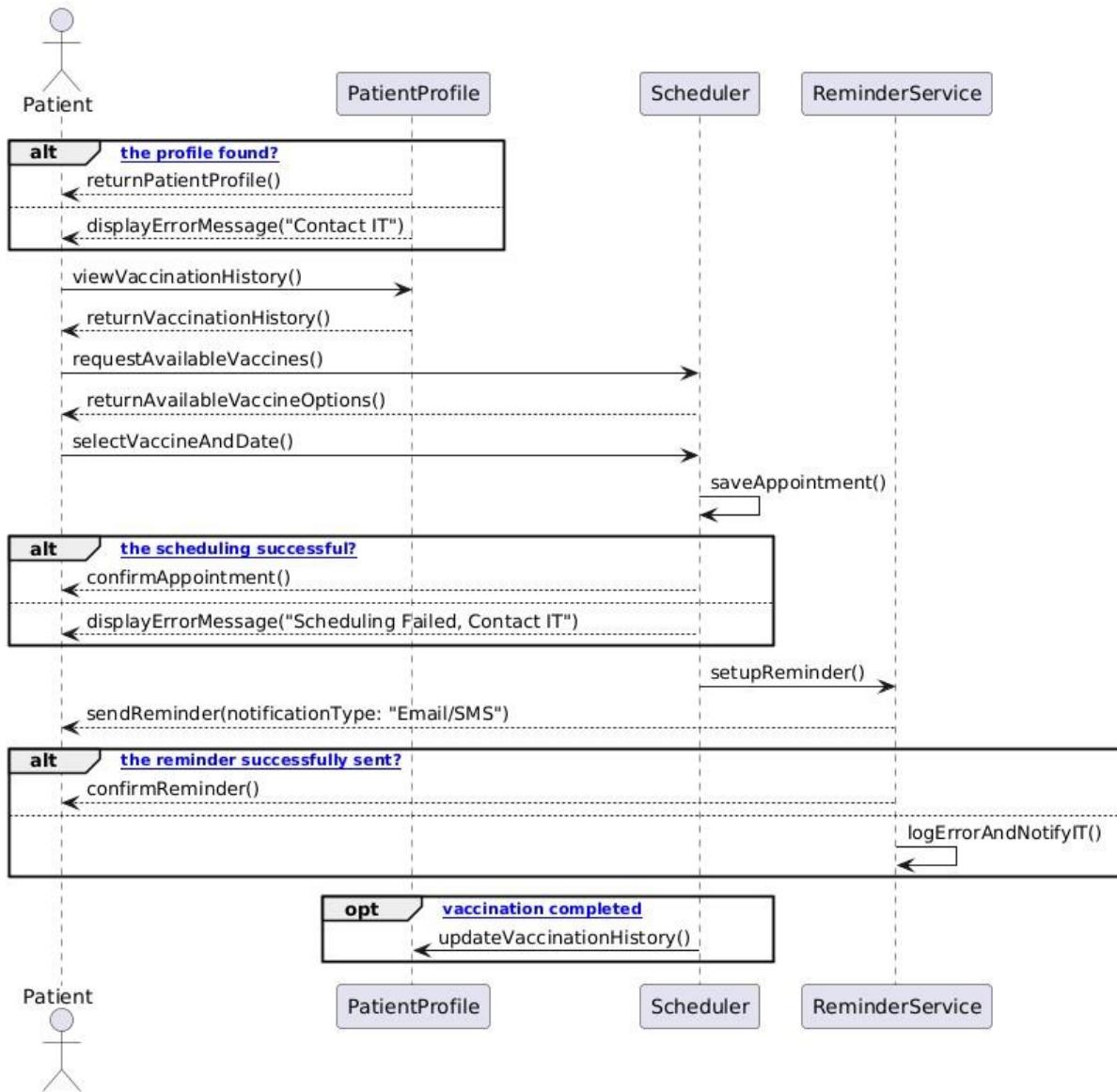


22 Library and Literature Search (Arlin Bashllari)

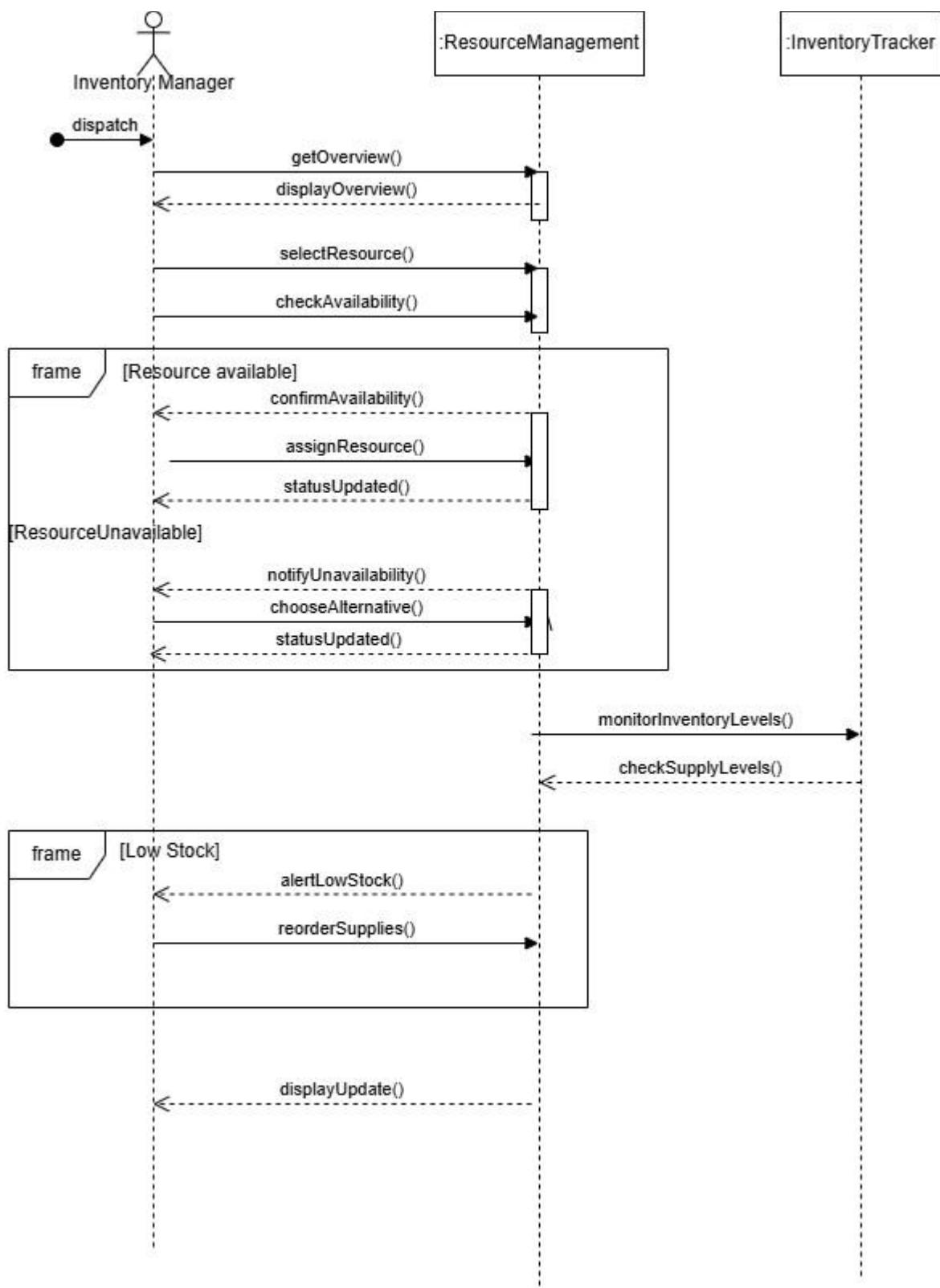


23 Library and Literature Management (Eglis Braho)

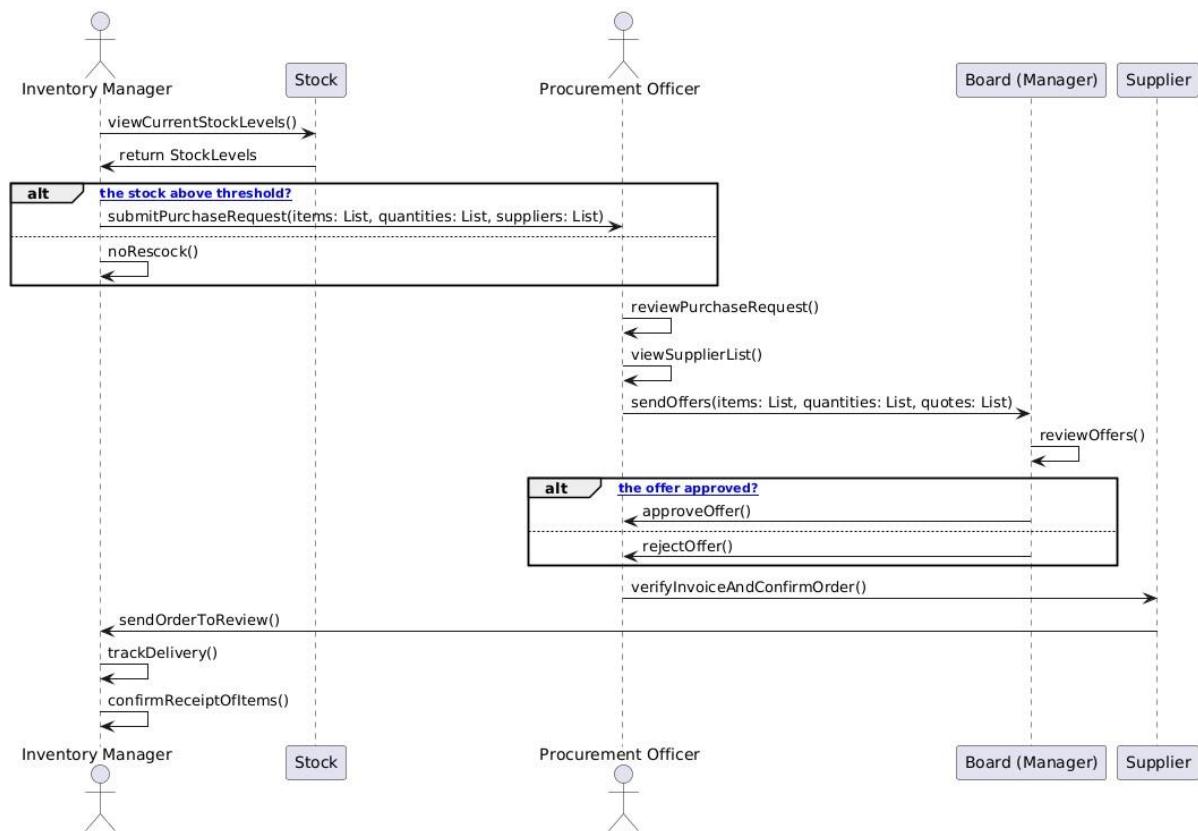




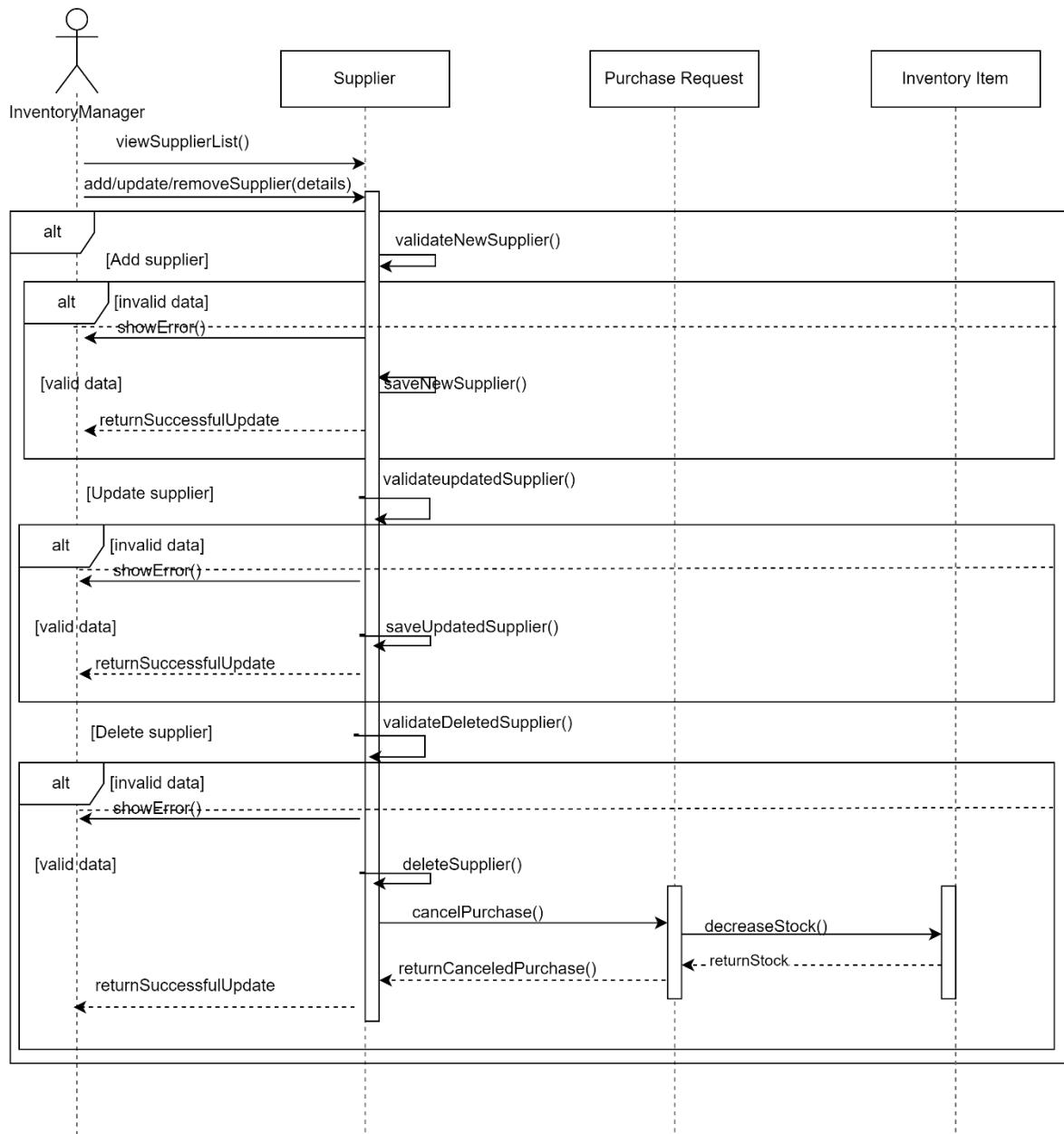
25 Resource Allocation (Eglis Braho)



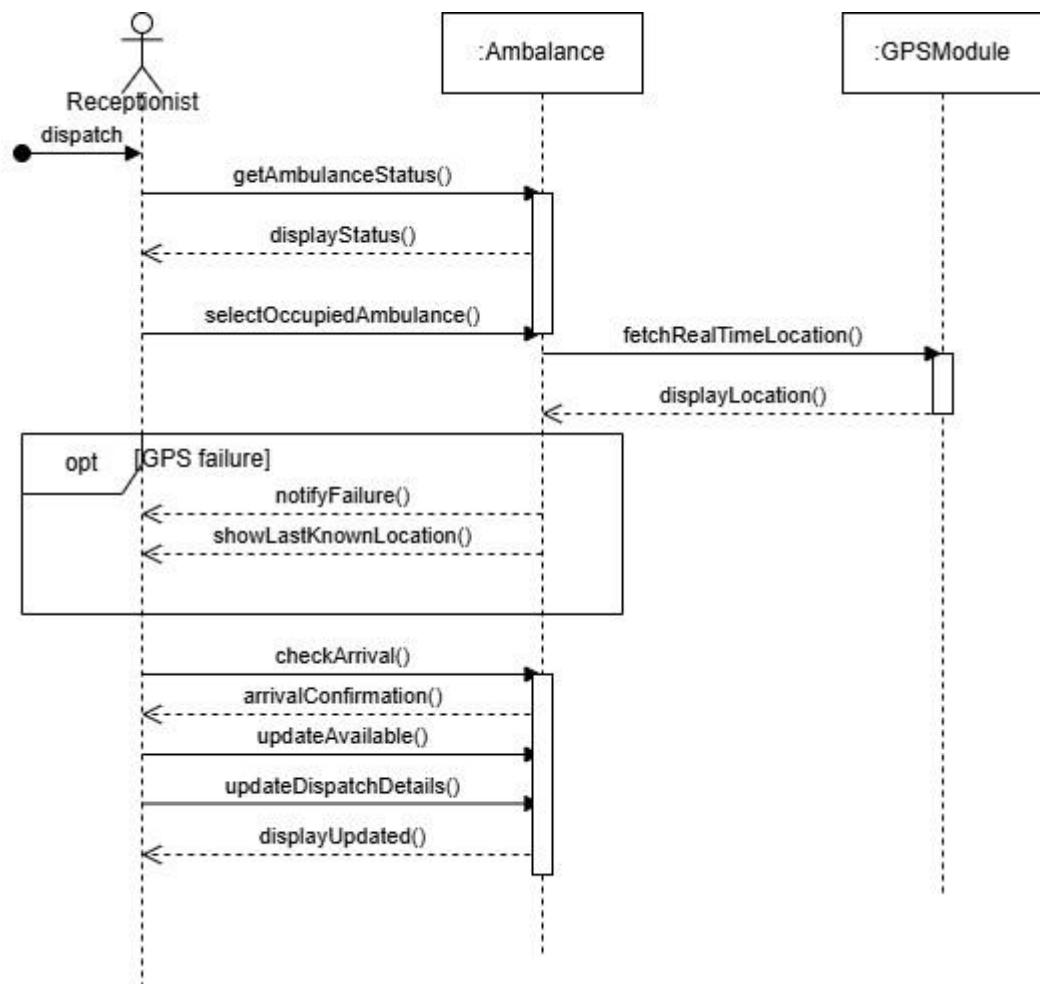
26 Inventory Item & Procurement Management (Arjan Muka)



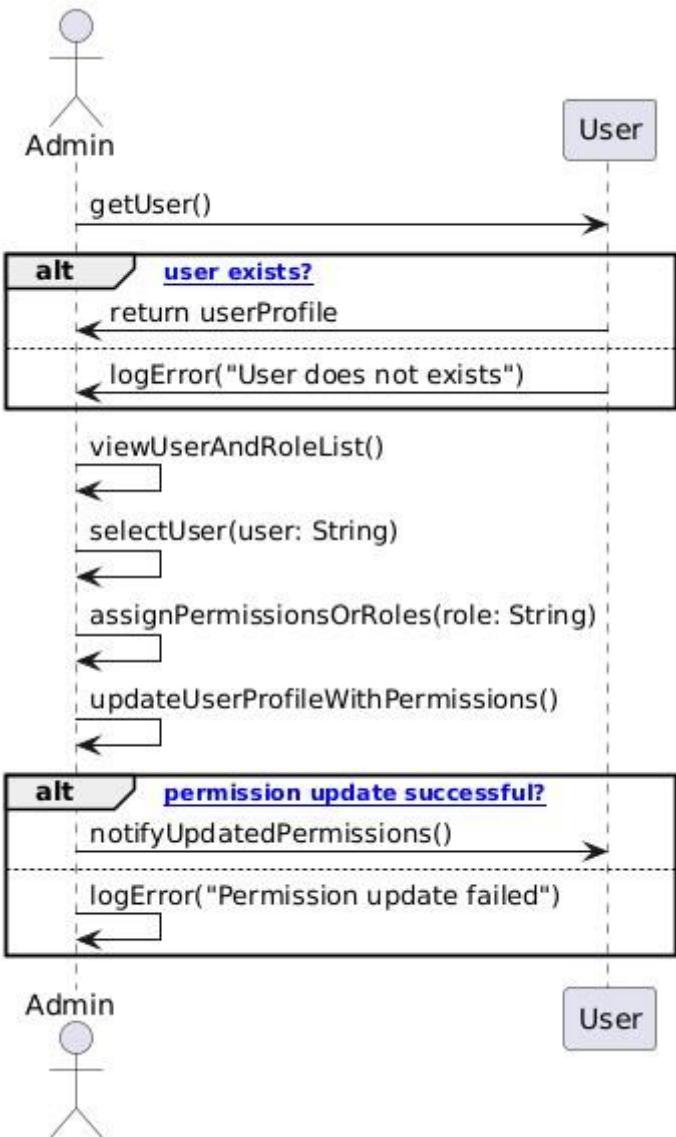
27 Supplier Management (Marin Tartaraj)



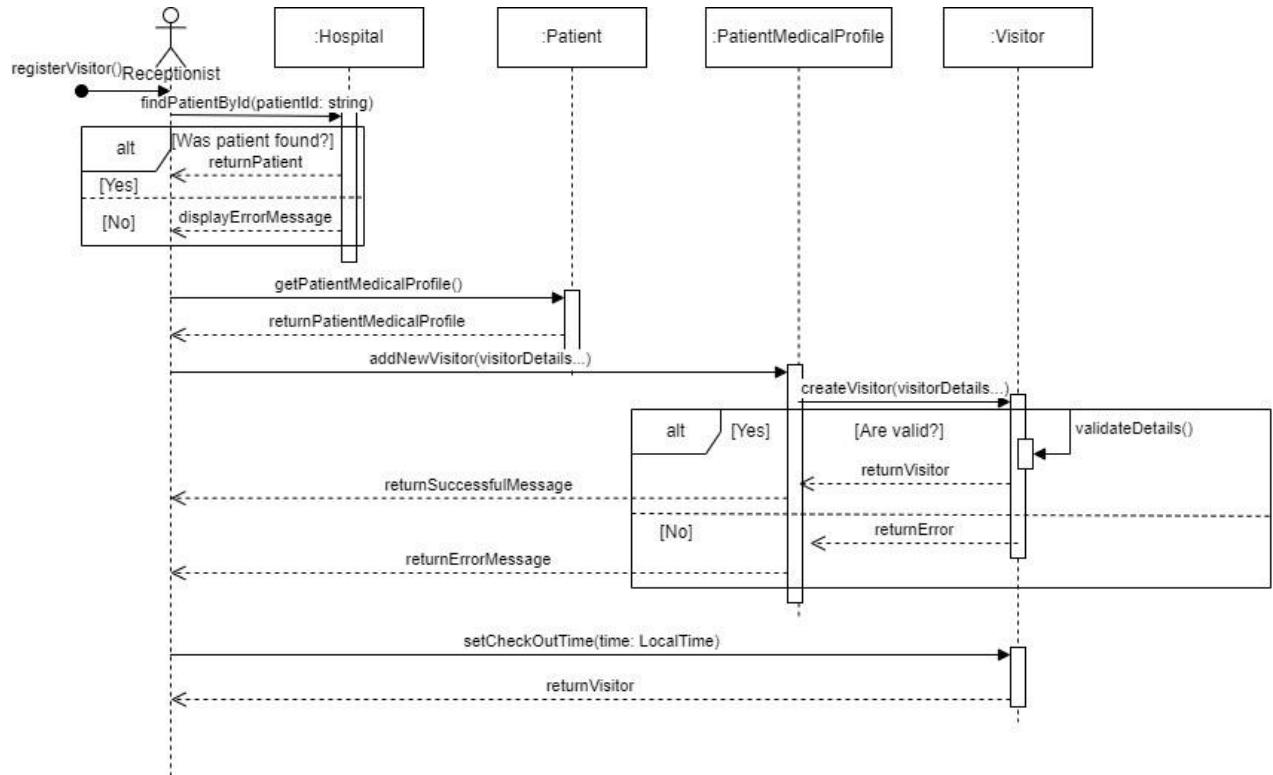
SD-28 Ambulance Management (Eglis Braho)



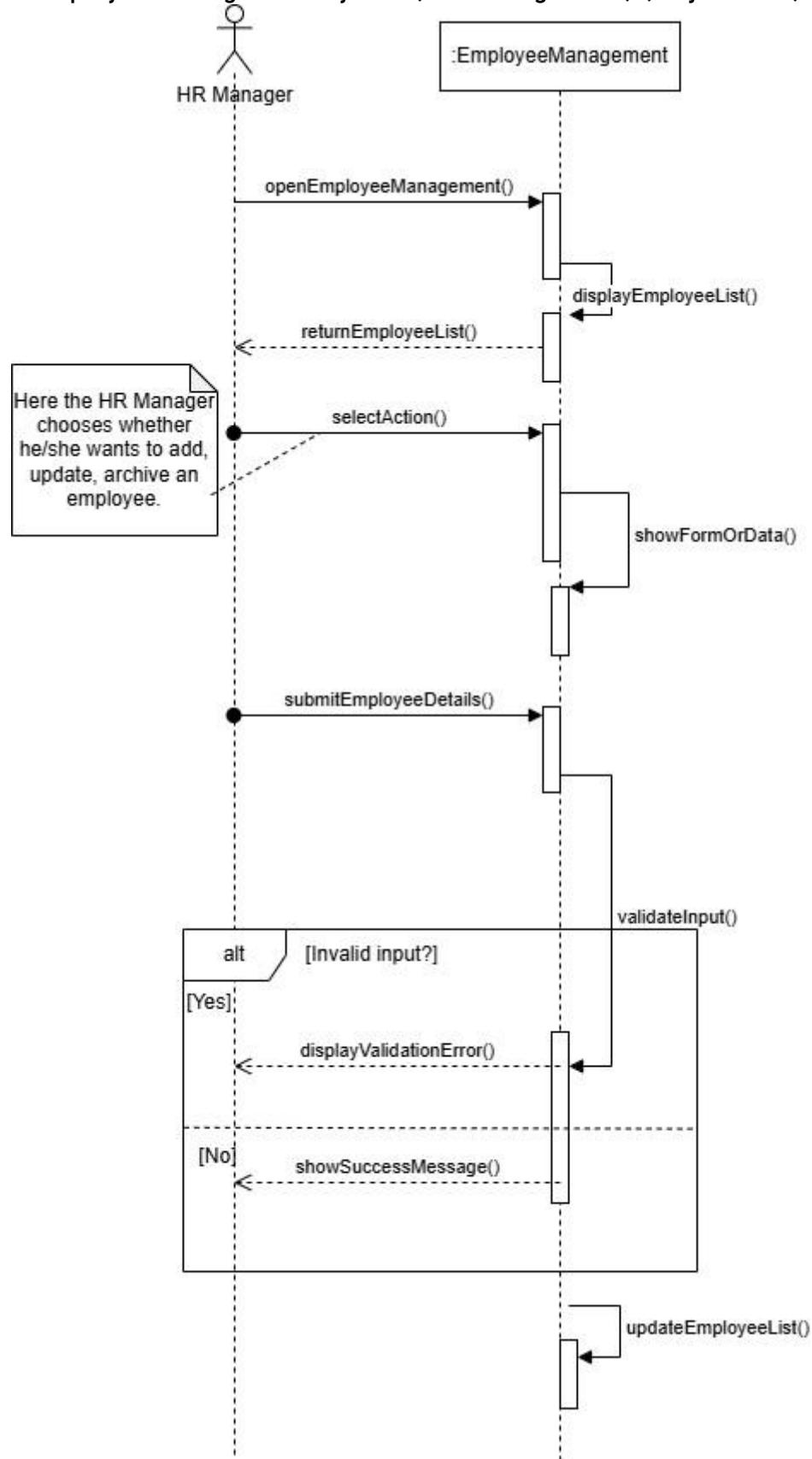
SD-29 Permission Granting (Arjan Muka)



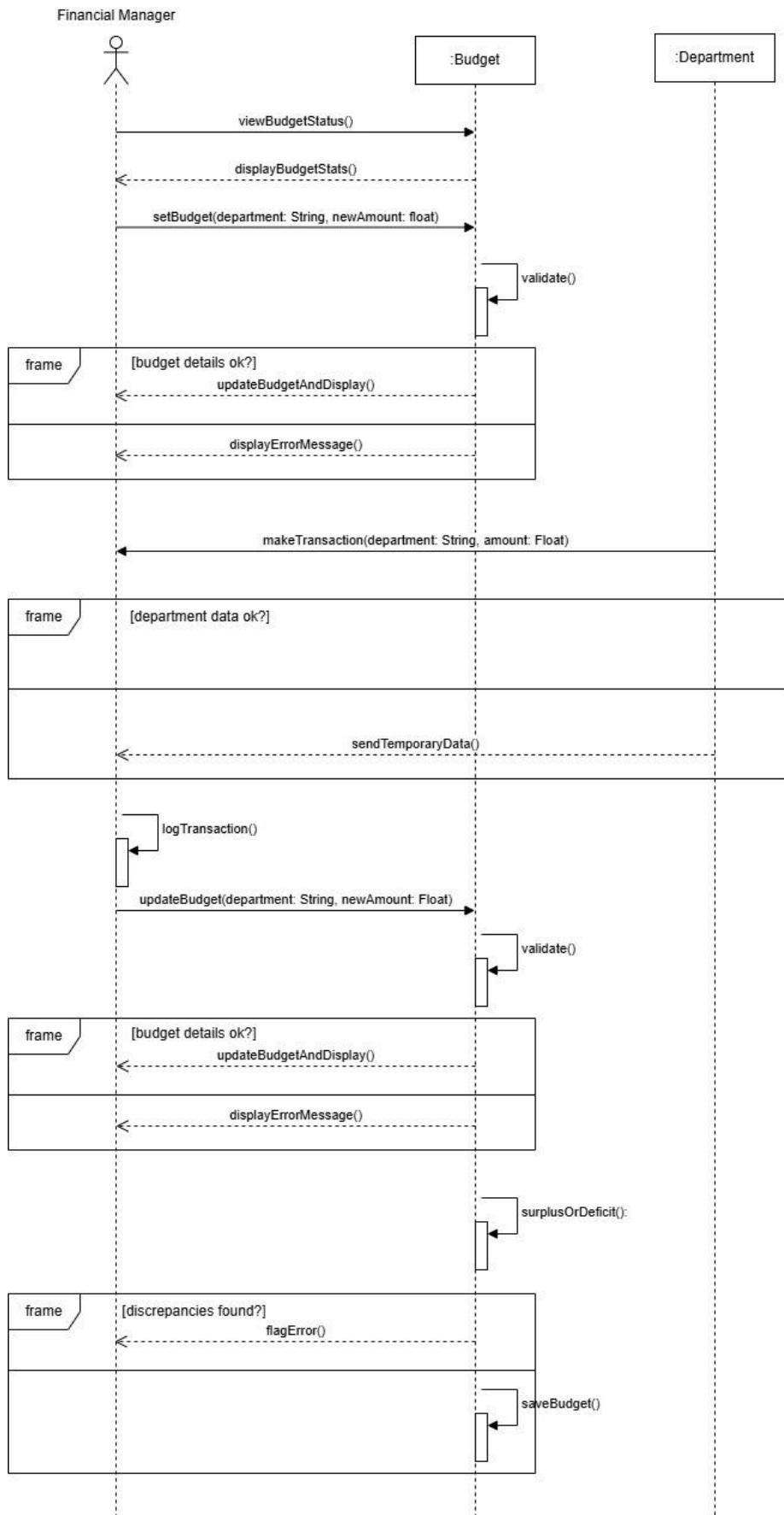
SD-30 Visitor Management (Shpetim Shabanaj)



SD-31 Employee Management System(HR management) (Artjol Zaimi)

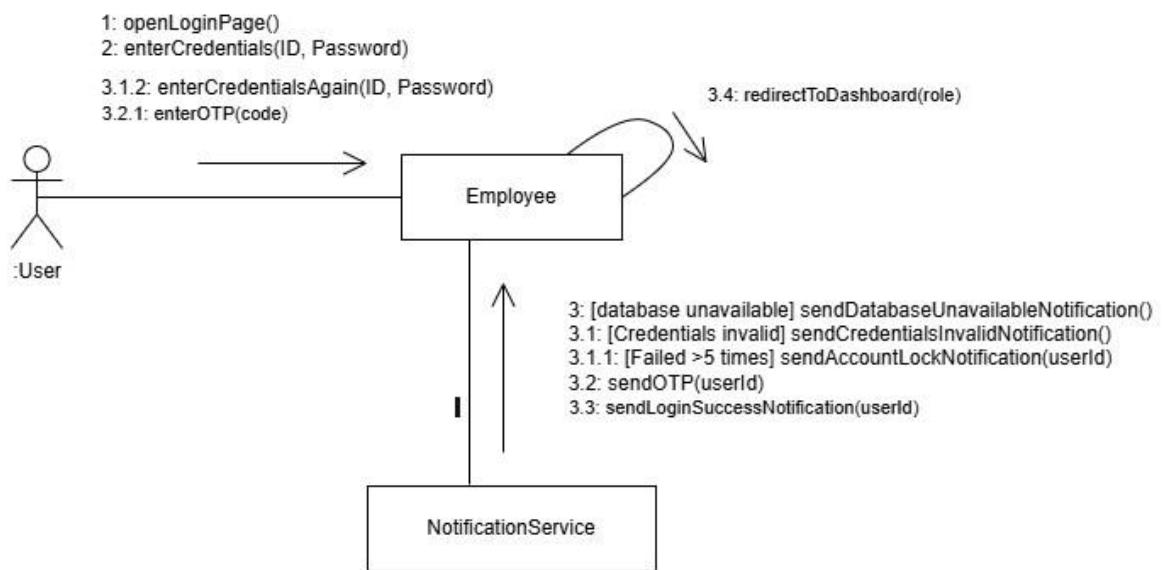


SD Name SD-32 Financial Management (Nikola Rigo)

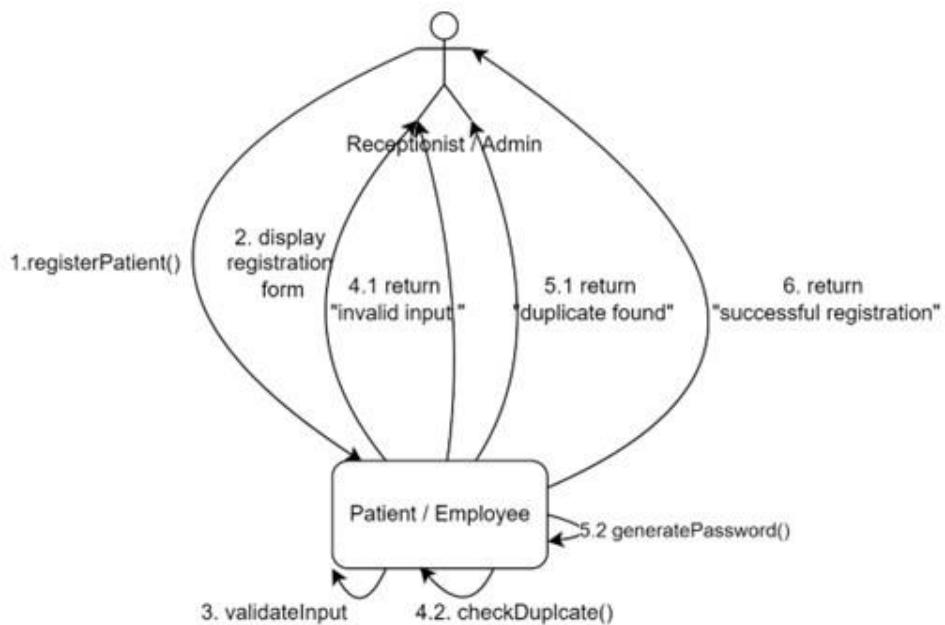


Collaboration/Communication Diagram

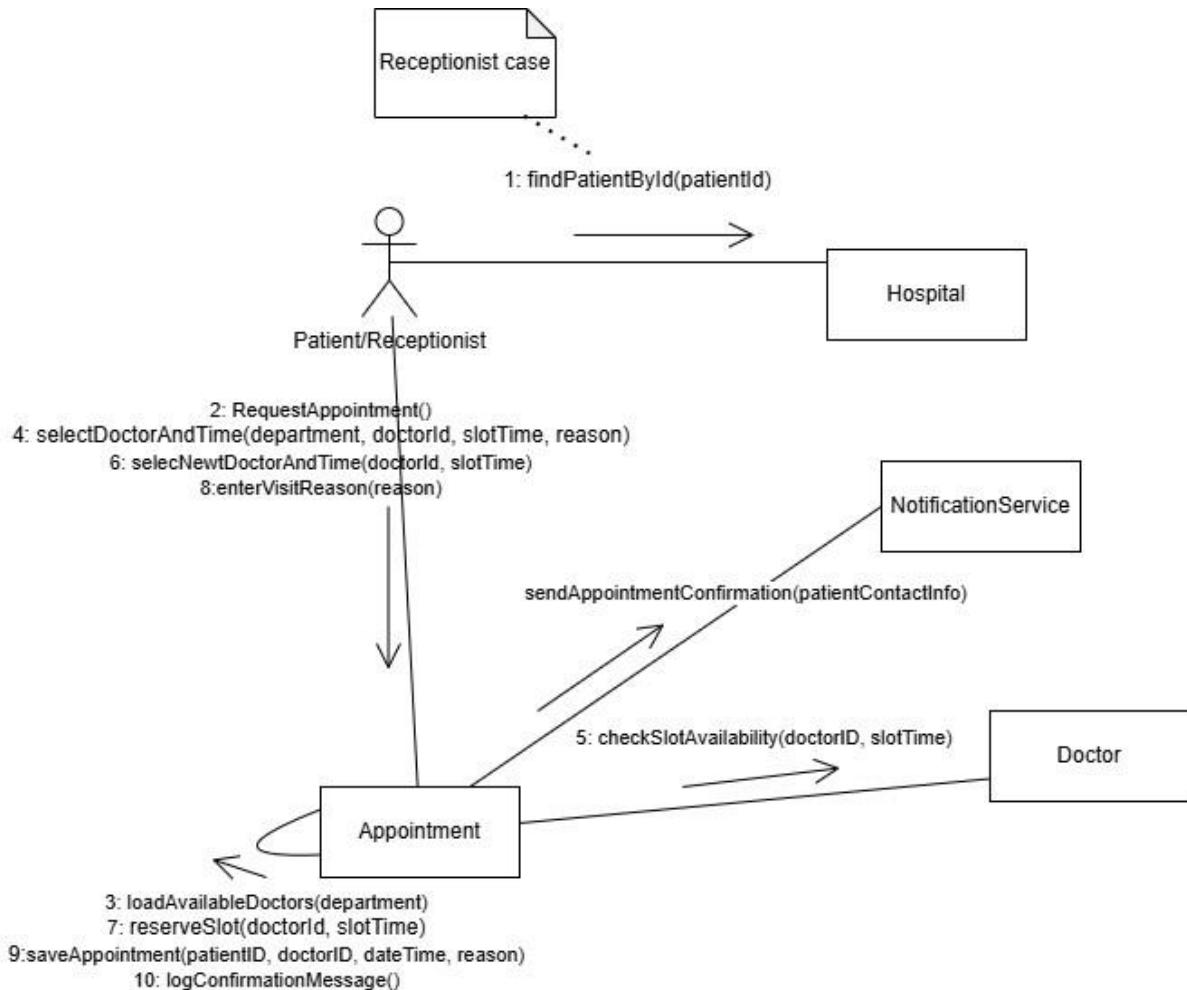
CD-01 User Login (Artjol Zaimi)



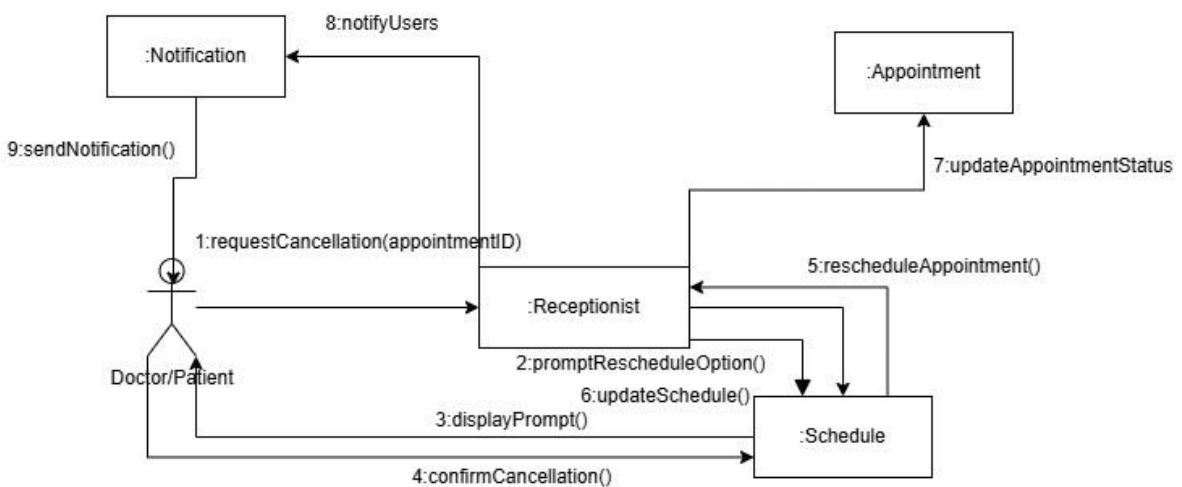
CD-02 Registration (Marin Tartara)



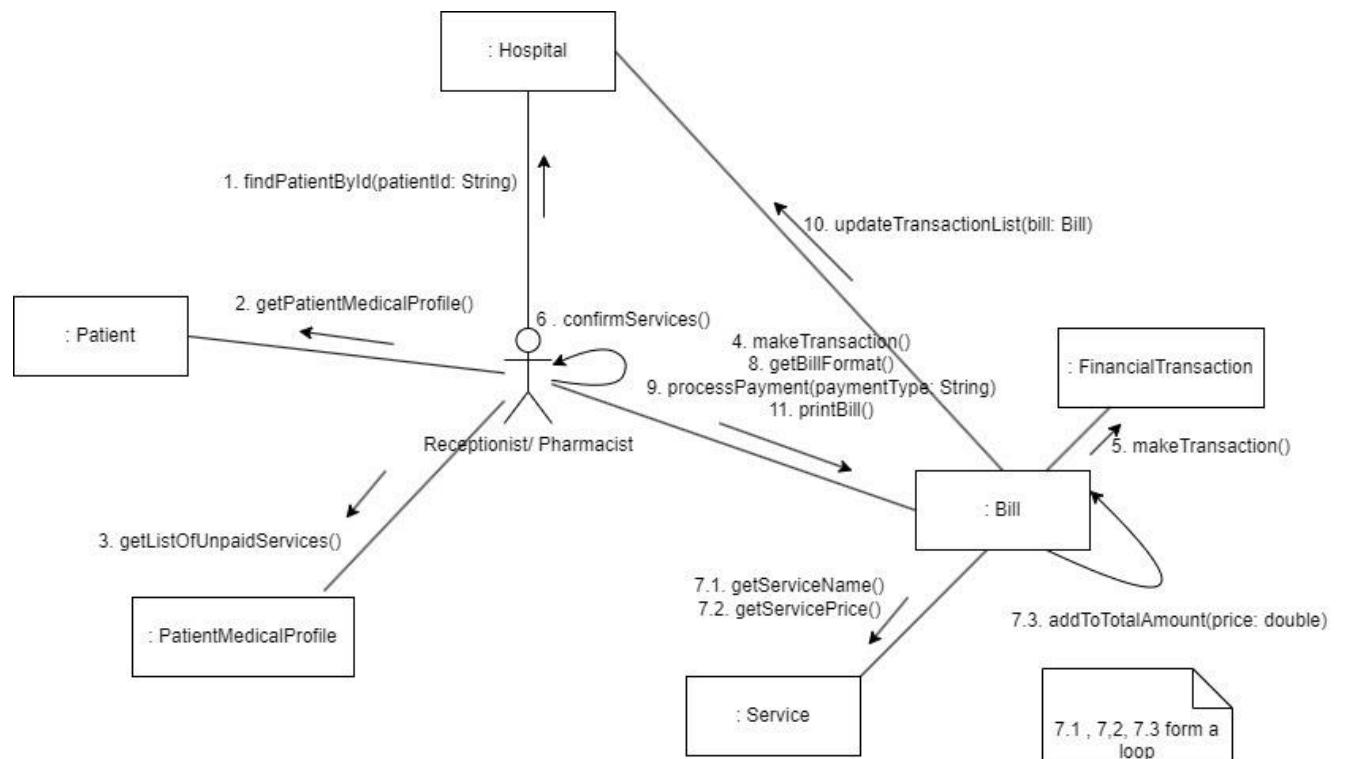
CD-03 Appointment Scheduling (Artjol Zaimi)



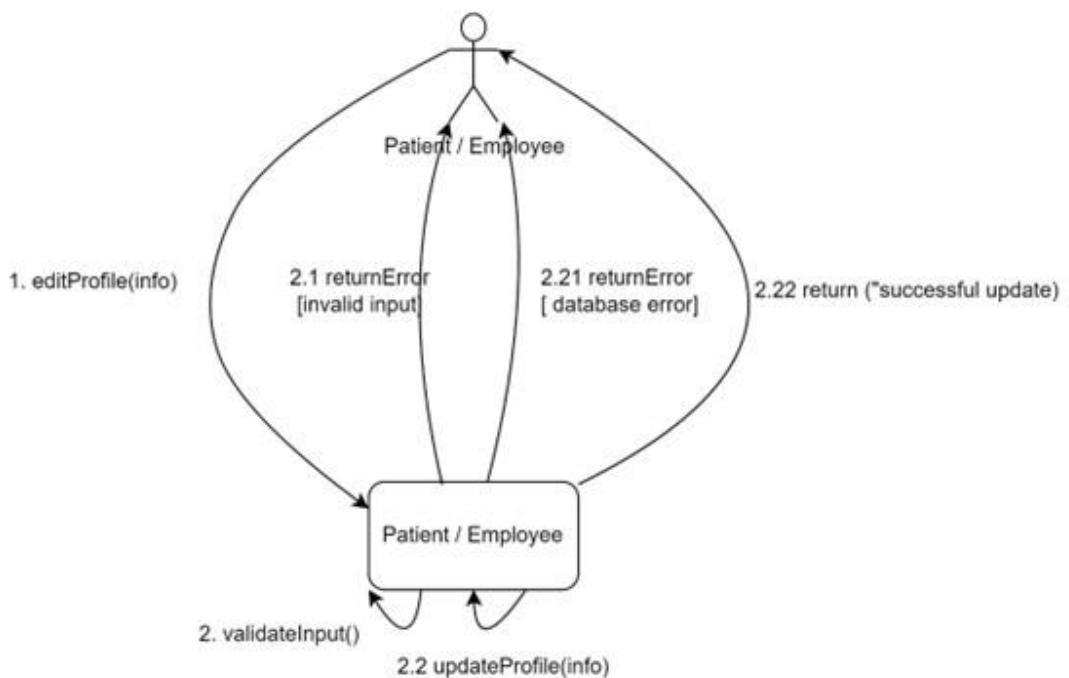
CD-04 Appointment Cancellation (Arlin Bashllari)



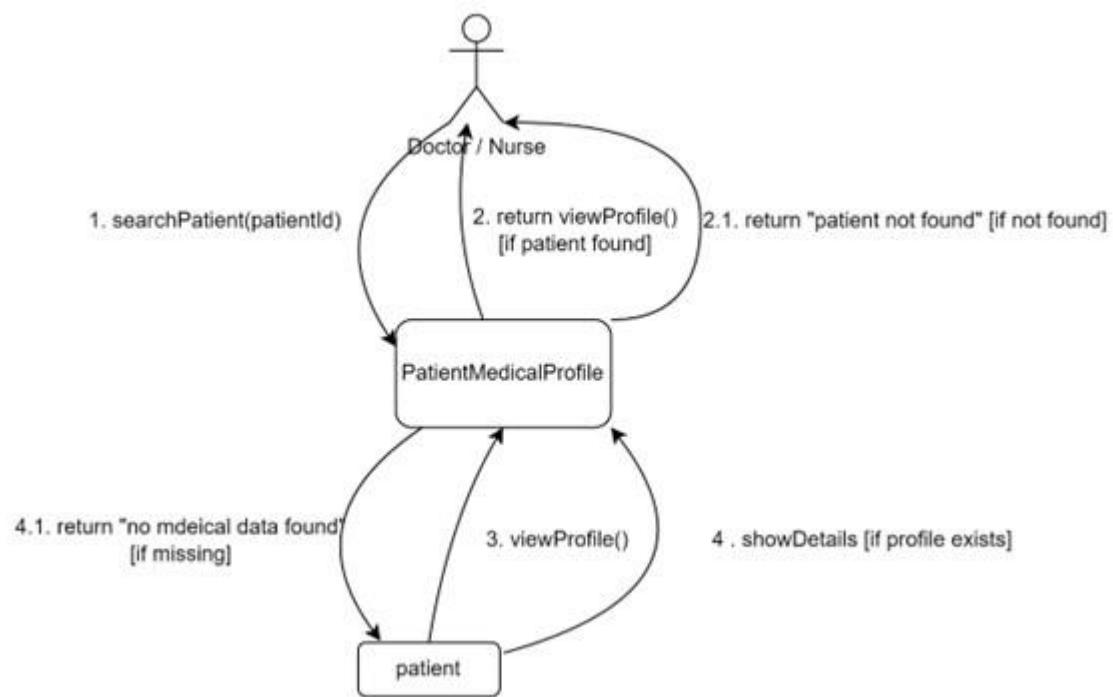
CD-05 Billing Component & Payment Processing (Shpetim Shabanaj)



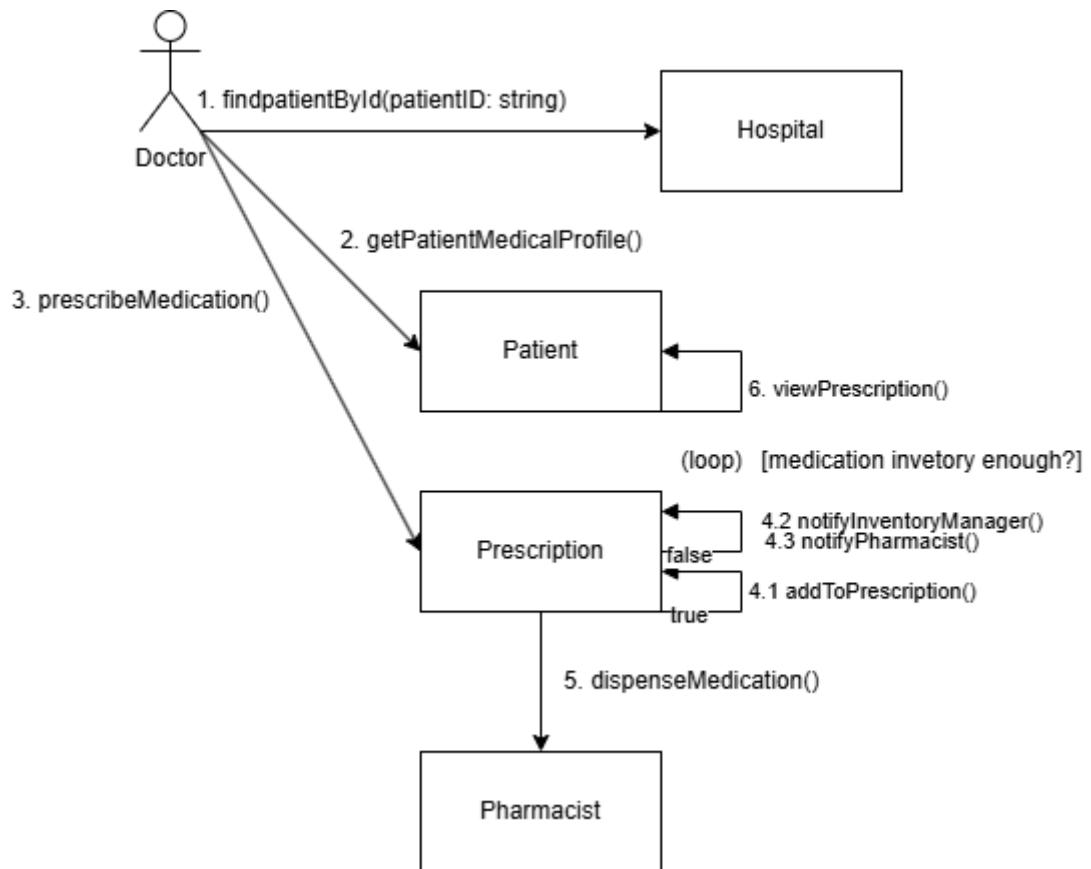
CD-06 Profile Management (Marin Tartaraj)



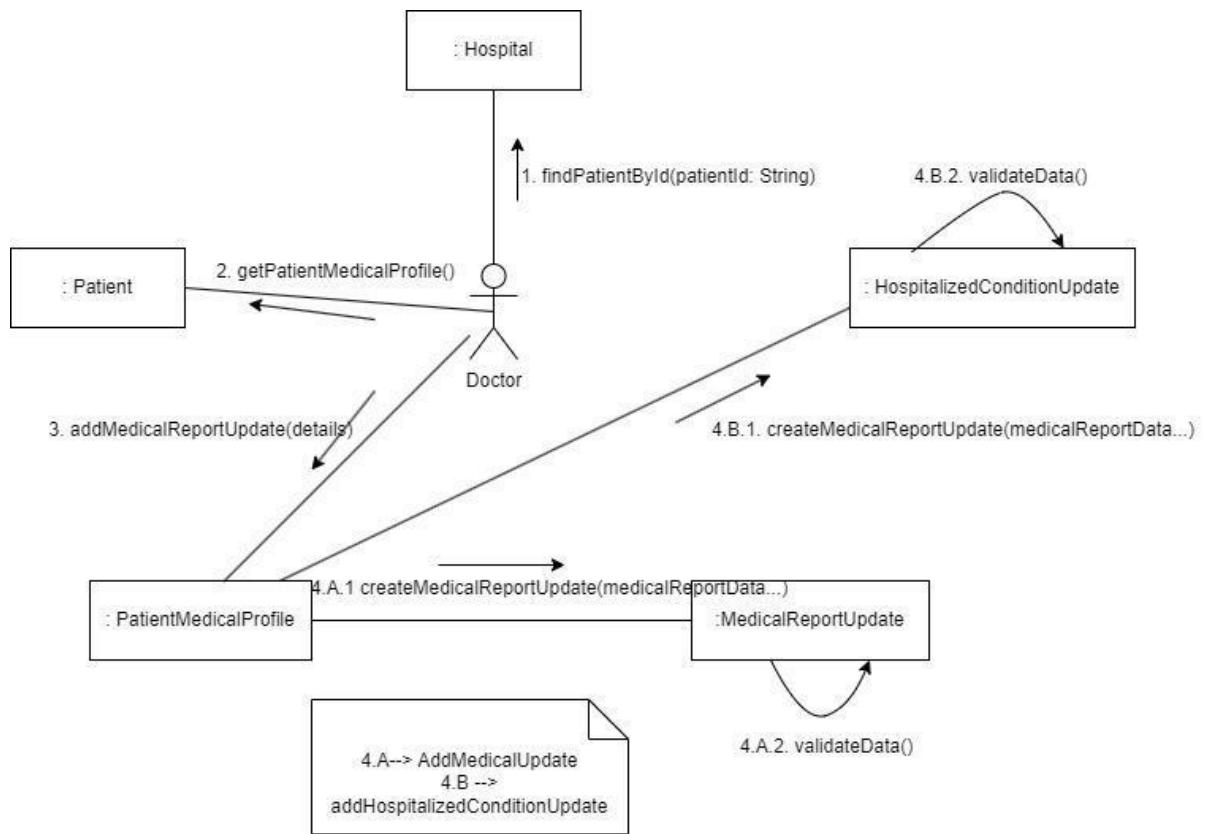
CD-07 Patient Medical Profile()



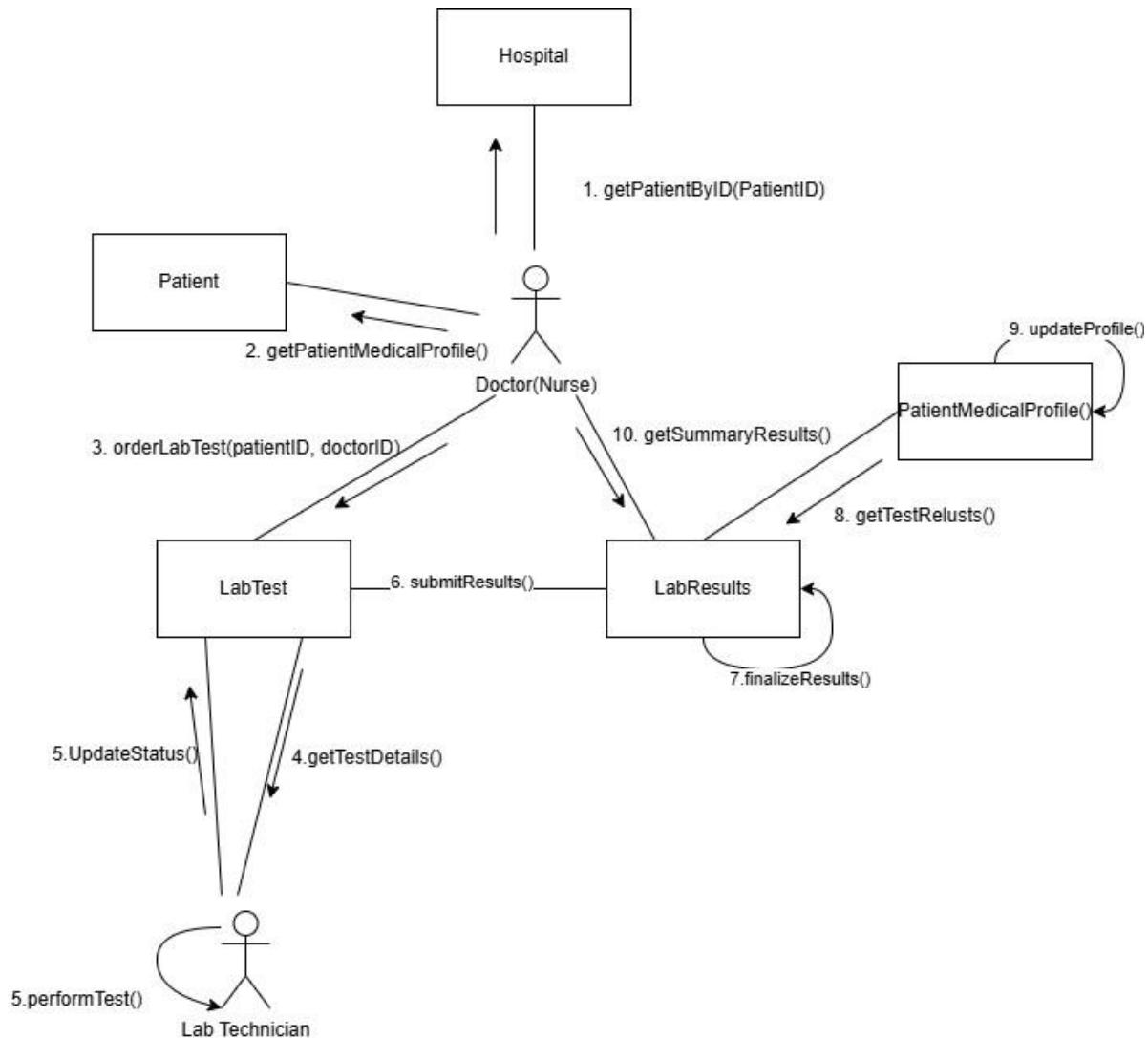
CD-08 Medication Prescription & Viewing (Nikola Rigo)



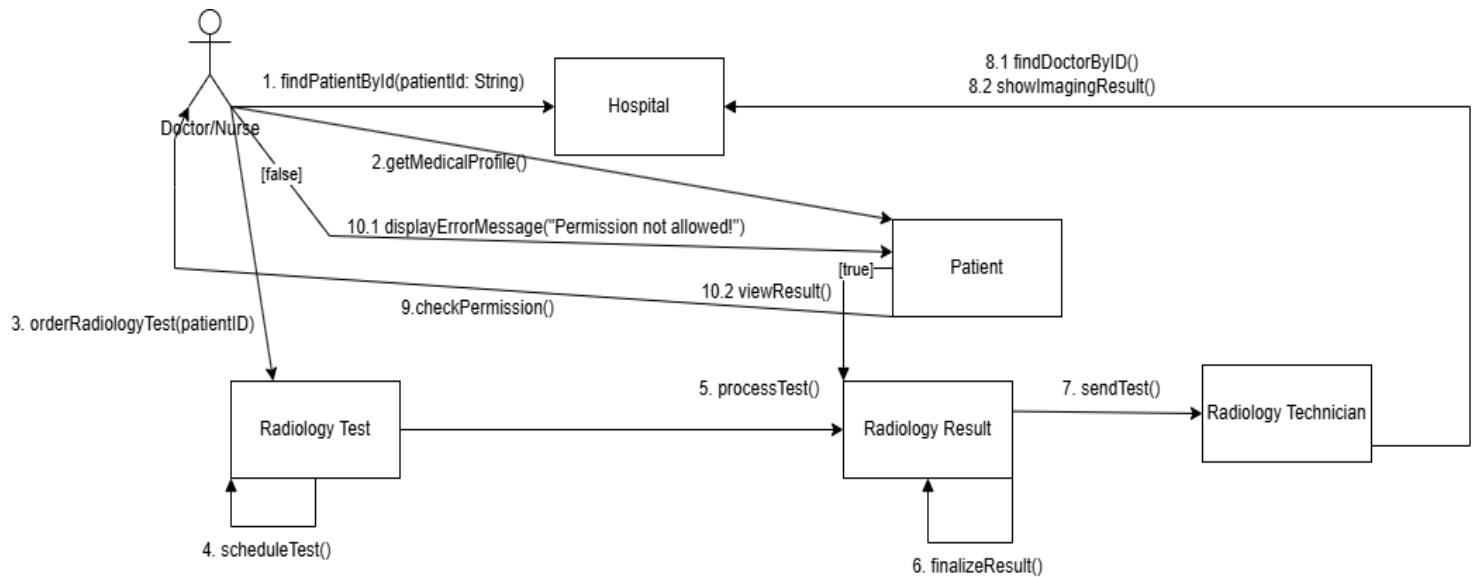
CD-09 Electronic Health Records Update (Shpetim Shabanaj)



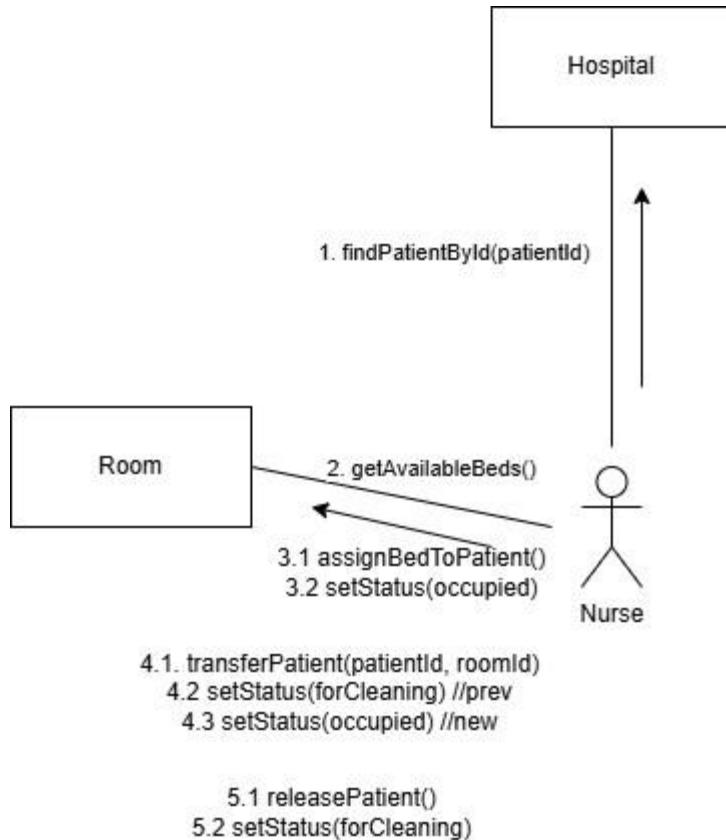
CD-10 Lab Test Ordering and Result Upload (Arjan Muka)



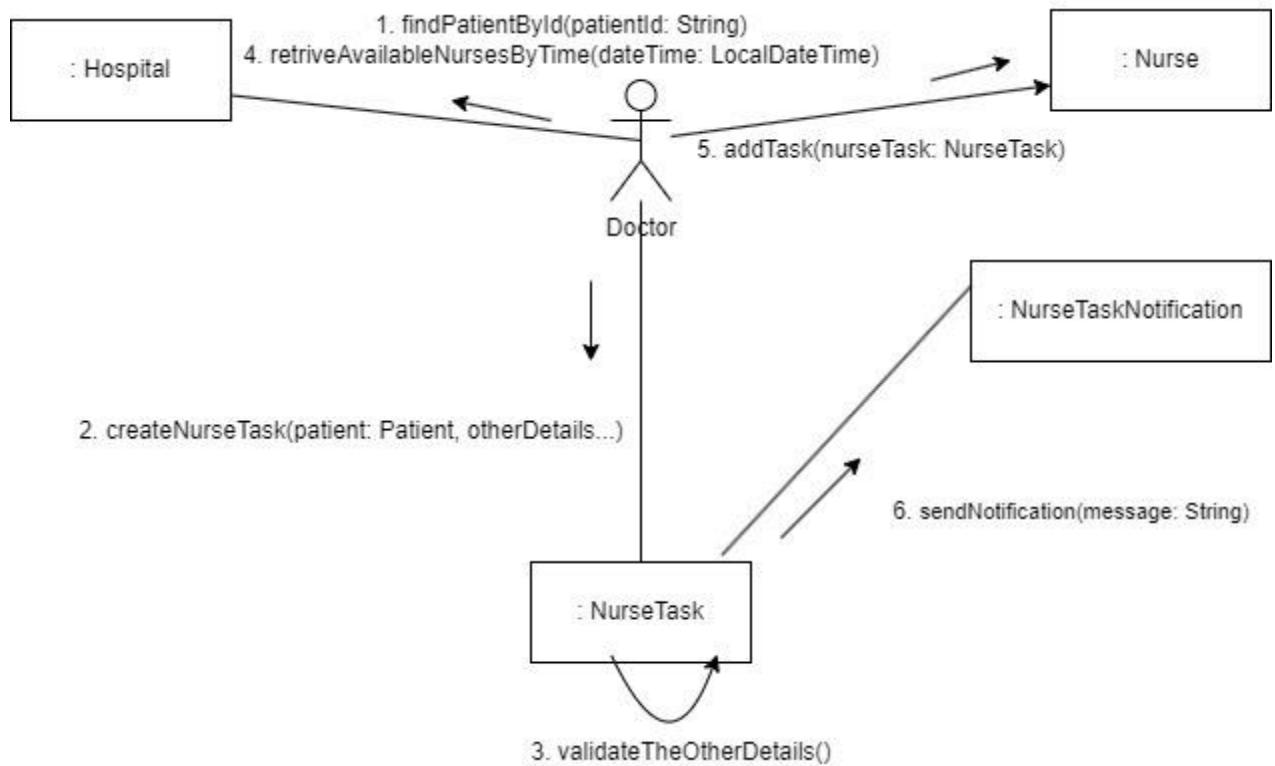
CD-11 Radiology & Imaging Ordering and Result Upload(Nikola Rigo)



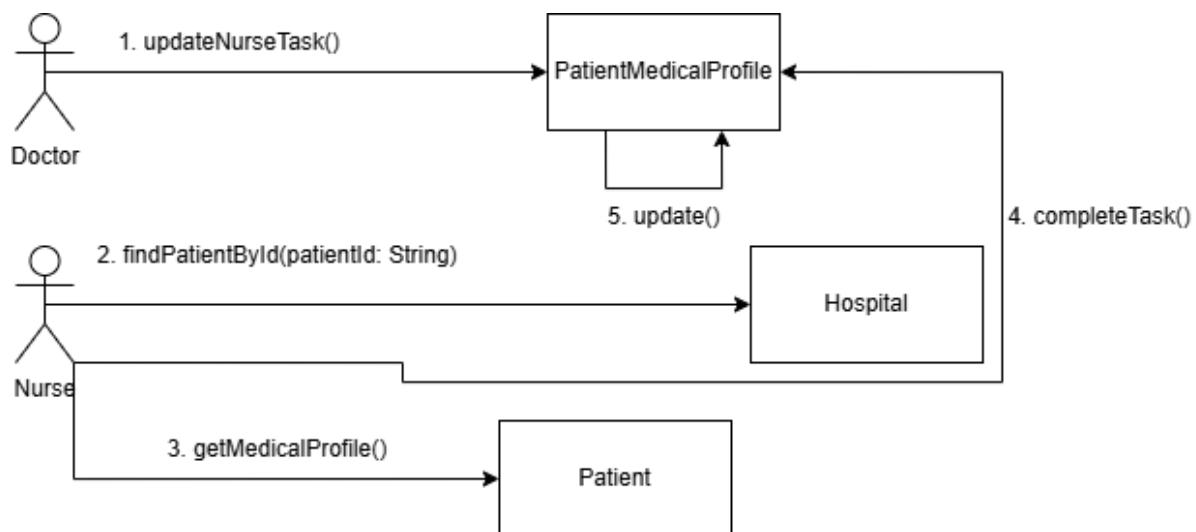
CD-12 Inpatient & Bed Management (Arjan Muka)



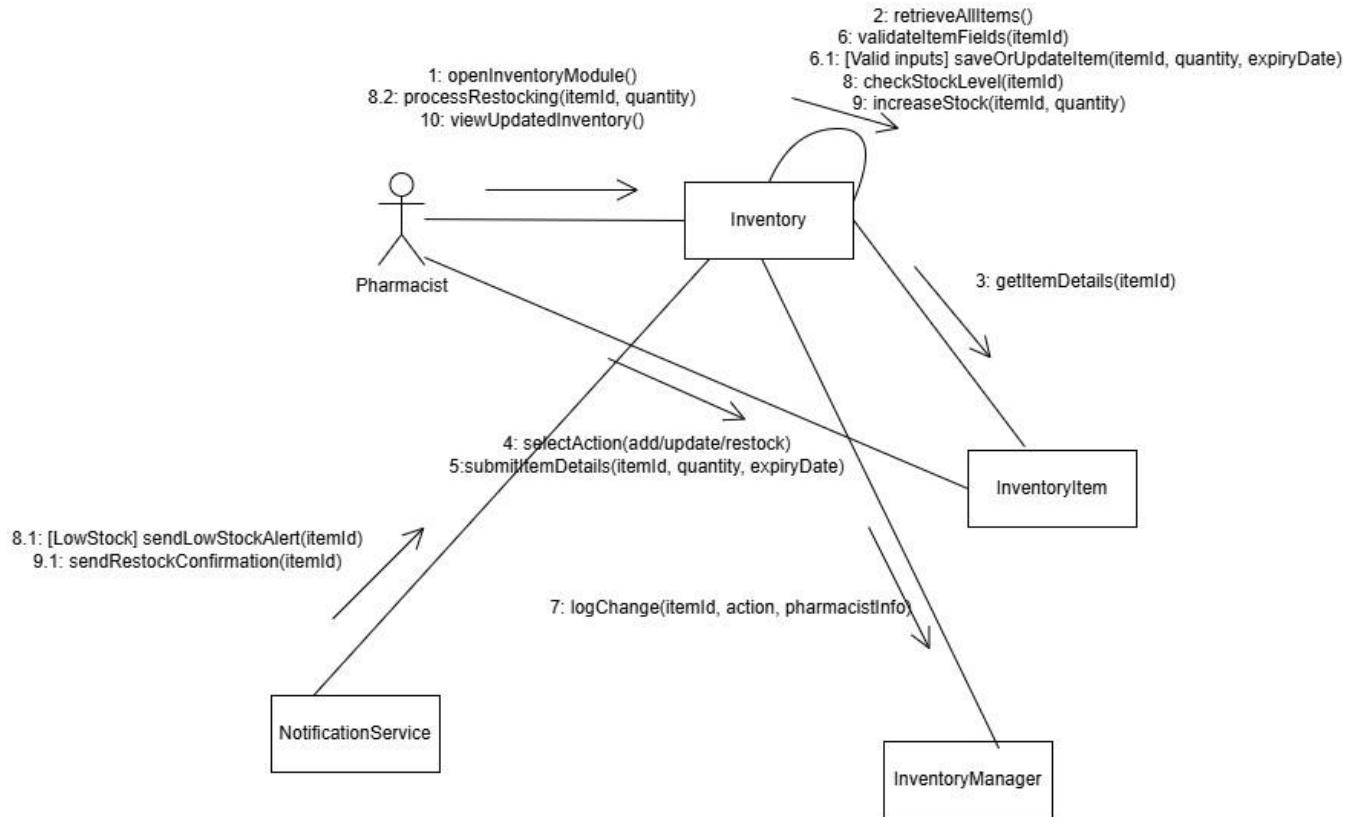
CD-13 Nurse Task Assignment (Shpetim Shabanaj)



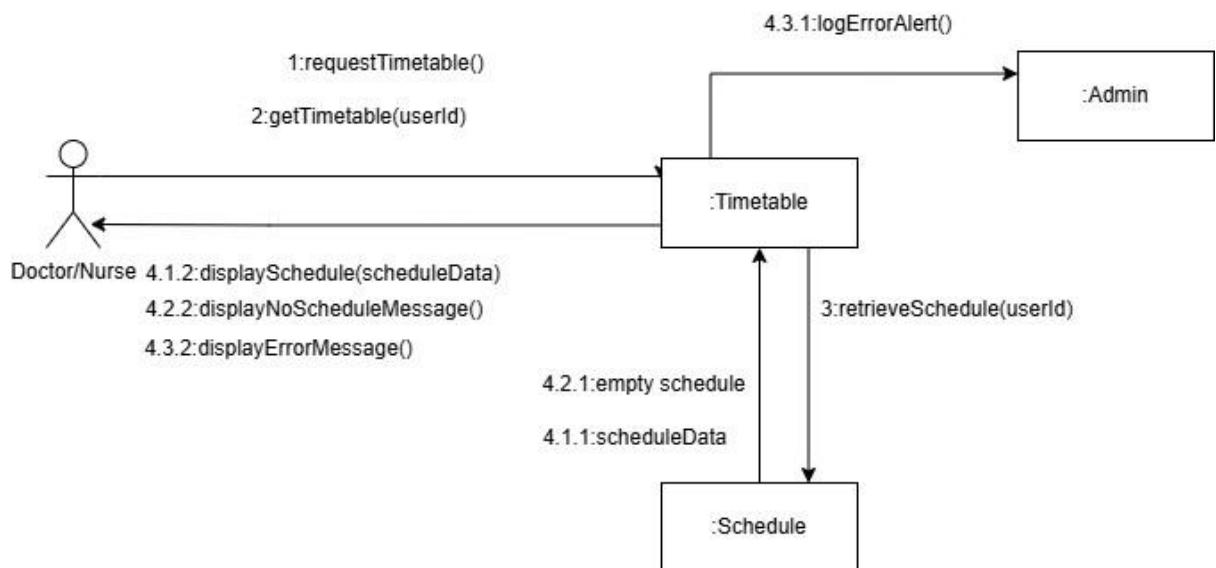
CD-14 Medication Processing for patients (Nikola Rigo)



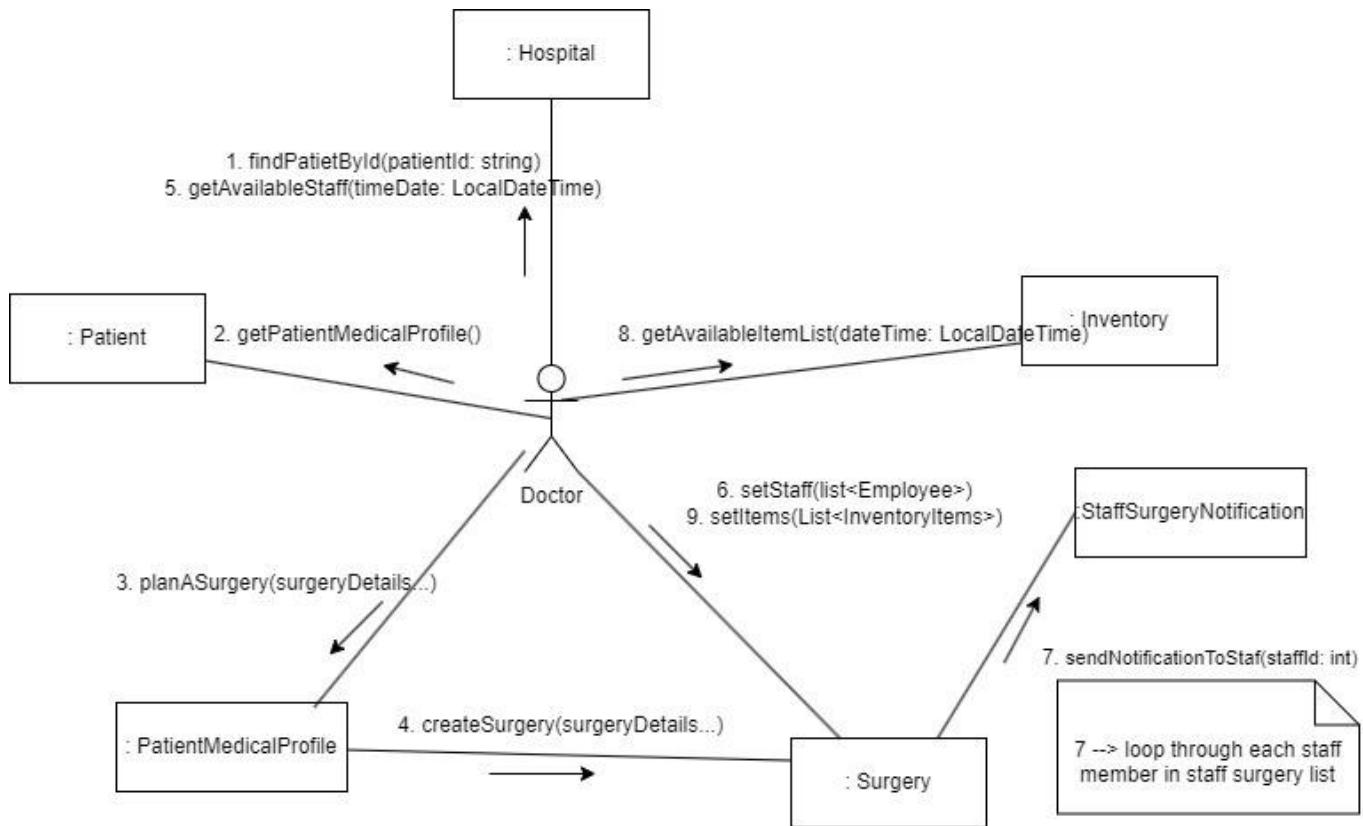
CD-15 Pharmacy & Stock Management (Artjol Zaimi)



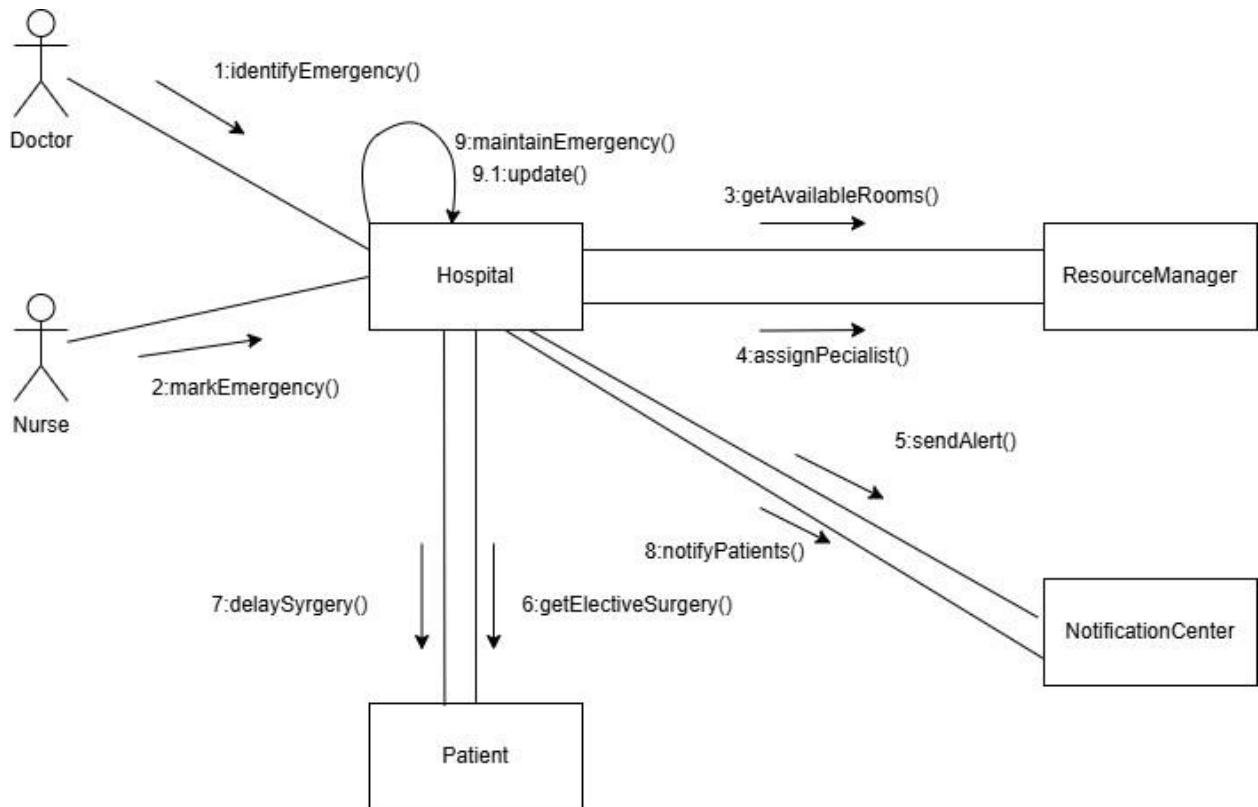
CD-16 Medical Staff Timetable (Arlin Bashllari)



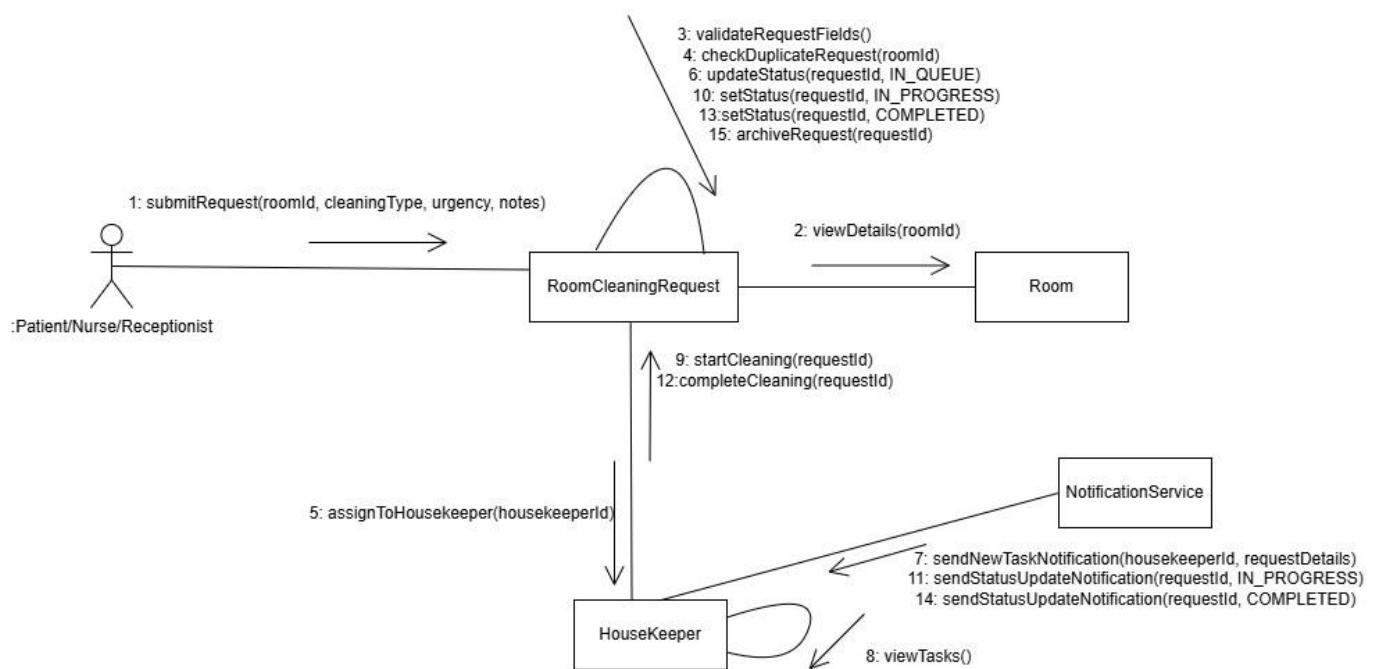
CD-17 Surgery Planning (Shpetim Shabana)



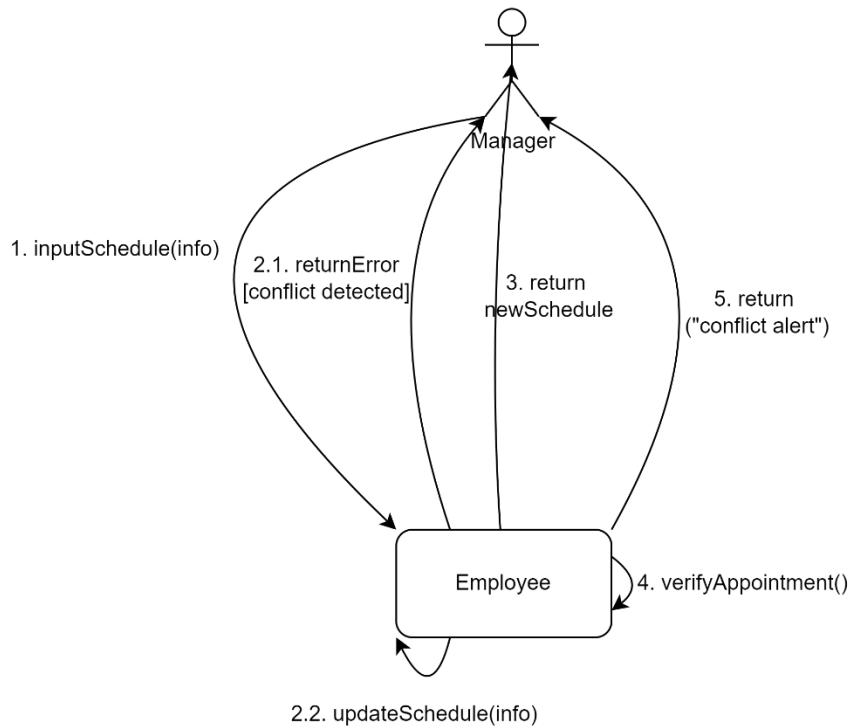
CD-18 Emergency Handling and Alerts (Eglis Braho)



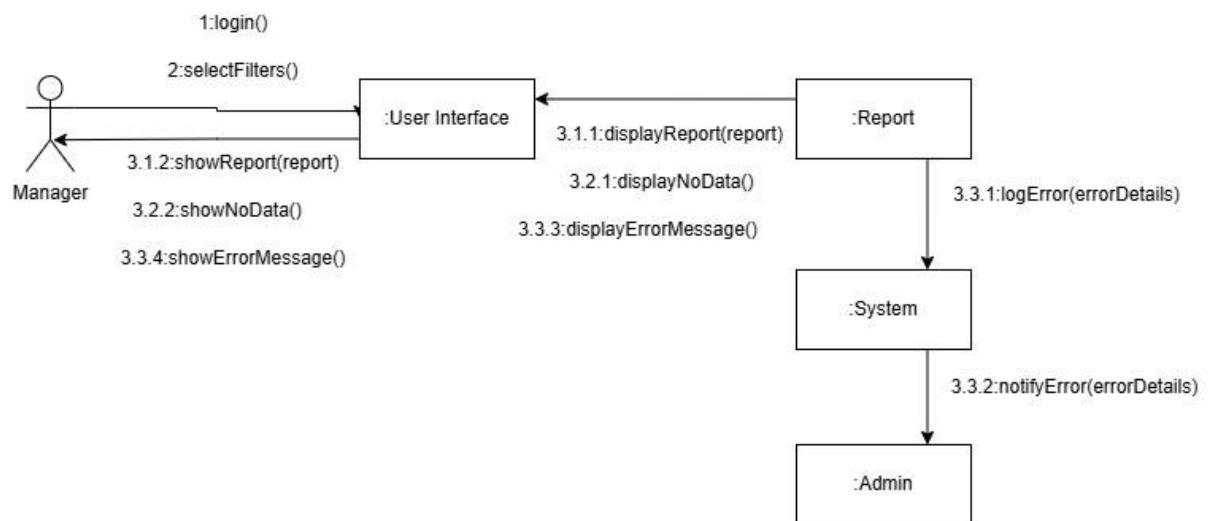
CD-19 Room Cleaning Management (Artjol Zaimi)



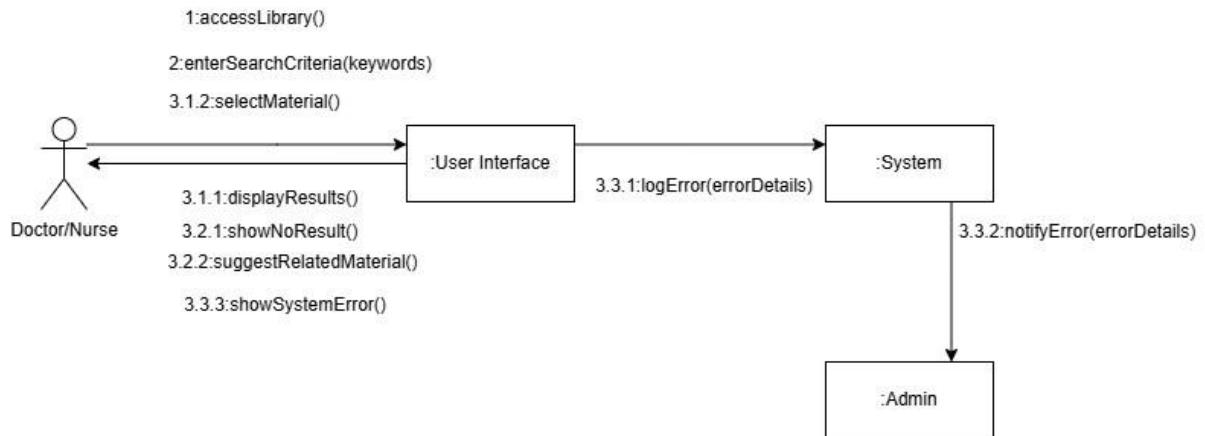
CD-20 Staff Scheduling (Marin Tartaraj)



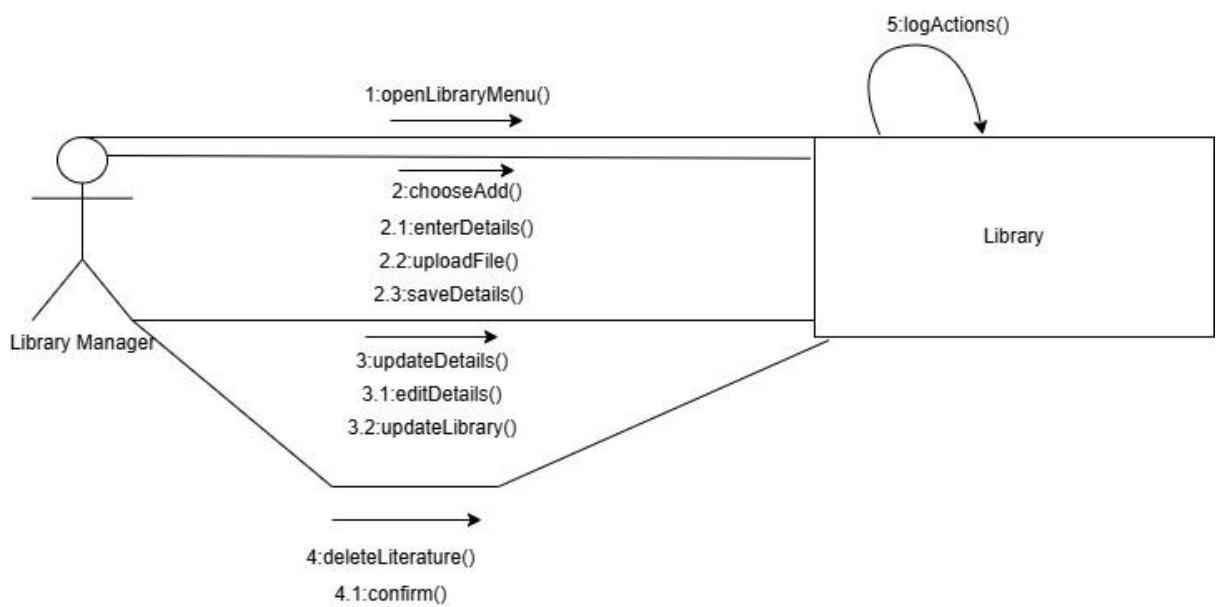
CD-21 Report Generation (Arlin Bashllari)



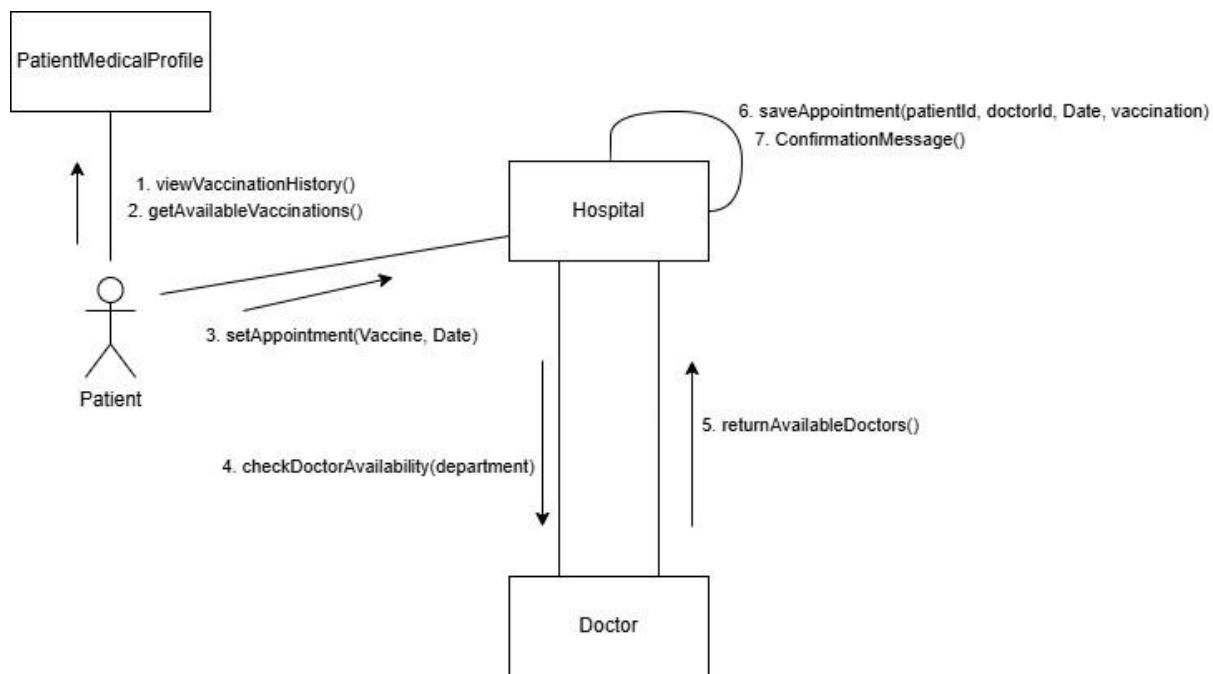
CD-22 Library and Literature Search (Arlin Bashllari)



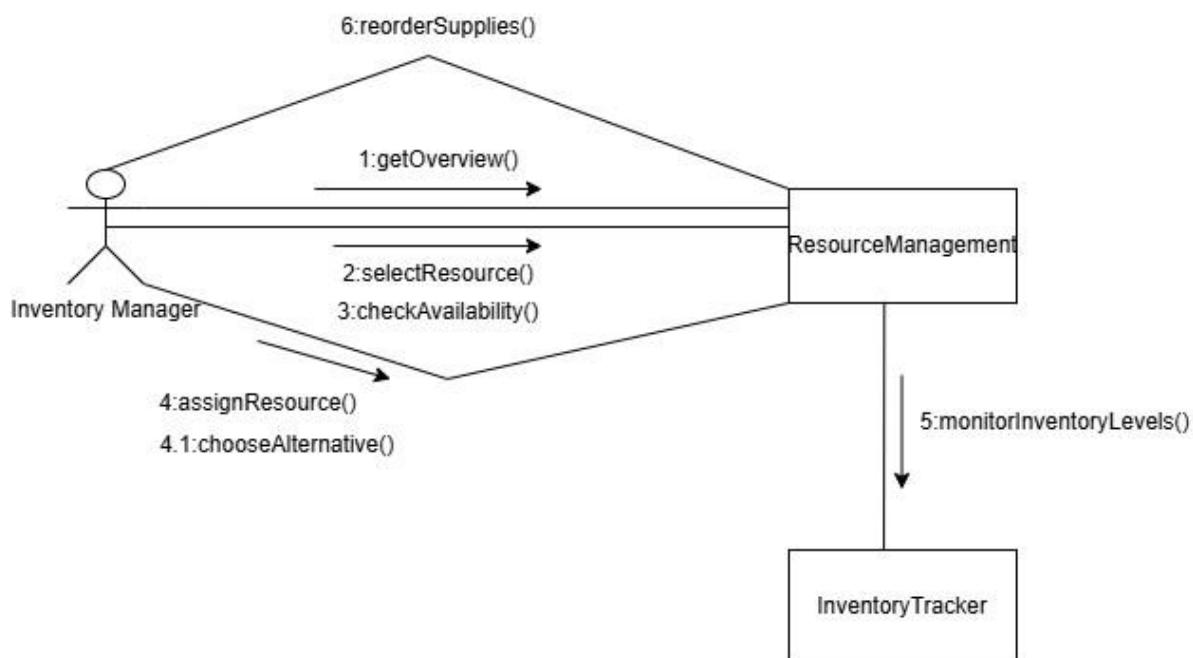
CD-23 Library and Literature Management (Eglis Braho)



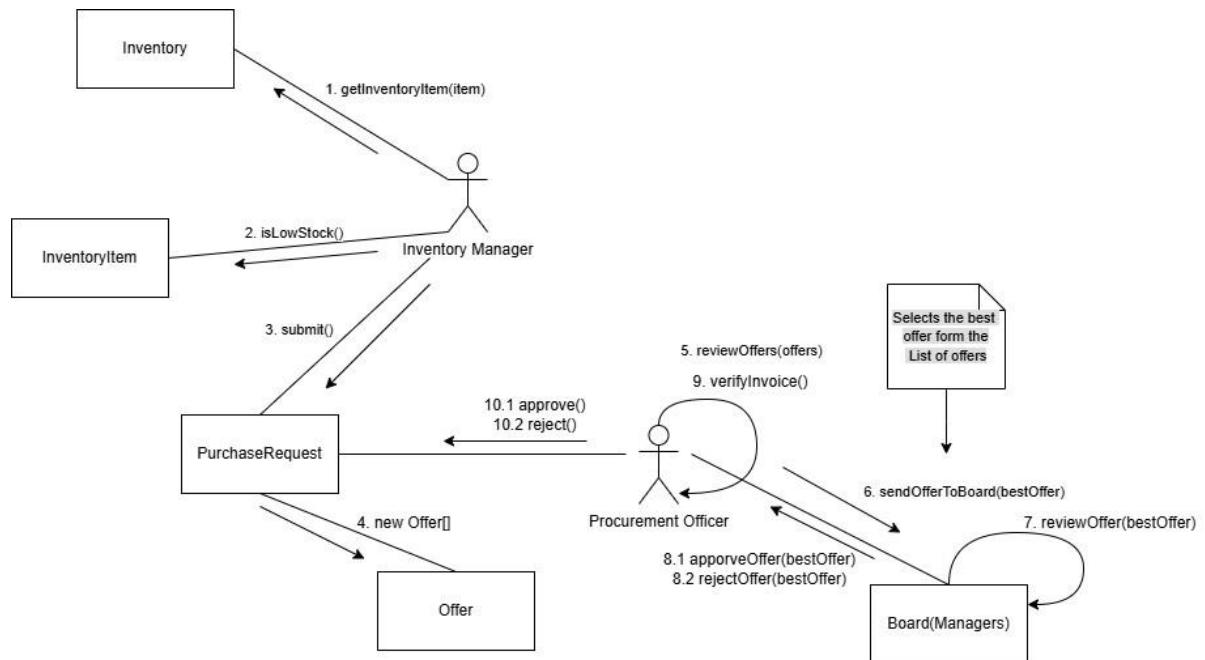
CD-24 Vaccination Management (Arjan Muka)



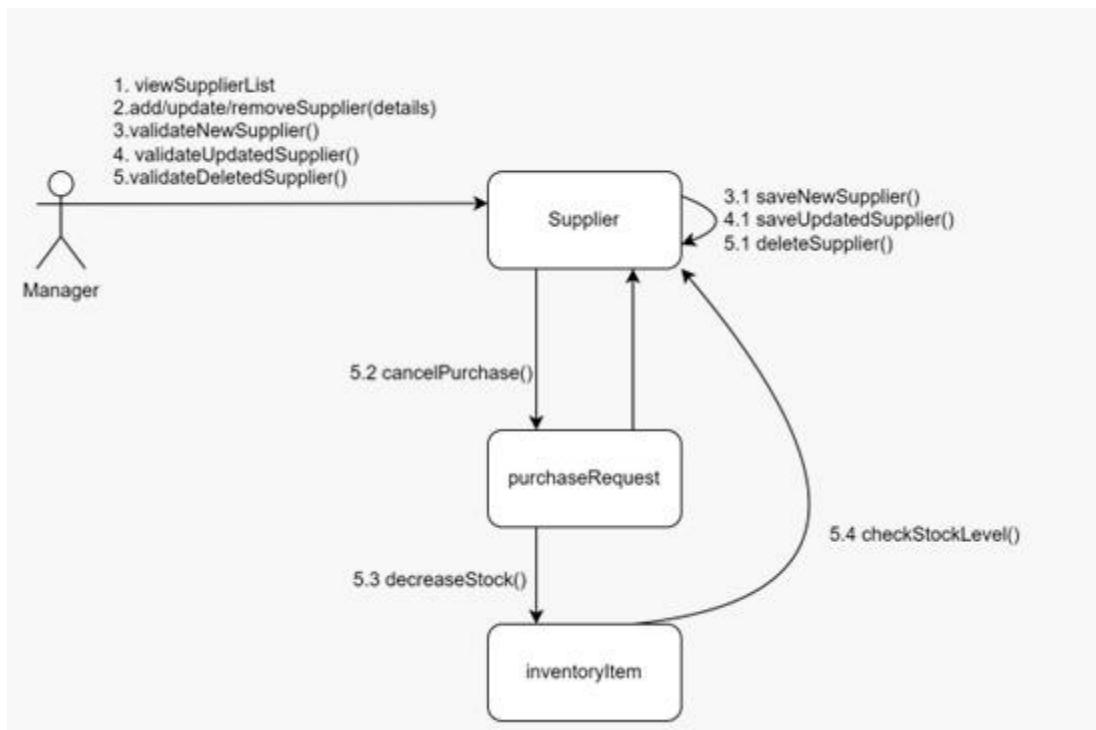
CD-25 Resource Allocation (Eglis Braho)



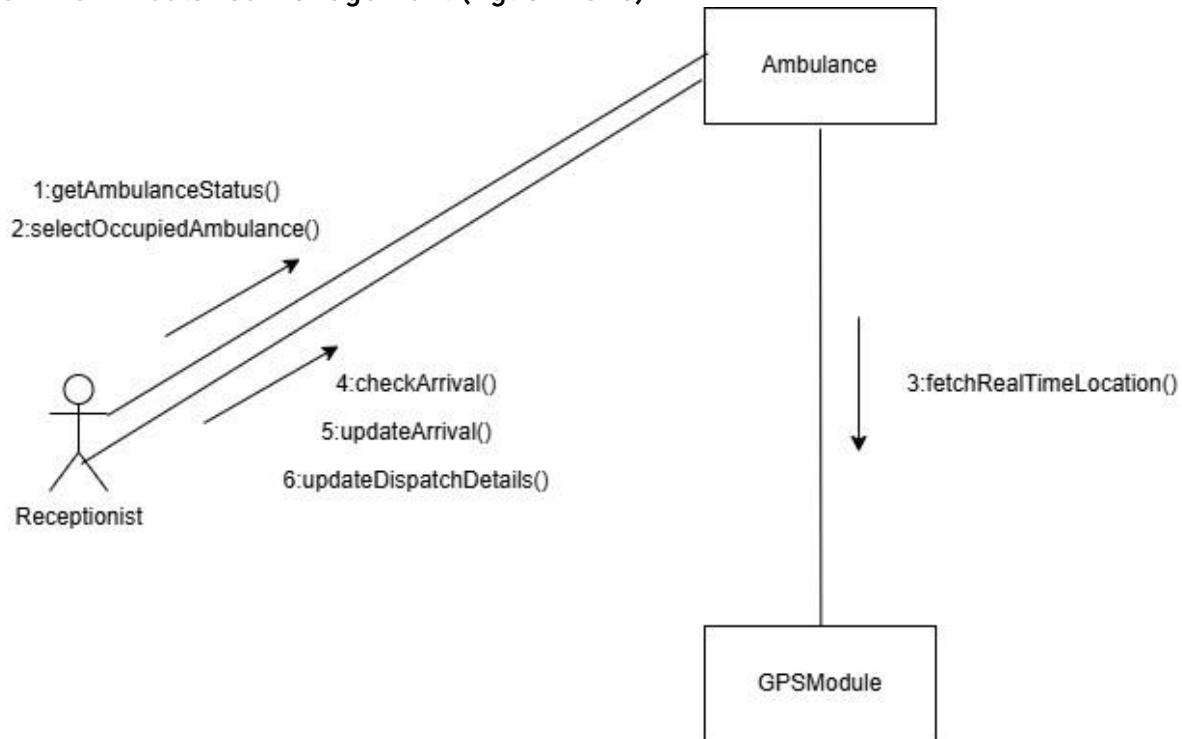
CD-26 Inventory Item & Procurement Management (Arjan Muka)



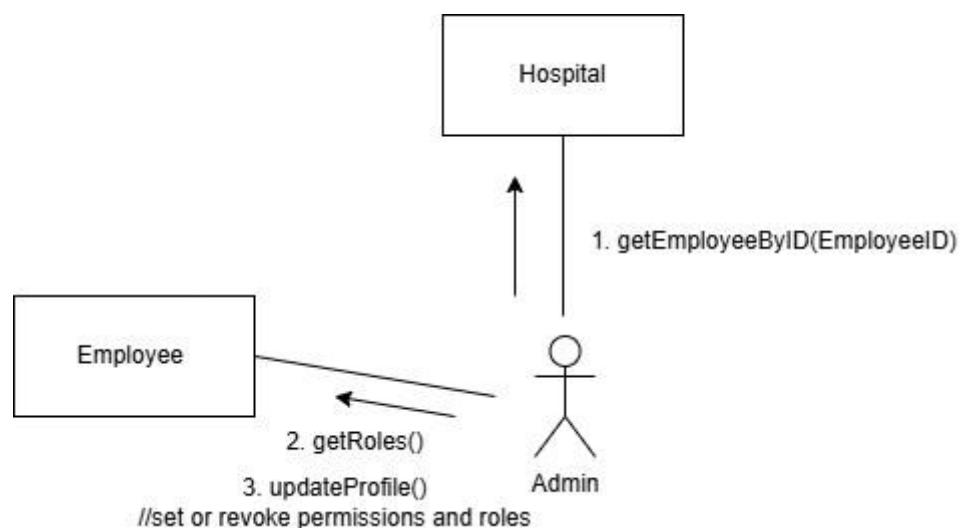
CD-27 Supplier Management (Marin Tartara)



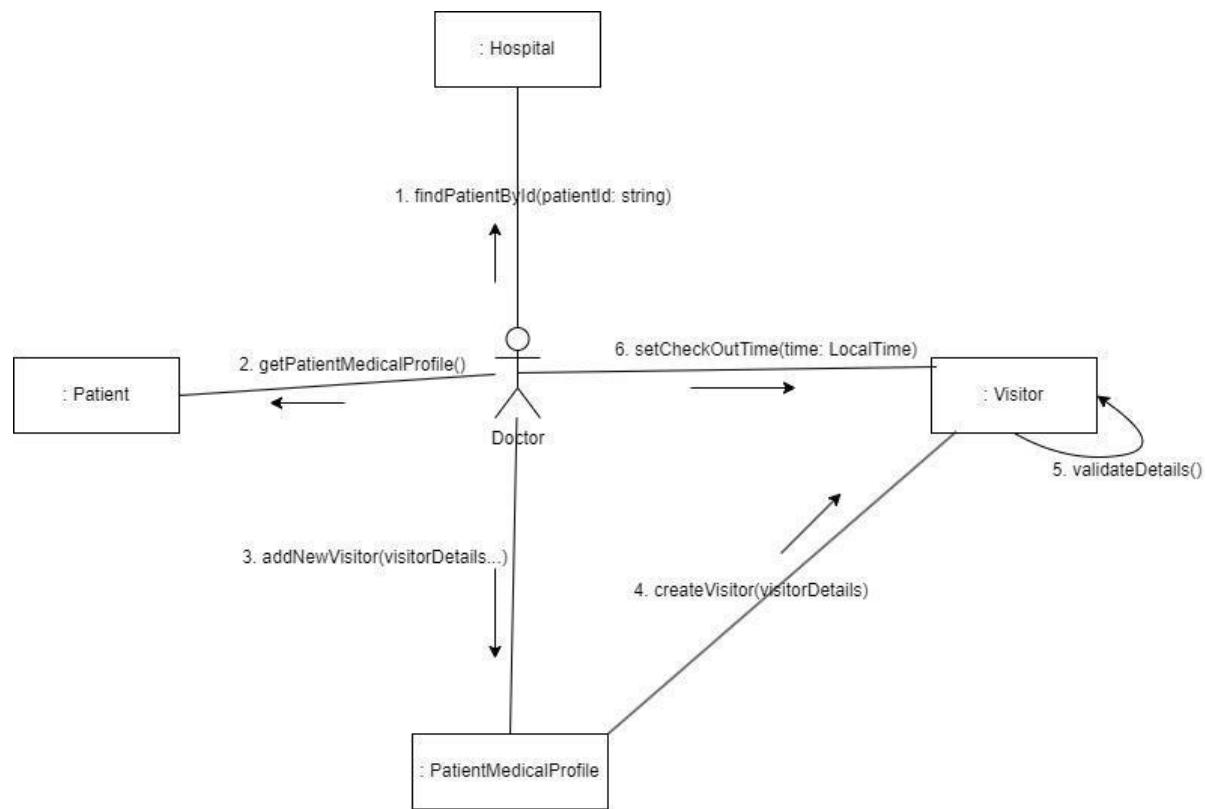
CD-28 Ambulance Management (Eglis Braho)



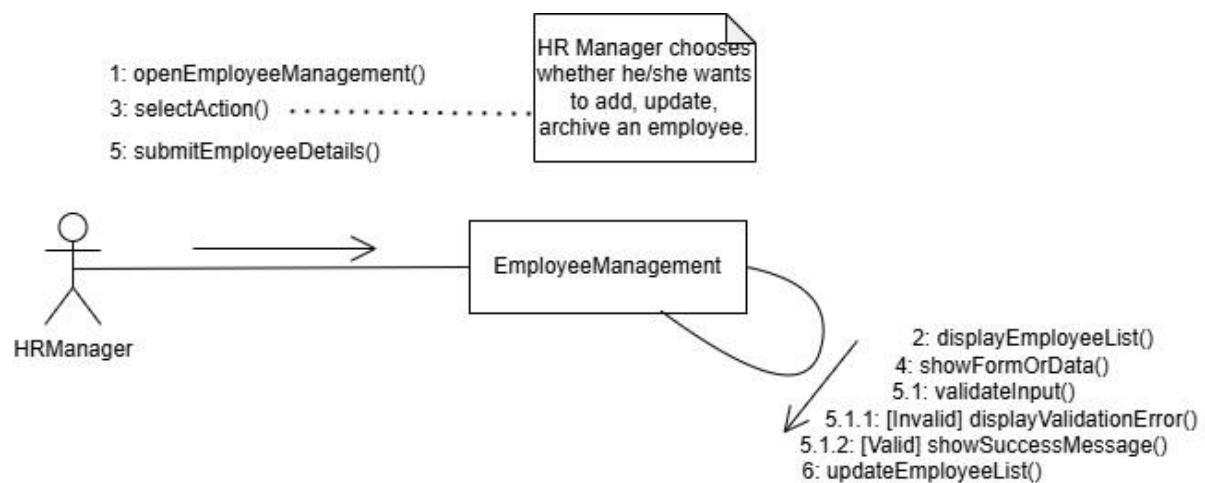
CD-29 Permission Granting (Arjan Muka)



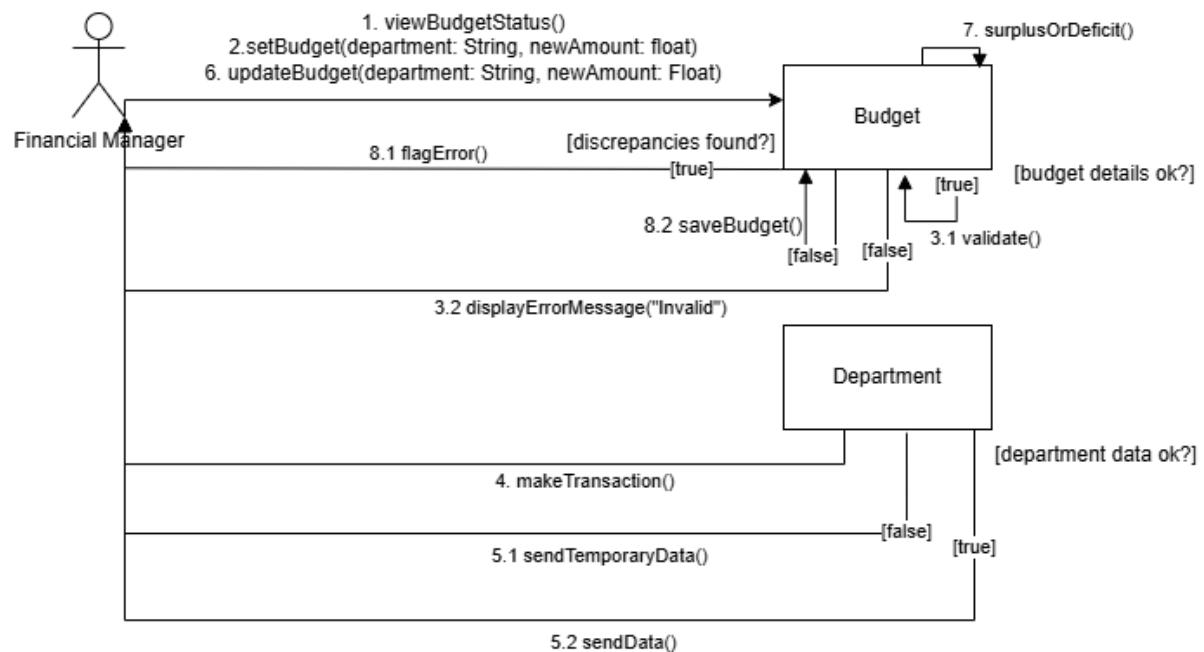
CD-30 Visitor Management (Shpetim Shabanaj)



CD-31 Employee Management System(HR management) (Artjol Zaimi)



C32: Financial Management (Nikola Rigo)



Design Patterns

Shpëtim Shabanaj

Implementation of Builder and Strategy Design Patterns

In this class diagram, I have utilized two design patterns to ensure flexibility and scalability

Creational Design Pattern – Builder

To handle the step-by-step construction of complex Bill objects, I implemented the Builder design pattern.

- I created a BillBuilder interface that defines the necessary steps for building a Bill.
- ConcreteBillBuilder is a concrete implementation of BillBuilder that encapsulates the actual creation logic for a Bill object.
- To manage the construction process, I introduced a BillCreationDirector class. This class holds a reference to a BillBuilder object and is responsible for orchestrating the step-by-step creation process by calling the builder's methods in a specific sequence.
- This design allows different types of builders (like DetailedBillBuilder, SimpleBillBuilder) to be introduced in the future without modifying the existing logic.

The final result is an instance of the Bill class constructed through the defined sequence of method calls on the builder.

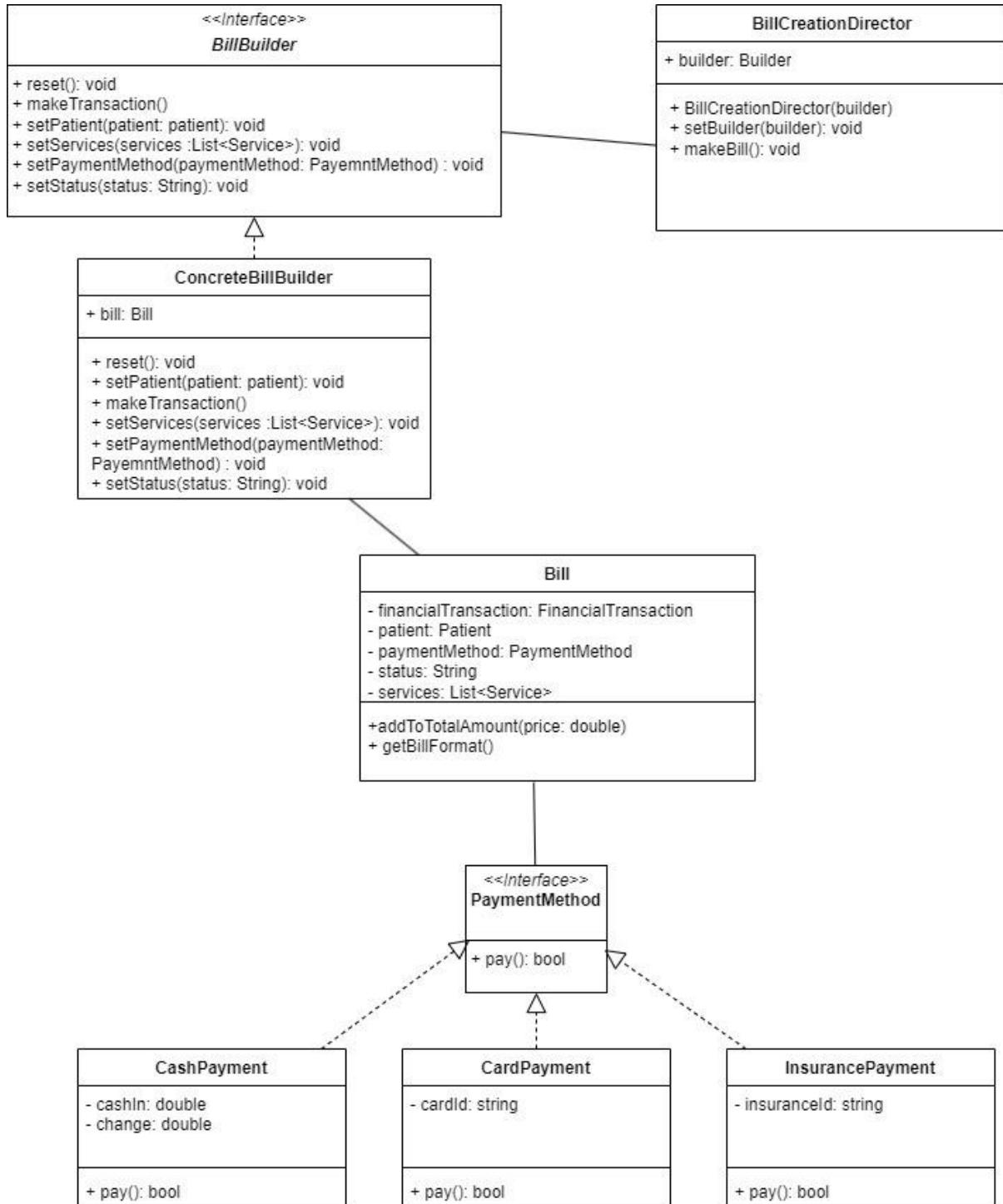
Behavioral Design Pattern – Strategy

Inside the Bill class, there is a reference to a PaymentMethod interface, representing a payment strategy.

- I implemented multiple concrete classes such as CreditCardPayment, CashPayment, and InsurancePayment, each providing its own implementation of the PaymentMethod interface.
- This follows the Strategy design pattern, enabling the Bill object to use different payment strategies at runtime without altering the core logic.
- This approach supports Open/Closed Principle — new payment methods can be added without changing the existing codebase.

Advantages

- The system can be easily extended with additional builder implementations or payment strategies, without violating SOLID principles or affecting existing functionality.



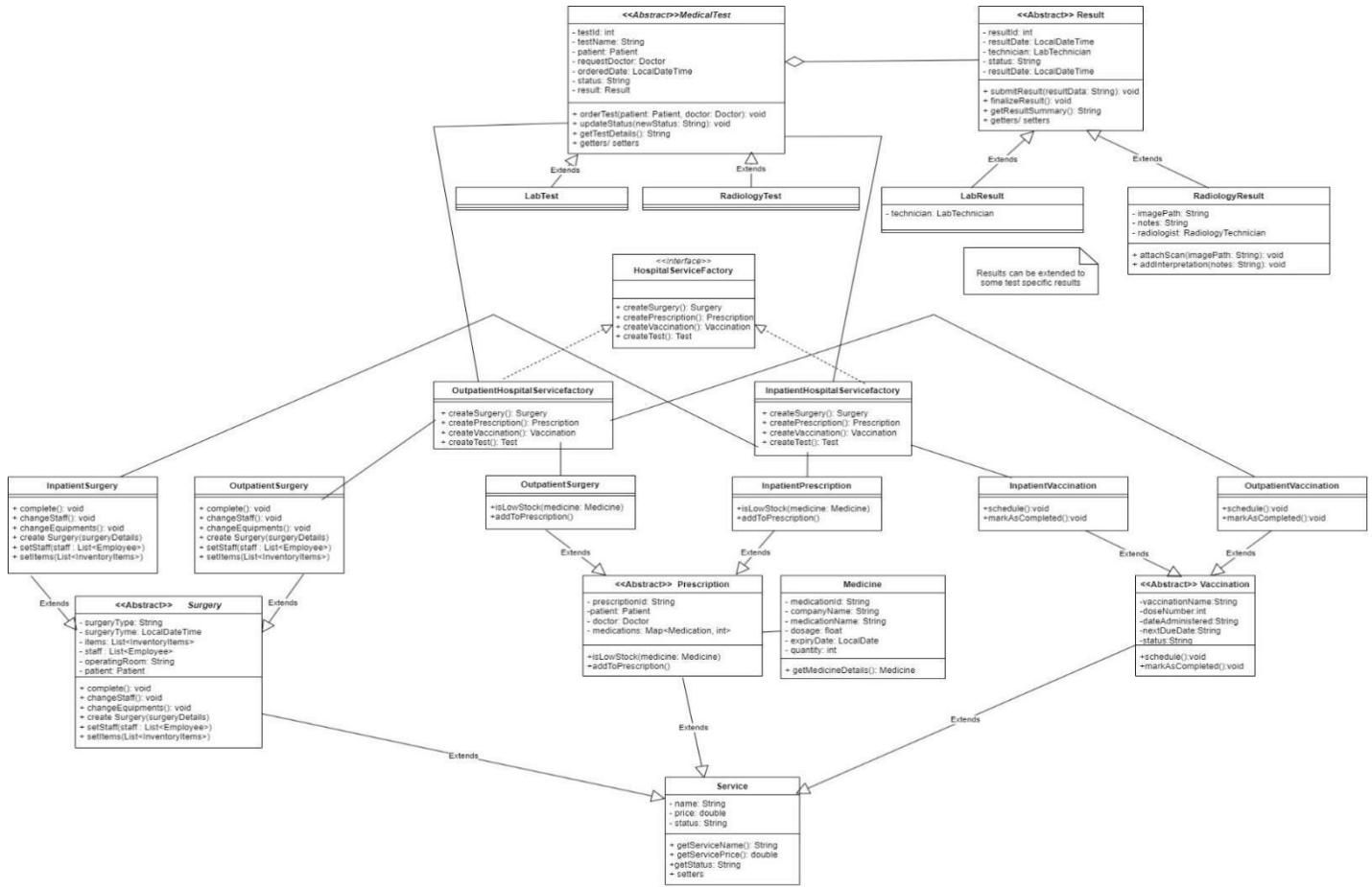
Shpëtim Shabanaj

Implementation of Abstract Factory pattern

In this design, I have implemented the Abstract Factory Pattern through an abstract class `HospitalServiceFactory`. This factory is responsible for creating families of related medical service objects for two distinct categories of patients: inpatients and outpatients.

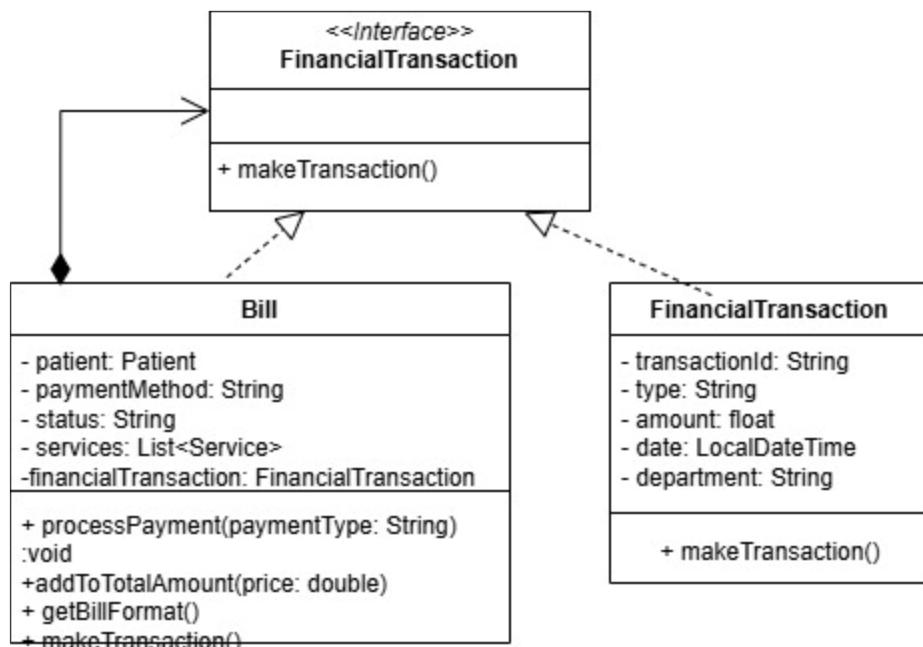
Each concrete factory—`InpatientHospitalServiceFactory` and `OutpatientHospitalServiceFactory`—encapsulates the creation of related services such as surgeries, prescriptions, vaccinations, and medical tests tailored to the specific type of patient.

This pattern promotes extensibility and scalability by allowing new types of services or patient categories to be added without modifying existing code. It also provides a clear and organized structure for object creation, ensuring consistency between related service components. In practice, this is especially useful in hospital systems where service implementations differ significantly between inpatient and outpatient workflows.



Shpetim Shabanaj

In this design, the Decorator Pattern is used to add additional behavior to financial transactions without changing their original implementation. We start with the `FinancialTransaction` interface, which defines the basic structure for any financial transaction. The class `FinancialTransaction` provides a standard implementation of this interface. To extend or customize transaction behavior, we introduce a `Bill` class that also implements `FinancialTransaction`. Instead of duplicating logic, `Bill` wraps an existing `FinancialTransaction` object and can add new features—like calculating tax, applying discounts, or generating receipts—before or after delegating the core transaction operations to the wrapped object. This pattern makes the system flexible and extensible, allowing new features to be added dynamically to transactions without modifying existing code

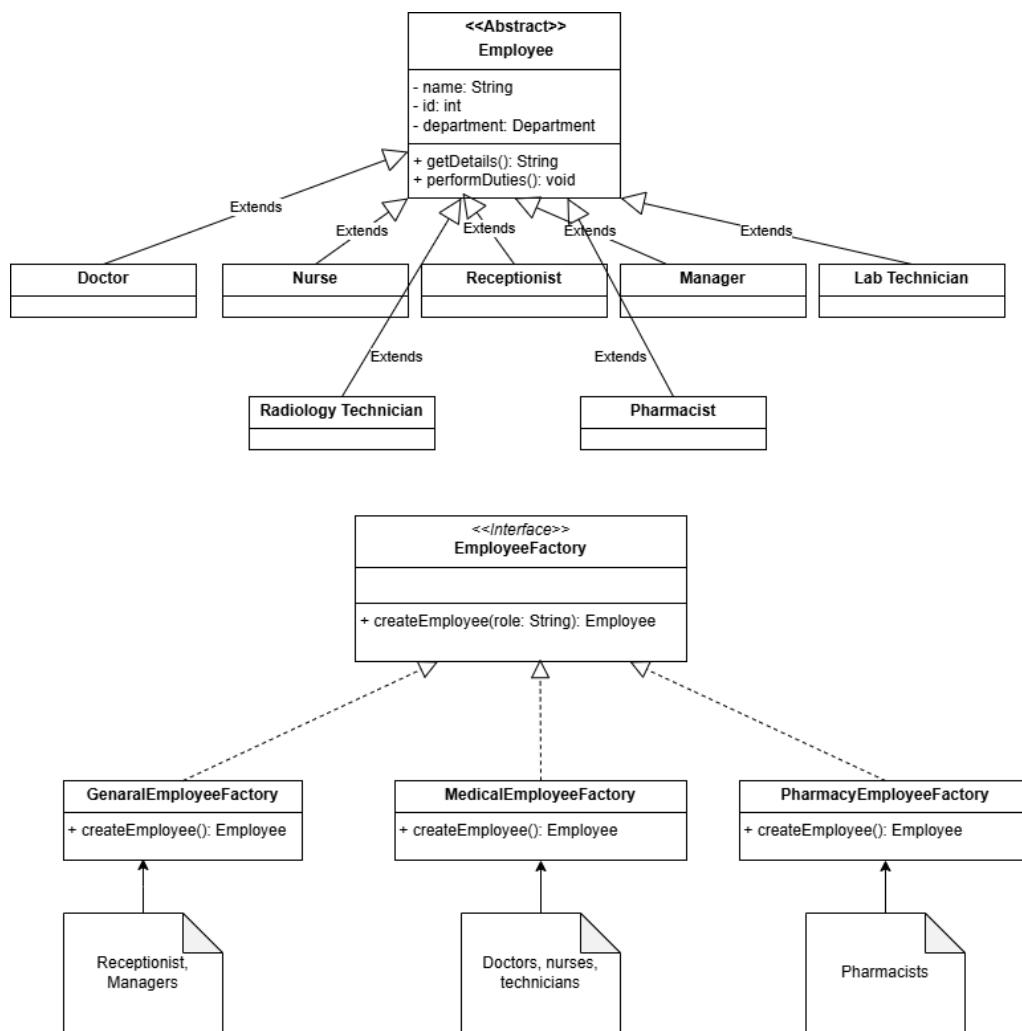


Arjan Muka

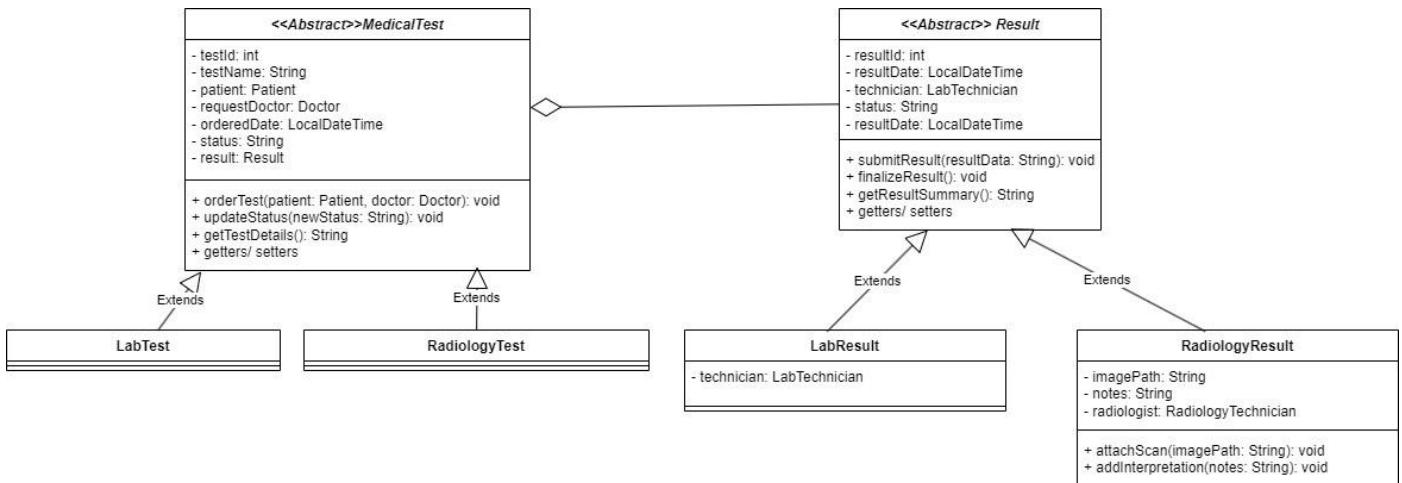
In this design, I have implemented the Factory Method Pattern through the use of an interface `EmployeeFactory`, which defines the method `createEmployee(String role)` for object creation. This pattern allows for the dynamic instantiation of different Employee types (such as Doctor, Nurse, Receptionist, Manager, etc.) based on input parameters like role or department.

Each concrete factory—such as `HospitalEmployeesFactory`, `EmergencyEmployeesFactory`, and `PharmacyEmployeesFactory`—encapsulates the logic for creating specific types of employees. This design provides a clean separation of concerns between client code and object instantiation logic, which adheres to the Open/Closed Principle and eliminates the need for conditional statements like switch-case or if-else in client logic.

An abstract base class `Employee`, which defines common attributes (name, id, department, etc.) and behaviors (`getDetails()`, `performDuties()`). Several concrete subclasses (Doctor, Nurse, Receptionist, Pharmacist, etc.), each overriding `performDuties()` with specific responsibilities relevant to their roles. Multiple factories implementing the `EmployeeFactory` interface to create employees in different hospital contexts (general, emergency, pharmacy).



Nikola Rigo



Bridge Pattern Explanation (Test and Result):

In this design, I have applied the Bridge Pattern to decouple the abstraction (Test) from its implementation details (Result). The Test class is an abstract representation of a medical test (like lab test, radiology test), and it maintains an *aggregation* relationship with the Result class hierarchy, which represents the results of those tests. By using this pattern, the Test abstraction and the Result implementation can evolve independently. For instance, different types of tests such as LabTest and RadiologyTest can be paired with different types of results like RadiologyResult (image-based) or lab-based reports without tightly coupling them. This provides a flexible and scalable solution where new test types or result formats can be introduced without altering the existing code structure. This separation of concerns promotes code reuse, independent extensibility, and maintainability in the system. It is especially valuable in healthcare applications where diagnostic tests and result formats are diverse and subject to change over time.

Artjol Zaimi

Why Use the Proxy Pattern in My Project?

In a hospital system, there are multiple scenarios where **not all users should have the same level of access to sensitive or resource-heavy objects**. For example:

- **Patient medical data must not be accessible to unauthorized staff.**
- **System reports or documents may need filtering or restricted generation rights.**
- **Expensive database queries (e.g., full financial reports) may need caching or delayed loading.**

The **Proxy Pattern** conveniently solves this by placing a "gatekeeper" class in front of sensitive or resource-intensive classes. This allows us to:

- **Apply security policies** (e.g., only doctors can update a patient's medical profile).
- **Add logging and tracking** for accesses to protected objects.
- **Delay the instantiation** of heavyweight objects unless actually needed (lazy loading).

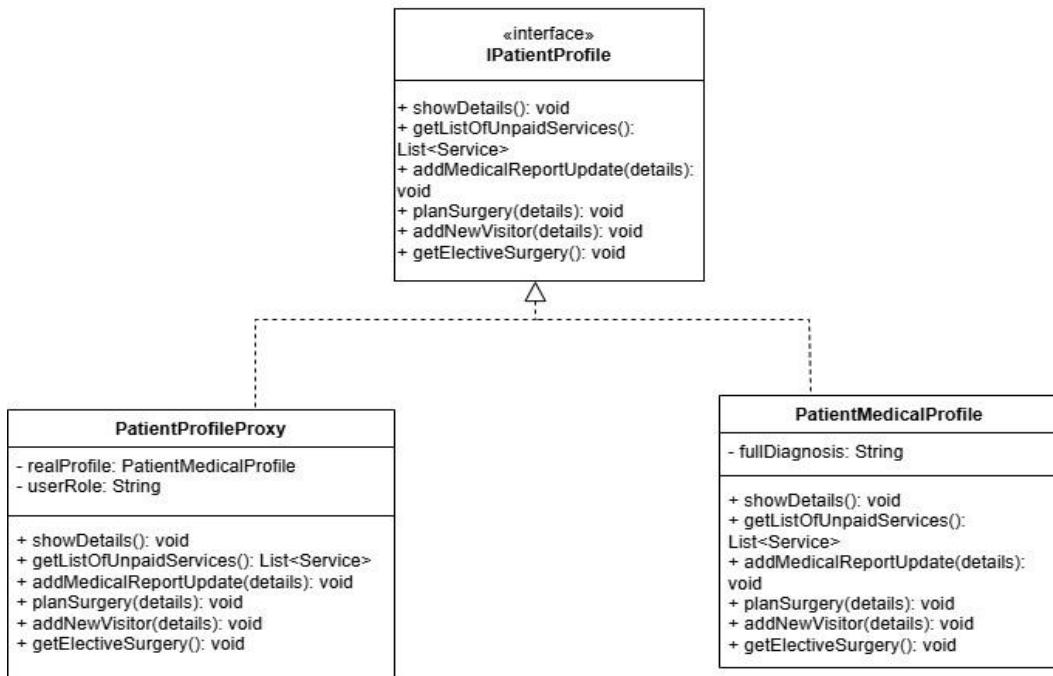
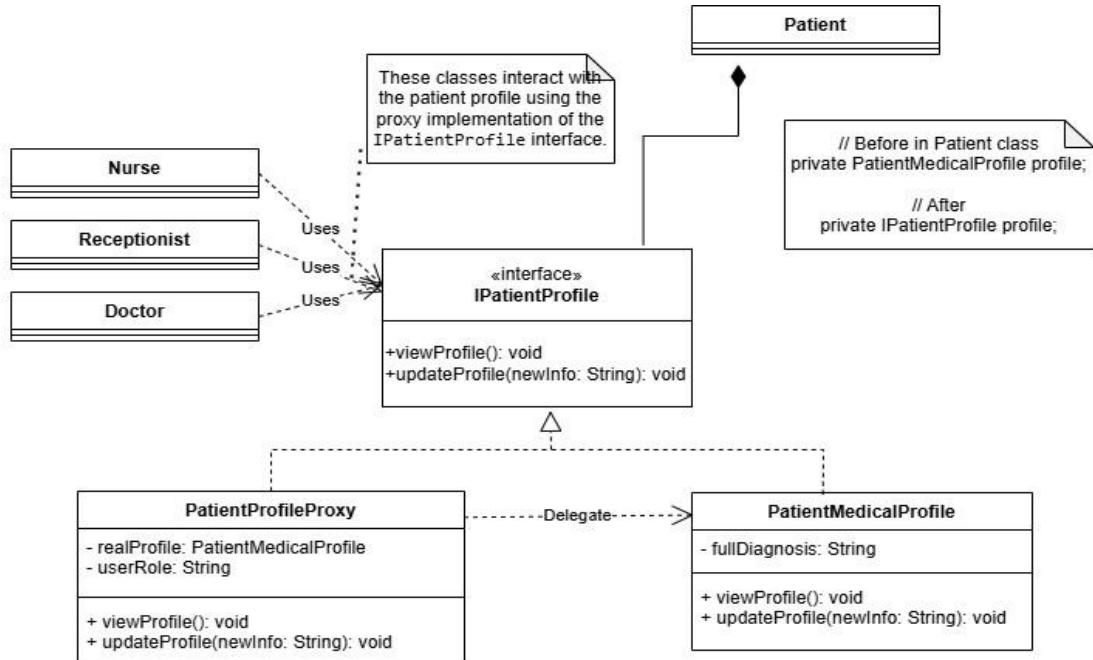
Proxy pattern — Example 1: PatientMedicalProfile

In the project, the **PatientMedicalProfile** class holds confidential medical data such as diagnosis history, test results, and prescriptions. Initially, this class was directly referenced by the **Patient** class and could be accessed by roles like Doctor, Nurse, and Receptionist. However, giving direct access to such sensitive information violates key privacy and security principles.

To solve this, I applied the **Proxy Pattern** by introducing a new class called **PatientProfileProxy**. This class implements the same interface (**IPatientProfile**) as the original profile and internally delegates actions to the real **PatientMedicalProfile**. The main advantage is that the proxy acts as a gatekeeper, adding role-based access control to determine whether the user can call **viewProfile()** or **updateProfile()** based on their role. For example, a Doctor or Nurse may have full access, while a Receptionist would be denied. In my implementation, classes like Doctor, Nurse, or Receptionist interact with the profile like this:

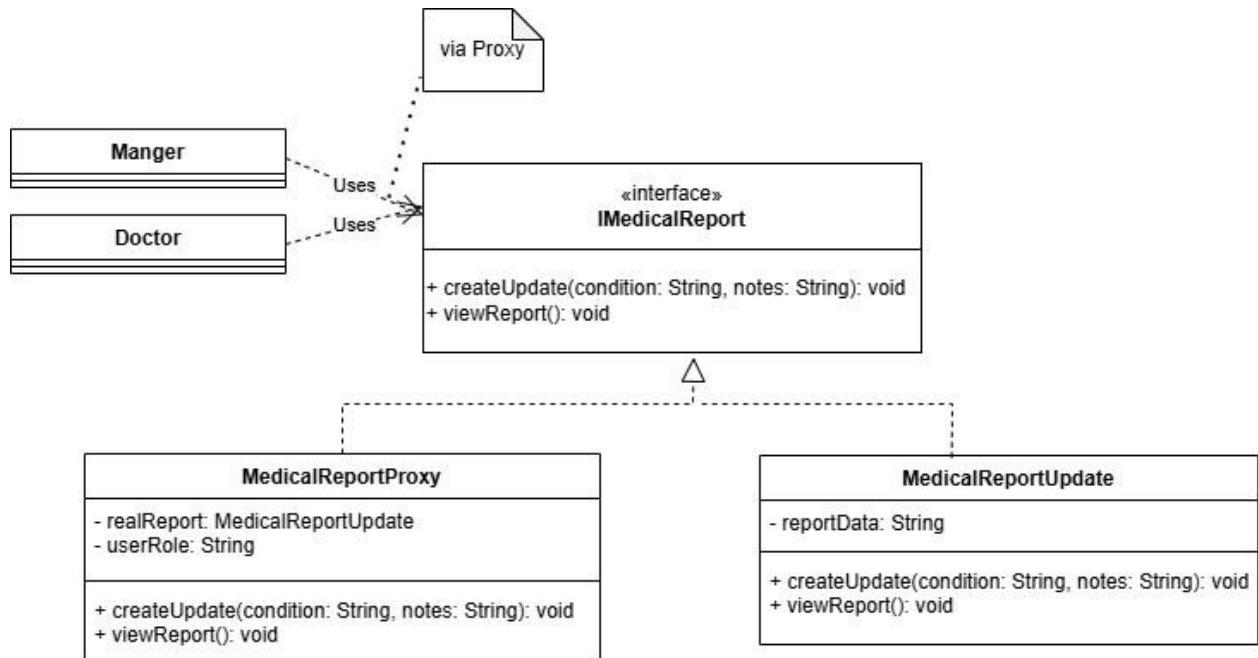
```
IPatientProfile profile = new PatientProfileProxy(patient.getProfile(),  
        userRole); profile.viewProfile();
```

The important part is that these roles only rely on the interface (IPatientProfile) and are unaware of whether they're using the proxy or the real object. This abstraction ensures modularity and securely controls access to medical records without affecting the existing logic of the system.



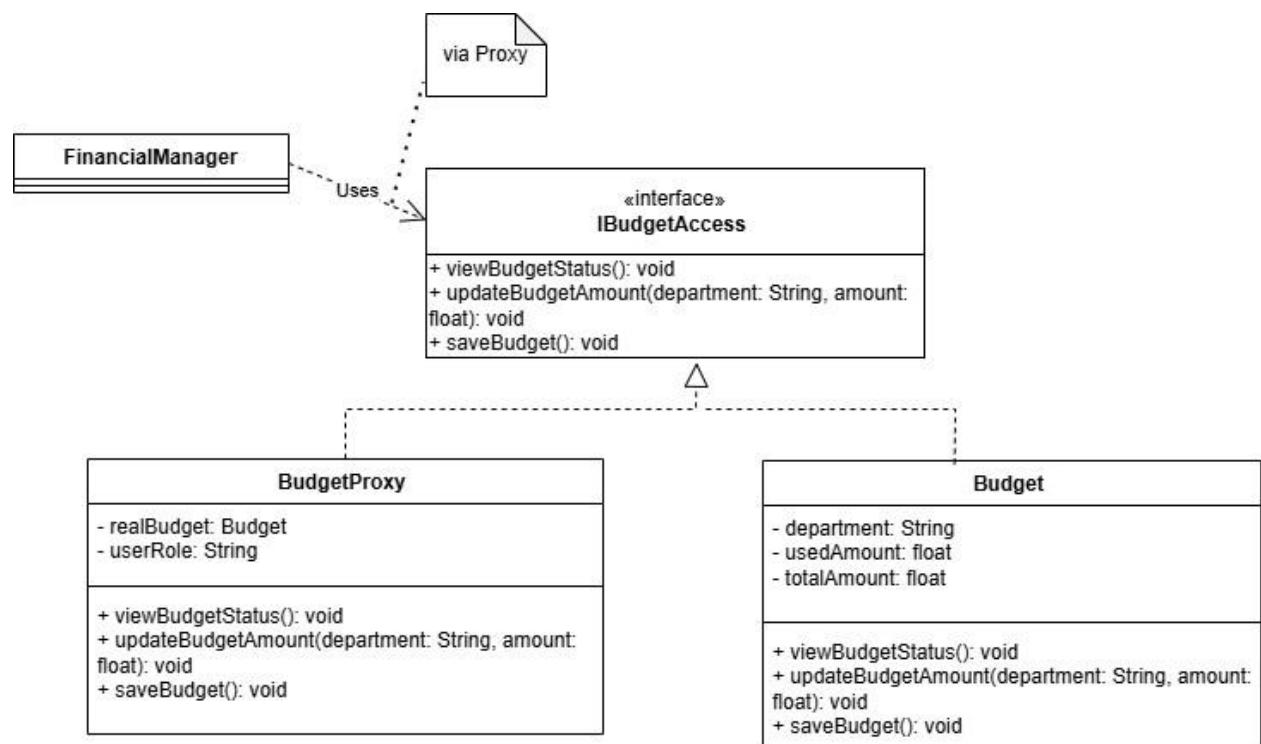
Proxy pattern — Example 2: MedicalReportUpdate

The Proxy Pattern is applied to the `MedicalReportUpdate` class in order to secure sensitive patient information. A proxy object, `MedicalReportProxy`, implements the same interface (`IMedicalReport`) and controls access based on user role. This ensures that only authorized users such as **Doctors** can update reports, while **Managers** may view them. Other roles are denied access. This approach keeps access logic separate from business logic, increases modularity, and complies with data protection standards in healthcare software.



Proxy pattern — Example 3: Budget and financialtransaction

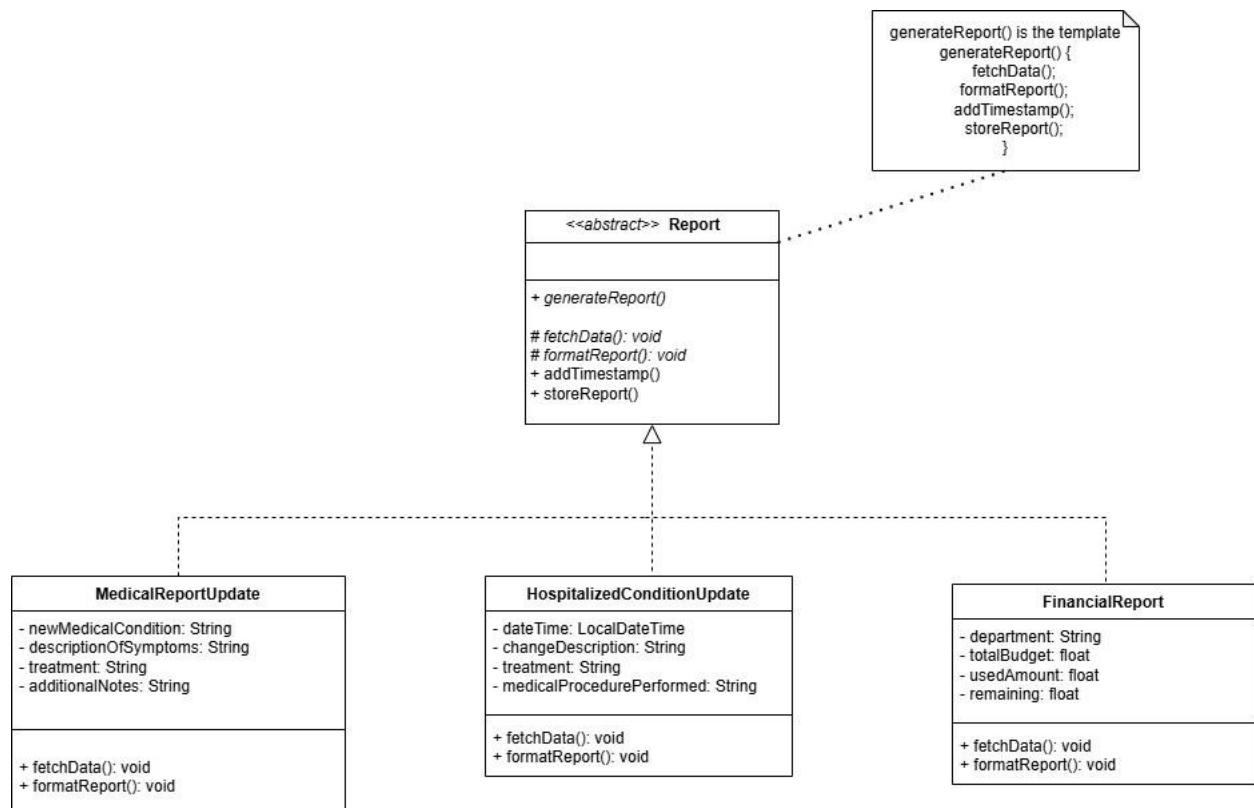
In the project, the `Budget` class manages sensitive financial data like department funds and usage. To make sure only authorized roles—like the Financial Manager—can modify this data, I applied the **Proxy Pattern**. I created a `BudgetProxy` that wraps the real `Budget` object and checks the user's role before allowing actions like updating or saving budget changes. Other roles can be limited to just viewing or even blocked entirely. This way, the proxy enforces access rules while keeping the actual budget logic clean and focused. Users only interact with the interface, without knowing whether it's the proxy or the real object behind it.



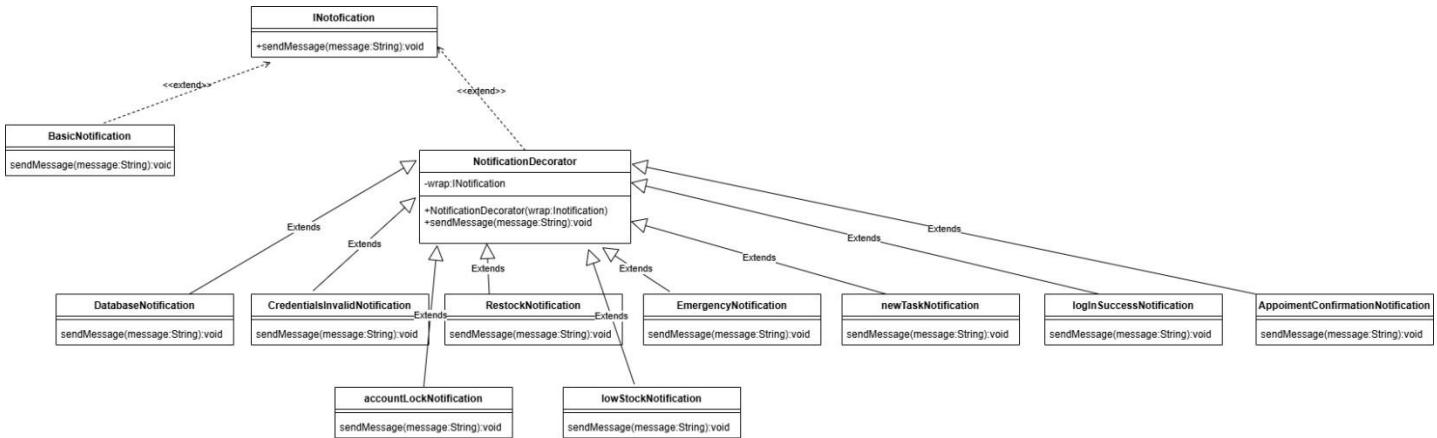
Template method pattern – Report Class Hierarchy

In the Hospital Management System, the report generation process was originally handled independently by each report class. For example, MedicalReportUpdate, HospitalizedConditionUpdate, and FinancialReport each had their own way of fetching data, formatting output, and storing results. This caused duplicated logic, inconsistencies in structure, and made it difficult to manage or scale the reporting system. To solve this, I applied the **Template Method Pattern** by creating an abstract class called Report which defines a fixed method called generateReport(). This method outlines the full workflow: fetchData(), formatReport(), addTimestamp(), and storeReport()—with the first two marked as abstract so each subclass can provide its own specific logic.

By doing this, I was able to ensure that every report follows the same structure while keeping the **flexibility to change only what's necessary**. For example, MedicalReportUpdate fetches diagnosis and treatment data, while FinancialReport pulls budget and transaction details—but both follow the same template when generating their report. The shared methods like addTimestamp() and storeReport() are handled in the base class, avoiding repetition. This not only made the code cleaner and more maintainable, but also aligned with the **Open/Closed Principle**, since I can now add new report types without changing the base logic. It's a scalable, modular solution that fits perfectly into a system where structured yet flexible workflows are essential.



Eglis Braho

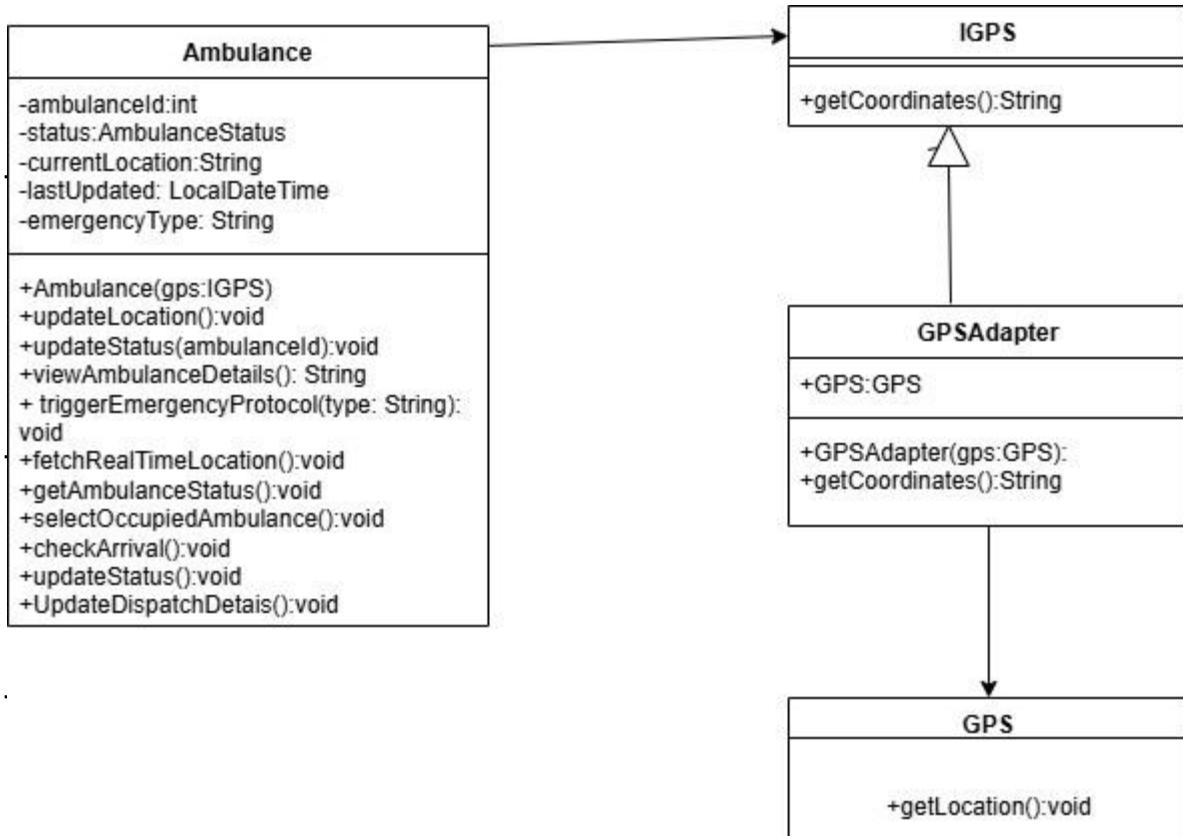


Decorator Design Pattern

In this design, the Decorator Design Pattern is used to extend the functionality of the **INotification** interface without modifying its existing implementations. The **BasicNotification** class provides the core notification behavior, while additional features are dynamically added using decorators.

The abstract class **NotificationDecorator** implements the **INotification** interface and serves as a base for all concrete decorators. Specific decorators such as **DatabaseNotification**, **CredentialsInvalidNotification**, and others extend **NotificationDecorator** and enhance or modify the behavior of the wrapped notification object. This pattern is particularly useful in scenarios where multiple notification types or behaviors (like logging to a database, alerting invalid credentials, sending to different channels) need to be composed flexibly and at runtime. It allows behaviors to be layered without changing the original code, promoting open/closed principle adherence. By using the Decorator Pattern, the system becomes highly extensible, maintainable, and modular, enabling developers to mix and match different notification features in a clean and reusable way.

Eglis Braho



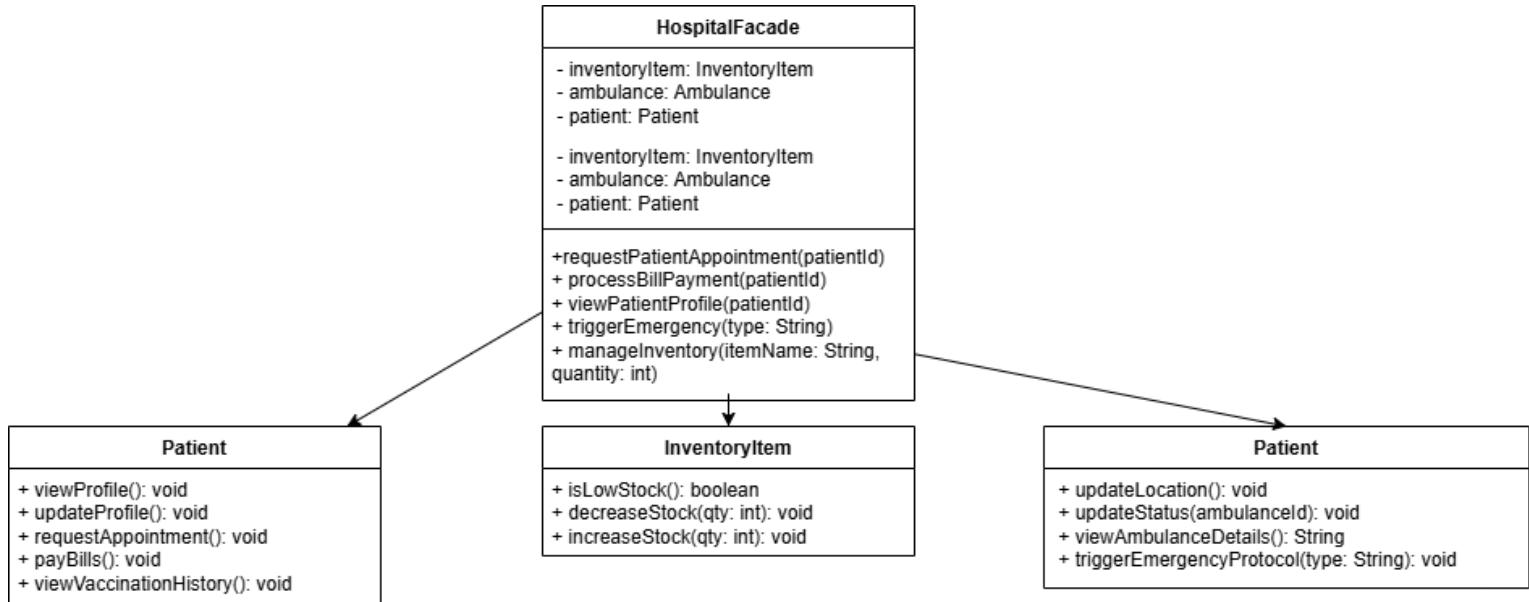
In this design, the Adapter Design Pattern is used to integrate an existing GPS class with the Ambulance system, which expects navigation functionality through the IGPS interface.

The GPS class represents a third-party or legacy component that provides location and routing services, but it does not implement the expected IGPS interface. To resolve this incompatibility without modifying the original GPS class, a GPSAdapter is introduced. This adapter class implements the IGPS interface and internally wraps an instance of GPS, translating calls from the interface into corresponding calls understood by the GPS object.

The Ambulance class depends on the IGPS abstraction, allowing it to work seamlessly with any future GPS implementations that also conform to this interface. The adapter thus enables reuse of existing functionality, ensures interface compatibility, and adheres to the dependency inversion principle by decoupling the high-level ambulance logic from the low-level GPS details.

This pattern is especially useful in systems that need to integrate with external or pre-existing components without altering their source code, providing a clean and flexible integration strategy.

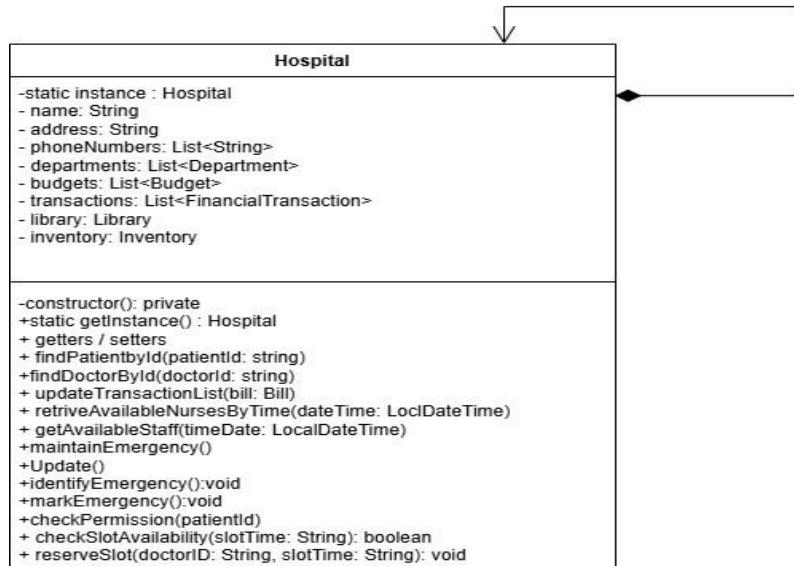
Marin Tartaraj



In this design, the **Facade Design Pattern** is used to provide a simplified interface for interacting with the complex subsystems of the hospital management system. These subsystems—such as **InventoryItem**, **Ambulance**, and **Patient** each handle specific responsibilities, but exposing all of them directly to the client would result in high coupling and increased complexity.

To address this, a **HospitalFacade** class is introduced, which encapsulates and coordinates these subsystems through a unified set of high-level methods like `requestPatientAppointment()` or `triggerEmergency()`. This abstraction hides the internal details of the subsystems, allowing external components (like UI or external services) to interact with the hospital system easily and consistently. By decoupling client code from the individual classes and their interactions, the facade improves maintainability, promotes modular design, and enables better scalability as the system grows.

Arlin Bashllari



In this UML diagram, the `Hospital` class is designed as a **Singleton**, which ensures that **only one instance of the hospital exists in the application**. Here's how it's implemented:

Private Static Instance:

- static instance: Hospital

This holds the single instance of the class.

Private Constructor:

- constructor(): private

The constructor is private so that no external class can instantiate the `Hospital` class directly.

Public Static Accessor:

- + static getInstance() : Hospital

This method returns the single instance of the `Hospital` class, creating it if it doesn't already exist.

Single Point of Access

A hospital system should have **one central controller** that manages resources like departments, staff, transactions, inventory, etc. Singleton ensures centralized management.

Shared Global State

Resources like doctors, patients, inventory, and budgets should be managed **consistently** across the entire application. Singleton ensures that all parts of the system refer to the same data.

Resource Optimization

Since hospital data is often **resource-intensive**, creating multiple instances would be inefficient. The singleton avoids unnecessary duplication and ensures **resource efficiency**.

Thread Safety and Coordination

In systems where multiple threads or modules might interact with hospital data (e.g., booking slots or updating budgets), using a singleton helps **synchronize access**.

□ Benefits of using this design pattern

Centralized control over the hospital system

Ensures consistency in data handling and updates

Easy to maintain and extend as hospital requirements grow

Prevents issues from multiple instances conflicting with one another

