Use Cases

Doctor Use Cases -Antea Koxherri

UC Name	UC_DOC_01 Manage Patient Records
Summary	The system allows doctors to create, edit, and update patient records, including medical history, diagnoses, treatments, and progress notes.
Dependency	**Manage Patient Records** depends on **UC_DOC_03: Manage Doctor's Timetable** for appointment-based updates.
Actors	Doctor
Preconditions	-Doctor must be logged into the system. -If the patient exists, their record must be retrievable from the database. -If the patient is new, the system must allow new patient entry.
Main Sequence	1. **Doctor logs into the system** using a secure username and password. 2. **Doctor navigates to the 'Patient Records' module** from the system dashboard. 3. **Doctor searches for the patient** by name, patient ID, or other identifier. If the patient exists: 4. **The system retrieves the patient's record** and displays the following: • Personal details (name, date of birth, contact information) • Medical history • Diagnoses and treatments • Prescribed medications • Progress notes 5. **Doctor reviews the record** and makes necessary updates. 6. **Doctor modifies or adds new details**, such as:

	• Updating medical history (e.g., new allergies or conditions). • Entering new diagnoses and treatments. • Adding new medications. • Writing progress notes. 7. **Doctor submits the changes.** 8. **The system validates and securely stores the updated record.** If the patient does not exist: 9. **The system informs the doctor that no matching record was found.** 10. **Doctor selects 'Create New Patient'.** 11. **Doctor enters the patient’s personal and medical details.** 12. **Doctor submits the new patient record.** 13. **The system validates and securely stores the new record.**
Alternative Sequence	- If a patient is not found, the system prompts the doctor to create a new record. - If data validation fails, the system requests corrections.
Non-Functional Requirements	- Secure access and encryption - Easy search and update functions
Postconditions	- The patient record is updated (if the patient existed). -A new patient record is created (if the patient did not exist). -The information is securely stored and available in real time.

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

	interacts with the **Pharmacy Use Case** for prescription fulfillment.
Actors	Actors Primary Actor: Doctor Secondary Actors: Pharmacy, Patients
Preconditions	- Doctor must be logged in. - The patient must have an active medical record.
Main Sequence	1. The doctor accesses the "Electronic Prescription" module. 2. The system displays a list of patients or allows the doctor to search for a specific patient. 3. The doctor selects a patient to open their medical profile. 4. The system retrieves the patient's medical history, including existing medications and allergies. 5. The doctor selects the option to create a new prescription. 6. The doctor enters prescription details such as: ○ Medication name ○ Dosage and frequency ○ Duration of use 7. The system automatically checks for potential drug interactions, allergies, and contraindications. 8. If interactions are detected, the system alerts the doctor and provides suggested alternatives. 9. The doctor reviews the prescription, makes adjustments if needed, and confirms it. 10. The system securely sends the prescription to the selected pharmacy. 11. The doctor receives confirmation that the prescription was successfully transmitted.
Alternative Sequence	- If the doctor enters incomplete prescription details, the system prompts for missing information before proceeding.

	-If the patient does not have a medical record, the doctor is prompted to create one before prescribing medication. -If the doctor accidentally submits a duplicate prescription, the system warns them and asks for confirmation.
Non-Functional Requirements	- High security and encryption - Real-time interaction checking - Integration with pharmacy systems
Postconditions	• The prescription is securely stored and transmitted to the pharmacy. • The pharmacy can proceed with fulfillment without delays. • The patient's medical record is updated with the new prescription.

UC Name	UC_DOC_03 Manage Doctor’s Timetable
Summary	This use case allows doctors to manage their schedules by viewing, updating, and organizing patient appointments, surgeries, and administrative tasks. The system ensures that patient bookings are automatically reflected in the timetable and prevents scheduling conflicts. Doctors can also manually add appointments if a patient is physically present or unable to book online.
Dependency	Manage Doctor’s Timetable depends on the UC_PT_02 for scheduling updates
Actors	• Primary Actor: Doctor • Secondary Actors: System, Patients
Preconditions	- The doctor must be logged into the system with valid credentials. -The system must have access to the doctor’s existing schedule and patient appointments. -The system must support patient-initiated bookings that automatically reflect in the doctor’s schedule. -The doctor must have the appropriate system permissions to modify the schedule.

Main Sequence	1. Doctor logs in using secure credentials. 2. Doctor navigates to the "Timetable" or "Schedule" section. 3. The system retrieves and displays the doctor’s full schedule, including: • Confirmed Patient Appointments (automatically booked by patients). • Pending Appointments (requests that require manual approval). • Surgery Schedules. • Administrative Meetings or Tasks. • Breaks or Blocked Time Slots. 4. Doctor views appointment details, including: • Patient’s name, contact details, and medical history (if authorized). • Reason for the visit. • Appointment type (physical visit, teleconsultation, follow-up). • Appointment status (confirmed, pending, rescheduled). 5. Doctor updates the schedule as needed: • Manually Add Appointments (New Step!): ○ If a patient is physically present or unable to book online, the doctor selects "Add Appointment". • Doctor enters patient details (or selects an existing patient from the system). • Doctor selects an available time slot. • System validates availability and prevents double booking. • Doctor confirms, and the system updates the timetable. • Patient receives an appointment confirmation. • Reschedule Appointments: Selects a new time slot → System checks for conflicts → Patient is notified. • Block Time Slots: Doctor blocks time for personal use or administrative tasks → System prevents bookings in those slots.

	6. System automatically prevents double bookings and ensures time slot availability. 7. Doctor confirms and saves changes. 8. System updates the timetable and sends notifications to affected patients (if changes were made).

Organ Donor Coordinator Use Cases- Antea Koxherri

UC Name	UC_ORG_01 Register Organ Donors
Summary	The system allows coordinators to register organ donors by storing personal and medical details. It automatically generates donor-recipient matches and notifies coordinators for further review.
Dependency	**Register Organ Donors** depends on the **Donor-Recipient Matching System** to generate matches
Actors	Organ Donor Coordinator
Preconditions	- The coordinator must be logged into the system with valid credentials. -The donor must provide required personal and medical details for registration. -The system must have access to the donor-recipient matching database to check for potential matches.
Main Sequence	1. Coordinator logs into the system using valid credentials. 2. Coordinator navigates to the 'Register Donor' section from the dashboard. 3. Coordinator selects 'Add New Donor'. 4. Coordinator enters donor's personal details, including: • Name • Date of birth • Contact information • Address 5. Coordinator enters donor's medical details, including: • Blood type • Known allergies • Any medical conditions that may affect donation

	6. The system validates the entered information to ensure completeness and correctness. 7. The system stores the donor details in the database for future reference. 8. The system attempts to generate a donor-recipient match based on: • Blood type compatibility • Urgency level of recipient • Geographic proximity 9. The system notifies the coordinator with a match recommendation if a match is found. 10. Coordinator reviews the match details and finalizes the registration. 11. System confirms donor registration and logs the activity.
Alternative Sequence	- If validation fails, the system prompts for corrections. - If no match is found, the donor remains in the database for future matches.
Non-Functional Requirements	- Secure data storage - Fast retrieval of donor information
Postconditions	- Donor details are saved in the system. - A match is generated (if available). - Notification sent to the coordinator.

UC Name	UC_ORG_02 Generate Reports
Summary	The system allows coordinators to generate monthly and yearly reports covering donor registrations, transplants, waiting list status, and key metrics.
Dependency	**Generate Reports** depends on **UC_ORG_01: Register Organ Donors** for donor data
Actors	Organ Donor Coordinator
Preconditions	- Coordinator must be logged into the system. - The database must contain donor and transplant data.

Main Sequence	1. Coordinator logs into the system using authorized credentials. 2. Coordinator selects 'Generate Report' from the dashboard. 3. Coordinator chooses the report type and time period (monthly or yearly). 4. • The system retrieves relevant data from the database, including: • Number of donor registrations • Number of successful transplants • Current waiting list statistics 5. The system processes the data and compiles it into a structured report. 6. The system formats the report (e.g., PDF, Excel, or dashboard view). 7. Coordinator reviews the generated report for accuracy. 8. Coordinator exports or shares the report with relevant stakeholders.
Alternative Sequence	- If no relevant data is available, the system notifies the coordinator that report generation is not possible. -If an error occurs during report generation, the system logs the error and provides a retry option. -If the system takes too long to generate the report, the coordinator is given an option to receive it via email when ready
Non-Functional Requirements	- Fast data retrieval - Secure report storage - Compliance with healthcare regulations
Postconditions	- The report is generated and available for review. - The coordinator can download or share the report.

Nutricionist Use Cases-Flavia Koco

UC Name	**Dietary Record Management & Monitoring**
UC Code and Name	UC_NUT_01: Dietary Record Management & Monitoring
Summary	The system shall allow the nutritionist to manage (view, add, update, and delete) patient dietary records, assess intake, create personalized plans, and generate progress reports for data-driven decision-making.
Dependency	This use case may depend on the "Patient Profile Management" use case for retrieving patient details.
Actors	**Primary Actor:** Nutritionist **Secondary Actors:** Patient (for providing dietary data)
Preconditions	● The nutritionist must be logged into the system. ● The patient must have a registered profile in the system. ● If the patient is new, the system must allow the entry of a new patient profile.
Description of the Main Sequence	1. The nutritionist logs into the system using a secure username and password. 2. The nutritionist navigates to the “Dietary Record Management” module from the dashboard. 3. The nutritionist searches for a patient using a unique identifier (e.g., name, patient ID, or contact details). 10. **If the patient exists:** 4. The system retrieves and displays the patient's dietary records. 5. The nutritionist reviews and updates dietary records. 6. The nutritionist modifies or adds new details. 7. The nutritionist submits the changes. 8. The system validates and securely stores the updated record. 11. **If the patient does not exist:** 9. The system informs the nutritionist. 12. The nutritionist creates a new patient record. 13. The system validates and securely stores the new record. 4. The nutritionist generates a progress report. 5. The system compiles and presents the report for review. 6. The nutritionist finalizes and saves the report.
Description of the Alternative Sequence	1. If the patient has no existing dietary records, the nutritionist creates a new dietary record. 2. If invalid data is entered, the system prompts the nutritionist to correct it.

	3. If the patient profile is incomplete, the system notifies the nutritionist.
Non-Functional Requirements	- The system should allow real-time access to dietary records. - Data security measures must be implemented to ensure patient confidentiality. - The system should support a user-friendly interface with role-based access control.
Postconditions	- The dietary records are updated and saved in the system. - A personalized dietary plan is created. - A progress report is generated for monitoring dietary intake.

UC Name	**Healthcare Collaboration**
UC Code and Name	UC_NUT_02: Healthcare Collaboration
Summary	The system shall enable the nutritionist to share patient data with other healthcare professionals for coordinated care.
Dependency	This use case may depend on "Dietary Record Management & Monitoring" for accessing patient dietary records.
Actors	**Primary Actor:** Nutritionist **Secondary Actors:** Other Healthcare Professionals (Doctors etc)
Preconditions	- The nutritionist must be logged into the system. - The patient's dietary records must exist in the system.
Description of the Main Sequence	1. The nutritionist logs into the system. 2. The nutritionist selects the "Share Patient Data" option. 3. The system displays a list of available patient records. 4. The nutritionist selects a patient record and chooses a healthcare professional to share it with.

	5. The system verifies access permissions and confirms data-sharing authorization. 6. The system securely shares the data with the selected healthcare professional. 7. The system notifies the recipient of the shared patient data.
Description of the Alternative Sequence	1. If the healthcare professional is not authorized, the system denies access. 2. If no patient records are available, the system notifies the nutritionist. 3. If network issues occur, the system prompts the user to retry later.
Non-Functional Requirements	- The system must ensure secure data sharing using encryption and access control mechanisms. - Only authorized healthcare professionals should access patient data. - The system should log all data-sharing activities for auditing purposes.
Postconditions	- Patient data is securely shared with authorized healthcare professionals. -The recipient healthcare professional receives a notification about the shared data.

UC Name	**Consultation Management**
UC Code and Name	UC_NUT_03: Consultation Management
Summary	The system shall allow the nutritionist to schedule, update, and conduct consultations, both in-person and virtual, for flexible patient care.
Dependency	This use case may depend on "Patient Profile Management" for retrieving patient details.

Actors	**Primary Actor:** Nutritionist **Secondary Actors:** Patient
Preconditions	- The nutritionist must be logged into the system. - The patient must have an active profile in the system.
Description of the Main Sequence	1. The nutritionist logs into the system. 2. The nutritionist accesses the "Consultation Management" module. 3. The system displays a calendar with available consultation slots. 4. The nutritionist selects an available time slot and schedules a consultation. 5. The system confirms the appointment and notifies the patient. 6. On the scheduled date, the nutritionist conducts the consultation (in-person or virtual). 7. The nutritionist records notes and recommendations after the consultation.
Description of the Alternative Sequence	1. If the patient requests rescheduling, the system allows the nutritionist to modify the appointment. 2. If the patient does not show up, the system logs a missed consultation. 3. If network issues occur in virtual consultations, the system suggests rescheduling.
Non-Functional Requirements	- The system should provide real-time scheduling updates to prevent conflicts. - Secure communication must be ensured for virtual consultations. - The system should provide automated reminders to both the nutritionist and patient.
Postconditions	- The consultation is successfully completed. - The system updates the consultation records.

UC Name	**Patient Engagement & Compliance**
UC Code and Name	UC_NUT_04: Patient Engagement & Compliance
Summary	The nutritionist shall be able to provide patients with educational materials, meal plans, send reminders, and track adherence to support patient engagement and compliance with dietary plans.
Dependency	This use case may depend on "Dietary Record Management & Monitoring" for accessing patient dietary data.
Actors	**Primary Actor:** Nutritionist **Secondary Actor:** Patient
Preconditions	- The nutritionist must be logged into the system. - The patient must have a registered profile in the system.
Description of the Main Sequence	1. The nutritionist logs into the system. 2. The nutritionist accesses the "Patient Engagement" module. 3. The system displays a list of patients and their dietary plans. 4. The nutritionist selects a patient and provides educational materials or updates meal plans. 5. The system sends reminders to the patient. 6. The patient logs dietary intake and progress updates. 7. The system tracks patient adherence and generates a compliance report. 8. The nutritionist reviews the report and adjusts the dietary plan if needed.

Description of the Alternative Sequence	1. If the patient does not engage with reminders, the system escalates the notification frequency. 2. If the patient misses logging their intake, the system prompts them to enter data. 3. If the nutritionist updates a dietary plan, the system automatically notifies the patient.
Non-Functional Requirements	- The system must ensure secure messaging for patient communication. - The system should provide automated reminders via multiple channels (email, SMS, app notifications). - Data analytics should track patient adherence trends.
Postconditions	- The patient receives educational materials and meal plans. - The system tracks and records patient adherence. - The nutritionist can assess patient compliance and modify plans accordingly.

Pharmacy Staff Use Cases-Flavia Koco

UC Name	**Prescription Management**
UC Code and Name	UC_PH_01: Prescription Management
Summary	The system shall manage prescriptions by allowing pharmacy staff to securely access, verify, process, and update patient prescriptions while checking for drug interactions, duplicate prescriptions, and ensuring secure communication with doctors.
Dependency	This use case may depend on "Patient Profile Management" and "Doctor Prescription Entry."
Actors	**Primary Actor:** Pharmacy Staff **Secondary Actors:** Doctors, Patients
Preconditions	- The pharmacy staff must be logged into the system. - The patient must have a valid prescription from a doctor.
Description of the Main Sequence	1. The pharmacy staff logs into the system using secure credentials. 2. The pharmacy staff navigates to the "Prescription Management" module. 3. The system displays a list of active prescriptions. 4. The pharmacy staff searches for a prescription using patient ID, prescription number, or doctor name. 5. The system retrieves the prescription details, including: ○ Patient information ○ Medication details (name, dosage, quantity) ○ Prescribing doctor details ○ Expiry date of prescription 6. The pharmacy staff verifies prescription validity and checks for: ○ Drug interactions ○ Duplicate prescriptions ○ Dosage errors

	7. If no issues are found, the pharmacy staff processes the prescription and dispenses the medication. 8. The system updates the prescription status to "Fulfilled." 9. The system notifies the patient that their medication is ready for collection. 10. If verification from the doctor is required, the system notifies the doctor.
Description of the Alternative Sequence	1. If a drug interaction is detected, the system alerts the pharmacy staff and suggests alternatives. 2. If a duplicate prescription is found, the system prompts for manual review. 3. If the prescription is incomplete or invalid, the system requests clarification from the doctor. 4. If the patient does not collect the prescription within a defined period, the system sends a reminder.
Non-Functional Requirements	- The system must encrypt prescription data to ensure security. - Secure messaging should be available for doctor communication. - The system should maintain a complete audit log of prescription handling.
Postconditions	- The prescription is processed, updated, and marked as fulfilled. - The patient is notified to collect their medication.

Use Case ID	UC_PH_03: Customer Loyalty & Discount Program
Summary	The system allows pharmacy staff to enroll patients in a loyalty program, track purchase history, apply discounts for regular customers, and generate reports on high-demand medications and customer purchasing trends.
Dependency	May depend on "Pharmacy Inventory Management" for tracking dispensed medications.

Actors	**Primary Actor:** Pharmacy Staff **Secondary Actor:** Patients
Preconditions	- Pharmacy staff must be logged into the system. - The patient must have a registered profile in the system.
Description of the Main Sequence	1. **Pharmacy staff logs into the system** using secure credentials. 2. **Pharmacy staff navigates to the "Customer Loyalty Program" module** from the system dashboard. 3. **Pharmacy staff searches for an existing patient profile** using name, patient ID, or contact details. ○ If the patient exists, their profile is displayed. ○ If the patient does not exist, staff must register the patient before proceeding. 4. **Pharmacy staff enrolls the patient in the loyalty program** by selecting the enrollment option and confirming participation. 5. **The system generates a unique loyalty ID** and links it to the patient profile. 6. **Patient makes a purchase** at the pharmacy. 7. **The system tracks the patient's purchase history, including:** ○ Medication names and quantities ○ Date of purchase ○ Total amount spent ○ Any applicable insurance or discounts used 8. **The system calculates loyalty points based on the purchase** and updates the patient's account. 9. **The system applies a discount if the patient has accumulated sufficient loyalty points.** ○ If the patient qualifies for a discount, it is applied automatically. ○ If not, the points are saved for future use.

	10. **The system generates a transaction receipt,** displaying purchase details, loyalty points earned, and available discounts. 11. **The pharmacy staff provides the receipt to the patient** and confirms the successful transaction. 12. **The system updates inventory records** to reflect the dispensed medications. 13. **The system periodically generates reports on:** ● High-demand medications ● Customer purchasing trends ● Effectiveness of the loyalty program 14. **Pharmacy staff reviews the reports** to make data-driven decisions regarding inventory and promotions.
Description of the Alternative Sequence	- If the patient is not registered, the system prompts pharmacy staff to create a new profile. - If there’s a discount calculation error, the system notifies pharmacy staff to apply the discount manually or adjust the loyalty balance. - If the patient opts out, the system removes their loyalty status and stops tracking purchases for rewards.
Non-Functional Requirements	- Secure handling of customer data and compliance with privacy regulations. - Real-time discount calculations at checkout. - Automated reports with analytics on customer trends and loyalty program effectiveness.
Postconditions	- Patient is enrolled in the loyalty program (if new). - Purchase is processed, loyalty points are updated, and discounts are applied. - System tracks purchasing trends and generates reports for decision-making.

Patient Use Cases- Elkier Ago

UC_PT_01	**Patient Appointment Management**
Summary	The system shall allow patients to schedule, modify, and cancel appointments.
Dependency	None
Actors	Patient
Preconditions	-The patient must be logged into the system. -The patient must have an active user account.
Main Sequence	**1. Patient logs into the system** using secure login credentials. **2. System verifies the login** and grants access to the patient dashboard. **3. Patient navigates to the "Appointment Management" module** from the dashboard. **4. System displays options** to schedule, modify, or cancel appointments. **5. Patient selects an action** based on their need. If the patient chooses to schedule an appointment:

	6. **System displays available time slots** for the selected provider and date range. 7. **Patient selects a preferred time slot** from the list. 8. **System checks time availability** in real time. 9. If the slot is available: a) Patient confirms the booking. b) **System saves the appointment** and updates the scheduling database. c) **System sends a confirmation notification** to the patient. If the patient chooses to modify an appointment: Patient views upcoming appointments. 6. Patient views upcoming appointments. 7. **Patient selects the appointment** to be modified. 8. System shows alternative available time slots. 9. Patient selects a new time slot. 10. System verifies the new slot's availability. 11. Patient confirms the change.

	12. System updates the appointment and sends a reschedule confirmation notification. If the patient chooses to cancel an appointment: **6.** Patient views their appointment list. **7.** Patient selects an appointment to cancel. **8. System checks cancellation policy** (e.g., timing before appointment). **9.** If cancellation is allowed: **a)** Patient confirms the cancellation. **b) System deletes the appointment** and sends a cancellation confirmation notification.

Alternative Sequence	1. If the patient selects an unavailable time slot: a) **System detects that the slot is already booked.** b) System displays a “No time available” message. c) **Patient selects a new time slot,** and the main sequence resumes from step 8 (scheduling). 2. If the patient attempts to cancel too close to the appointment time: a) **System checks the time remaining until the scheduled appointment.** b) System determines that cancellation is restricted based on policy. c) **System displays a warning message** explaining the cancellation restriction. d) **Patient is prompted to acknowledge the policy** or return to the appointment view without canceling.
Non-Functional Requirements	- The system must ensure real-time appointment availability updates. - Patients must receive automated notifications for appointment actions

	.- The interface must be user-friendly and accessible.
Postconditions	- The appointment database is updated. - The patient receives a confirmation or reschedule notification.

UC_PT_02	**Patient Medical Record Access**
Summary	The system shall provide patients with secure access to their medical records and test results.
Dependency	None
Actors	Patient
Preconditions	-The patient must be logged into the system. - The patient must provide valid authentication credentials. - The system must have existing records for the patient (if viewing).

Main Sequence	**1** **Patient logs into the system** using secure credentials.. **2** **System verifies login credentials** and checks authentication validity. **3** If credentials are valid: **a)** **System grants access** and redirects the patient to the dashboard. **4** **Patient navigates to the “Medical Records” module** from the dashboard. 5 **System initiates a request** to retrieve the patient’s medical records and test results from the database. **6** **System checks if records exist** for the logged-in patient. **7** If records are available: a) **System displays the medical records and test results,** including: • Personal information (e.g., name, birthdate) • Medical history

	• Lab test results • Diagnoses and treatments 8 **Patient views the records** through the interface. 9 **Patient may download or export records** if needed. 10 **System logs the access** and records the timestamp for auditing purposes.
Alternative Sequence	1 If login credentials are invalid: a) System shows an authentication error. b) Patient is prompted to re-enter correct login credentials. 2 If no records are available: a) System shows a "No records found" message.

	b) Patient is redirected back to the dashboard or remains in the module.
Non-Functional Requirements	- The system must ensure encrypted storage and secure access. - Patients must be notified of any changes in their records. - Audit logs should track record access history.
Postconditions	- The patient successfully views or downloads their medical records. - The system logs the access history for security and compliance.

UC_PT_03	**Prescription Refill Management**
Summary	The system shall allow patients to request prescription refills through the platform.

Dependency	None
Actors	Patient
Preconditions	-The patient must be logged into the system. - The patient must have an **active prescription** listed in their records. - The pharmacy system must be connected to receive and process requests.
Main Sequence	1 **Patient logs into the system** using valid credentials. 2 System verifies the login credentials. 3 If the login is valid: a) **System grants access** and redirects the patient to the dashboard. **4. Patient navigates to the “Prescription Refill” module** from the dashboard. **5. System displays a list of active prescriptions** eligible for refill. **6. Patient selects a prescription** from the list. **7.** Patient submits a refill request. **8. System checks the prescription validity** (e.g., not expired, refill limit not reached)

	9. If the prescription is valid: a) System forwards the refill request to the designated pharmacist. b) **System sends a confirmation message** to the patient. **10. Patient views the request status** in the system interface.
Alternative Sequence	1 If login credentials are invalid: a) **System displays a login error** and prompts the patient to re-enter credentials. 2 If the selected prescription is invalid (e.g., expired or limit exceeded): a) **System shows an error message** indicating the issue. 3 If the pharmacy is unable to fulfill the request (e.g., out of stock): a) System notifies the patient of the issue.
Non-Functional Requirements	- The system must ensure secure and accurate prescription management. - Patients must receive timely updates on refill status.
Postconditions	- The pharmacy receives the refill request

	.- The patient is notified of the prescription status.

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

Main Sequence	1. **Patient logs into the system** using valid credentials. 2. **System verifies login credentials** and redirects the patient to the dashboard. 3. Patient navigates to the “Telemedicine” module. 4. **System displays a list of available doctors** along with their open time slots. 5. **Patient selects a doctor** and chooses an available time for the consultation. 6. **System verifies the doctor's availability** and confirms the appointment. 7. **System sends confirmation notifications** to both the patient and the doctor. 8. **At the scheduled time,** the system initiates a secure video/audio session between the patient and the doctor. 9. Patient and doctor conduct the consultation. 10. **Upon completion, the session is closed** and the system logs the interaction. 11. **Doctor may issue follow-up instructions or prescriptions** through the system.

	12. **Patient receives a summary or next steps** via notification or in-app message.
Alternative Sequence	1 If the selected doctor is unavailable (e.g., slot already booked): a) System displays a conflict message. b) 2 If the patient's connection is lost during consultation: a) **System attempts to reconnect automatically.** 3 If the doctor ends the session early due to emergency or technical issue:

	a) **System notifies the patient** and reschedules if needed.
Non-Functional Requirements	- The system must ensure encrypted and high-quality video communication. - Patients must receive timely reminders before consultations.
Postconditions	- The consultation is successfully completed and logged. - The patient receives follow-up instructions or prescriptions if needed.

UC_PT_05	**Patient Feedback and Rating**

Summary	The system shall allow patients to provide feedback and rate their care experience.
Dependency	None
Actors	Patient
Preconditions	-The patient must have received a healthcare service (e.g., appointment, consultation, treatment). -The system must have a record of the completed service linked to the patient’s account.
Main Sequence	1 **Patient logs into the system** using secure credentials. 2 **System verifies login credentials** and grants access to the dashboard. 3 **Patient navigates to the “Feedback and Rating” module** from the dashboard. 4 **System displays a feedback form**. 5 Patient enters a rating and provides optional written feedback. 6 Patient submits the feedback form. 7 **System validates the submission** (e.g., required fields, content length). 8 **System stores the feedback** in the database.

Alternative Sequence	1 If the feedback submission fails, the system notifies the patient and allows retrying. 2 If the feedback is inappropriate, the system flags it for review.
Non-Functional Requirements	- - The system must ensure confidentiality of patient feedback. - Feedback must be structured and categorized for analysis.
Postconditions	- The feedback is stored and available for service improvement. - The healthcare provider receives anonymous reports on ratings.

UC_PT_06	**Emergency Service Access**
Summary	The system shall allow patients to quickly access emergency services, enabling one-click alerts to emergency responders.
Dependency	None
Actors	Patient
Preconditions	- The patient must be logged into the system.
Main Sequence	1 **Patient logs into the system** using valid credentials. 2 **System verifies the login** and redirects the patient to the dashboard. 3 **Patient accesses the “Emergency Services” module** from the dashboard. 4 **System displays a one-click emergency alert button** with clear emergency labeling. 5 Patient clicks the emergency alert button. 6 **System captures the patient's location** (via GPS or profile information). 7 **System compiles the patient’s emergency profile**, including:

	• Full name and age • Medical history or critical conditions (if available) • Emergency contact info • Location coordinates 8 **System immediately sends the alert** to pre-designated emergency responders or local authorities. 9 **System displays confirmation** to the patient that the alert has been successfully sent. 10 **Patient receives real-time updates** on responder status (e.g., “Help is on the way”).
Alternative Sequence	**4.** If location data is not accessible (e.g., GPS disabled): a) System prompts the patient to manually enter their location. b) **Patient enters location**, and the system continues the alert process. **5.** If the emergency request fails (e.g., no internet connection or server error): a) System retries sending the alert up to a set number of times. b) If all retries fail, **system notifies the patient** and attempts to alert a **backup contact** (e.g., via SMS or secondary channel)..

Non-Functional Requirements	- The emergency alert feature must operate with **low latency and high reliability**. - All transmissions must be **encrypted and secure**, especially patient identity and location. -The system must maintain **uptime guarantees** for this critical service. -Alerts should be **timestamped and logged** for auditing and response tracking.
Postconditions	- **Emergency responders receive a complete alert** containing the patient's data and location. -The patient is informed about response status. -**All activity is logged** for future reporting and accountability.

UC_PT_07	Organ Match Request

Summary	The system shall allow patients or their representatives to request organ matches, linking their profiles directly with potential donor databases.
Dependency	None
Actors	**Primary Actor:** Patient
Preconditions	-The patient must be registered and logged into the system. - The patient's medical profile must include up-to-date health records relevant to organ compatibility (e.g., blood type, organ need, urgency status).
Main Sequence	1. **Patient logs into the system** using secure login credentials. 2. **System verifies the login** and redirects the patient to the dashboard. 3. Patient navigates to the "Organ Match Request" module. 4. **System displays an organ match request form**, pre-filled with relevant patient medical data. 5. **Patient reviews and confirms the details** (e.g., organ needed, urgency level). 6. Patient submits the organ match request.

	7. **System connects to national/international donor databases** via secure API. 8. **System performs compatibility analysis** using criteria such as: - Blood type - Organ availability - Medical urgency 9. **System notifies the patient** with a confirmation and next-step instructions.
Alternative Sequence	1. If no match is found during the initial search: a) **System notifies the patient** that no current matches exist. b) **System retains the request** in a monitoring queue and continues searching in real time. c) **Patient receives periodic updates** when new data is available. 2. If patient data is incomplete or outdated: a) System prompts the patient to update their medical profile. b) Request submission is paused until required information is completed.

Non-Functional Requirements	- All data exchange between the platform and donor databases must be **encrypted** -The matching process must be **accurate and efficient**, even under high system load. - The platform should support **real-time alerts** and offer **multilingual support** for users. - Requests and matches should be **timestamped and logged** for legal, ethical, and medical traceability.
Postconditions	- The **organ match request is processed** and entered into the active monitoring system. - If matches are found, **the transplant team is notified** to coordinate the next steps. -The **patient receives a status update**, whether a match is found or pending. -**All actions are logged** in the patient’s medical and administrative history for reference.

IT Support Use Cases- Elkier Ago

UC_IT_01	**IT Support Manages User Accounts**
Summary	The system shall allow IT support to manage user accounts, including creating, modifying, and deactivating user accounts.
Dependency	None
Actors	Primary Actor: IT Support
Preconditions	-IT support must be logged into the admin panel with appropriate permissions.
Main Sequence	1 IT support accesses the "User Account Management" module. 2 The system displays a list of existing user accounts. 3 IT support selects the account to be managed. 4 IT support can create, modify, or deactivate the account. 5 The system confirms the action and updates the user account details. 6 IT support receives confirmation of the action.
Alternative Sequence	1. If IT support attempts to create a duplicate account, the system notifies them and suggests merging or modifying the existing account. 2. If IT support attempts to deactivate an account with active sessions, the system asks for confirmation before proceeding.
Non-Functional Requirements	- The system must ensure that **user account changes are logged** for auditing purposes. - Changes to user accounts should be **reflected in real time**. -The interface must be **secure and user-friendly** for IT support staff.
Postconditions	- **User account data is updated or deactivated** as per IT support's actions. -**IT support receives confirmation** of the completed action.

UC_IT_02	**System Monitors Performance**

Summary	The system shall continuously monitor its performance, including CPU, memory, disk usage, and application status, and notify stakeholders of any performance issues.
Dependency	None
Actors	Primary Actor: System Secondary Actor: IT Support (when intervention is required)
Preconditions	-The system must be running, and performance monitoring tools must be active.
Main Sequence	1. The system starts monitoring performance metrics such as CPU, memory usage, disk space, and network activity. 2. The system collects data at regular intervals and stores it in a log. 3. If a performance threshold (e.g., high CPU usage) is exceeded, the system triggers an alert. 4. IT support receives an alert with performance details. 5. IT support investigates the issue and takes corrective action, if necessary (e.g., scaling resources, terminating processes). The system logs the resolution of the performance issue.
Description of the Alternative Sequence	1. If the system detects a minor issue (e.g., temporary spike in CPU usage), it can either attempt automatic recovery (e.g., load balancing) or escalate the alert to IT support. 2. If IT support is unavailable, the system continues to monitor and escalate issues through automated channels (e.g., emails, SMS alerts).
Non-Functional Requirements	- Performance data should be logged securely and with high accuracy. - Alerts must be timely and provide sufficient data for IT support to diagnose the issue. - The system should be designed for minimal impact on performance during monitoring.
Postconditions	- Performance data is recorded and accessible for review. - IT support resolves or acknowledges the performance issue. - The system continues its performance monitoring process.

Emergency Service Use Cases- Evelina Gace

UC Name:	**UC_ES_01 Real-time Patient Vitals Streaming**
Summary:	The system provides emergency responders with real-time streaming of patient vitals from wearable health devices, integrates with GPS for automatic ETA updates, and pre-notifies hospital staff about incoming cases.
Actors:	• **Primary Actor:** Emergency responders • **Secondary Actor:** Hospital staff
Preconditions:	-The emergency responder must be logged into the system with valid credentials. -Wearable health devices must be functional and paired with the system. -GPS tracking must be enabled.
Description of the Main Sequence:	1. The emergency responder logs into the system using secure credentials. 2. The system verifies the login details and grants access to the dashboard. 3. The responder selects the 'Real-time Patient Vitals' module from the dashboard. 4. The responder searches for the patient by entering their unique ID or scanning the wearable device pairing code. 5. The system retrieves the patient's details and confirms a connection with the wearable device. 6. The system starts receiving real-time vital signs from the patient's device (e.g., heart rate, blood pressure, oxygen levels). 7. The responder reviews the incoming vitals on their interface.

	8. The system streams the vitals in real-time to the emergency response team's dashboard. 9. The system calculates the estimated time of arrival (ETA) using GPS data and updates it dynamically. 10. The hospital staff receives a pre-notification with the patient's details, current vitals, and ETA. 11. The emergency responder continuously monitors the vitals and updates necessary records. 12. The system securely stores the patient's vitals and logs for future reference.
Description of the Alternative Sequence:	1. If the wearable device fails to transmit data, responders are alerted to check vitals and input them manually. 2. If GPS tracking is lost, responders manually update ETA and notify hospital staff.
Non-functional Requirements:	- Secure data transmission. - Must support low-latency data transmission.
Postconditions:	- Hospital staff receive the pre-notification with updated patient status. -Patient vitals are securely stored for reference.

UC Name:	**UC_ES_02 Digital Emergency Checklist**
Summary:	The system provides a digital checklist that guides responders based on patient conditions and enables automated report generation through voice commands and sensor data.
Actors:	• **Primary Actor: Emergency responders**
Preconditions:	-The emergency responder must be logged into the system with valid credentials. -The digital checklist system must be functional.

	-The system should have voice recognition enabled.
Description of the Main Sequence:	1. The emergency responder logs into the system using secure credentials. 2. The system verifies the credentials and grants access. 3. The responder navigates to the ‘Digital Emergency Checklist’ module. 4. The responder inputs or selects the patient's condition or symptoms. 5. The system customizes the checklist based on the patient’s symptoms and displays step-by-step guidance. 6. The responder follows the guided steps, checking off completed tasks. 7. If applicable, the system collects real-time sensor data to update the checklist dynamically. 8. The responder provides voice inputs for additional details (e.g., consciousness level, pain response). 9. The system records the voice input and automatically generates a report.

	10. The system securely saves the report and sends it to relevant hospital staff.
Description of the Alternative Sequence:	1. If voice recognition fails, responders manually input checklist data. 2. If automated report generation is not possible, responders complete reports manually.
Non-functional Requirements:	- The checklist must be accessible offline. - Support voice recognition. - Integrate with hospital records.
Postconditions:	-A completed emergency report is generated and available for medical staff review.

UC Name:	**UC_ES_03 Dynamic Ambulance Rerouting**
Summary:	The system allows ambulances to reroute dynamically based on real-time traffic conditions and enables automatic hospital bed availability checks.
Actors:	• **Primary Actor:** Emergency responder
Preconditions:	- The emergency responder must be logged into the system with valid credentials. -GPS tracking must be active. - Hospital databases must be accessible.
Description of the Main Sequence:	1. The emergency responder logs into the system using secure credentials.

	2. The system verifies the login and grants access. 3. The responder accesses the ‘Dynamic Ambulance Rerouting’ module. 4. The system continuously monitors real-time traffic conditions using GPS data. 5. The system suggests the fastest available route based on traffic updates. 6. The responder confirms the suggested route or selects an alternative if necessary. 7. The system checks hospital bed availability in the nearest facilities. 8. If the initially intended hospital has available capacity, the system confirms the route. 9. If the preferred hospital is full, the system automatically suggests the next best option. 10. The system reroutes the ambulance accordingly and updates ETA. 11. The hospital staff is notified about the incoming patient and estimated arrival time.

	12. The responder follows the optimized route while monitoring patient condition updates.
Description of the Alternative Sequence:	1. If GPS tracking fails, responders manually select a route. 2. If no nearby hospitals have available beds, the system suggests the next best option.
Non-functional Requirements:	- Must support real-time updates. - Ensure accurate data processing. - Provide high system reliability.