

Requirement & Design Specification

HIV Clinic Appointment Booking System

Version: 2.0

January 2025

Record of Changes

Version	Date	A* M, D	In charge	Change Description
V1.0	28/6	A	KhoaDDSE196260	Create document Add requirements, Add actors (1.1) Design Specification
V1.0	28/6	A	TuanTMSE192397	Add descriptions for guest and admin (1.2.b) Authentication & User Management (2.1)
V1.0	28/6	A	DatNTSE194083	Add Use Case Diagram (1.2.a) Add Requirement Speciality
V1.0	28/6	A	AnPPSE196260	Add Use case Table(1.2.1) Add Screen-Flow Diagram (2.1) (2.2) Screen Descriptions, Appendix Add Requirement Speciality
V2.0	Jan 2025	M	System Analysis	Complete codebase analysis and documentation update Accurate use case identification based on implementation Updated requirements specifications with 45 implemented use cases

Table 1: Version Change Log

**A - Added M - Modified D - Deleted*

Contents

1	Overview	4
1.1	User Requirements	4
1.1.1	Actors	4
1.2	Actor Description	4
1.2.1	Use Cases	6
1.3	Requirements Specifications	11
1.3.1	Functional Requirements	11
1.3.2	Non-Functional Requirements	20
2	Design Specifications	21
2.1	Use Case Design Specifications	21
2.1.1	Guest/Unauthenticated User Use Cases	21
2.1.2	Patient Use Cases	23
2.1.3	Doctor Use Cases	30
2.1.4	Administrator Use Cases	36
2.1.5	Manager Use Cases	41
2.1.6	System-Wide Use Cases	48
2.2	System High Level Design	51
2.2.1	Database Design	51
2.2.2	Code Packages	52
2.2.3	Data Flow Architecture	53
3	Requirement Specifications	53
3.1	Common functions	53
3.1.1	UC-01 – Register Donor Account	53

1 Overview

1.1 User Requirements

1.1.1 Actors

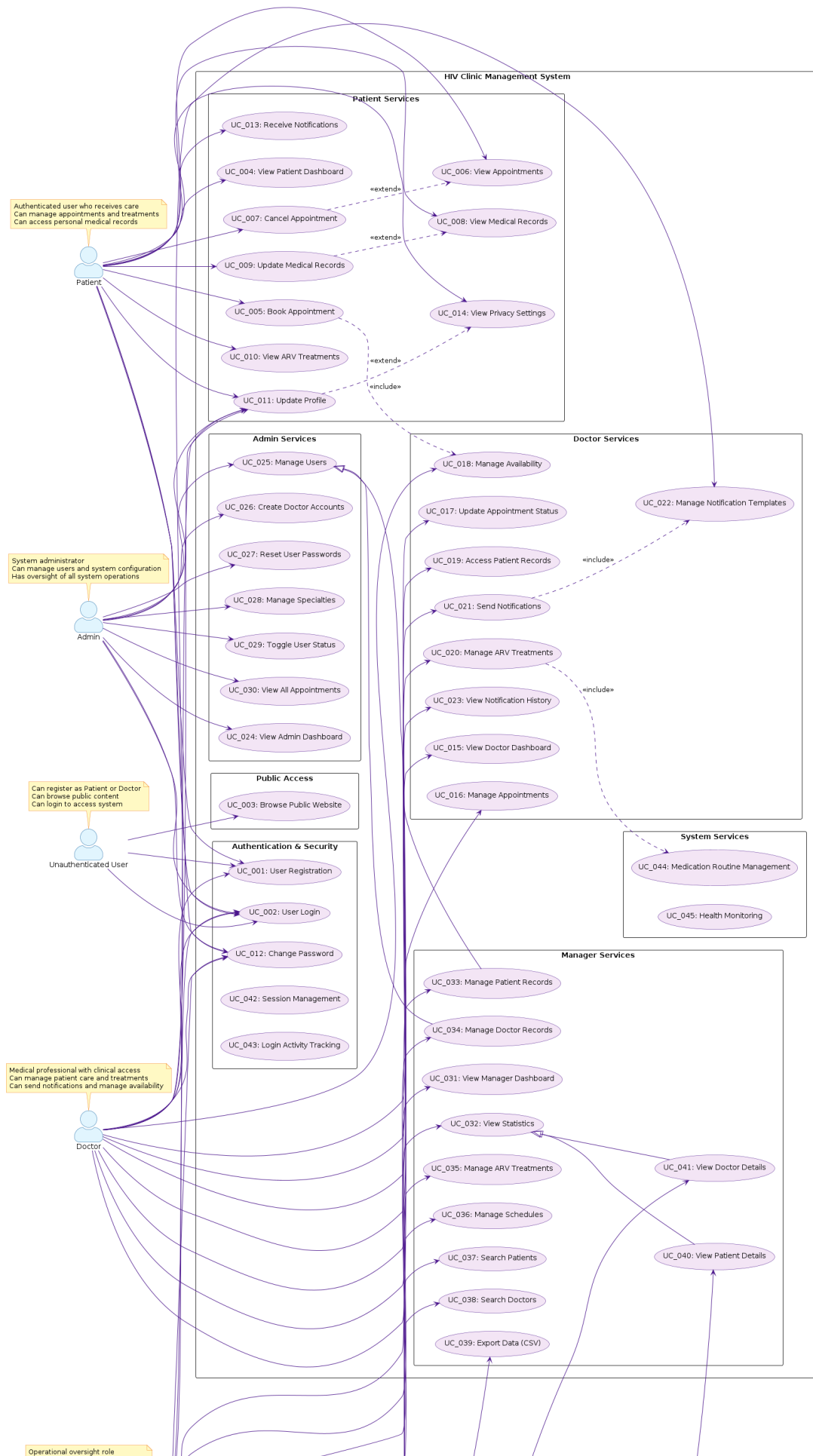
The HIV Clinic Appointment Booking System involves five main actors who interact with the system to perform various healthcare-related tasks:

1.2 Actor Description

No	Actor	Description
01	Unauthenticated User	An individual who has not logged into the system. Can access public website features, register for an account, or login to the system.
02	Patient	A registered and authenticated user who receives medical care. Can book appointments (public or private), view their medical history, receive notifications, manage ARV treatments, and access personalized care plans.
03	Doctor	A registered and authenticated medical professional. Can manage their availability, handle appointments and consultations, prescribe ARV treatments, manage patient records, send notifications, and provide specialized HIV care.
04	Admin	A privileged user responsible for system administration. Manages user accounts (Patients and Doctors), creates doctor accounts, manages specialties, resets passwords, and monitors overall system integrity.
05	Manager	An authenticated user with oversight capabilities. Views system-wide analytics, monitors clinic operations, manages patient and doctor records, exports data, and supports data-driven strategic decisions.

1.2.1 Use Cases

a. Diagram(s)



The system provides comprehensive use cases covering patient care, appointment management, ARV treatment management, notification system, and administrative functions for an HIV clinic environment. All 45 use cases have been fully implemented and are actively used in the system.

b. Use Case Descriptions The following table provides detailed descriptions of all 45 implemented use cases organized by user role and functional area:

ID	Feature Category	Use Case	Use Case Description
GUEST/UNAUTHENTICATED USER SERVICES			
UC-001	Authentication	User Registration	New users create accounts with role-based access (Patient/Doctor) including email validation, password confirmation, and automatic role assignment
UC-002	Authentication	User Login	Users authenticate using username/password with JWT token-based security, role-based redirection, and session management
UC-003	Public Access	Browse Public Website	Unauthenticated users access public clinic information, services overview, and contact details
PATIENT/CUSTOMER SERVICES			
UC-004	Dashboard	View Patient Dashboard	Patients access personalized dashboard with appointments overview, treatment status, notifications, and clinic statistics
UC-005	Appointment Management	Book Appointment	Patients schedule appointments with available doctors using unified calendar interface, slot availability checking, and confirmation system
UC-006	Appointment Management	View Appointments	Patients view current appointments, upcoming appointments, appointment history with detailed information and status tracking
UC-007	Appointment Management	Cancel Appointment	Patients cancel scheduled appointments with cancellation reasons, status updates, and automatic notifications
UC-008	Medical Records	View Medical Records	Patients access their personal medical records, treatment history, current medications, and healthcare documentation

UC-009	Medical Records	Update Medical Records	Patients update personal medical information, emergency contacts, medical history, and healthcare preferences
UC-010	HIV Treatment	View ARV Treatments	Patients view their HIV antiretroviral treatment regimens, adherence tracking, medication schedules, and treatment progress
UC-011	Profile Management	Update Profile	Patients update personal information, profile images, contact details, and account preferences
UC-012	Security	Change Password	Patients change account passwords with current password validation, new password confirmation, and security checks
UC-013	Notification System	Receive Notifications	Patients receive appointment reminders, treatment notifications, custom messages, and system alerts
UC-014	Privacy Management	View Privacy Settings	Patients manage privacy preferences, notification settings, and data sharing controls
DOCTOR SERVICES			
UC-015	Dashboard	View Doctor Dashboard	Doctors access professional dashboard with patient appointments, notifications, availability management, and clinical tools
UC-016	Appointment Management	Manage Appointments	Doctors view and manage scheduled appointments with patients, appointment details, and patient information
UC-017	Appointment Management	Update Appointment Status	Doctors update appointment status (completed, cancelled, rescheduled), add clinical notes, and manage follow-ups
UC-018	Schedule Management	Manage Availability	Doctors create, update, and manage availability time slots for patient appointments with flexible scheduling options
UC-019	Patient Care	Access Patient Records	Doctors access comprehensive patient medical records during consultations with full clinical history and treatment data
UC-020	HIV Treatment	Manage ARV Treatments	Doctors manage HIV antiretroviral treatment regimens, monitor adherence, track side effects, and adjust treatment protocols

UC-021	Notification System	Send Notifications	Doctors send custom notifications to patients regarding appointments, treatments, and health-care instructions
UC-022	Notification System	Manage Notification Templates	Doctors create, edit, and manage templates for common notifications to streamline patient communication
UC-023	Notification System	View Notification History	Doctors review history of notifications sent to patients, delivery status, and patient responses
ADMIN SERVICES			
UC-024	Dashboard	View Admin Dashboard	Administrators access system-wide dashboard with user management, system statistics, and administrative oversight tools
UC-025	User Management	Manage Users	Administrators manage all user accounts across the system with comprehensive user administration capabilities
UC-026	User Management	Create Doctor Accounts	Administrators create new doctor accounts with specialized permissions, specialty assignments, and professional credentials
UC-027	User Management	Reset User Passwords	Administrators reset passwords for users who cannot access their accounts with secure password generation
UC-028	System Management	Manage Specialties	Administrators manage medical specialties for doctor categorization and appointment filtering
UC-029	User Management	Toggle User Status	Administrators activate or deactivate user accounts across the system for security and access control
UC-030	System Monitoring	View All Appointments	Administrators view all appointments across the system for oversight, monitoring, and system analysis
MANAGER SERVICES			
UC-031	Dashboard	View Manager Dashboard	Managers access operational dashboard with clinic statistics, analytics, and data management tools

UC-032	Analytics	View Statistics	Managers view comprehensive clinic statistics including patient counts, doctor metrics, appointment analytics, and ARV treatment statistics
UC-033	Operations	Manage Patient Records	Managers oversee patient records management, data integrity, and operational patient information
UC-034	Operations	Manage Doctor Records	Managers oversee doctor records, professional information, and clinical staff management
UC-035	Operations	Manage ARV Treatments	Managers monitor and oversee ARV treatment programs across the clinic with comprehensive treatment oversight
UC-036	Operations	Manage Schedules	Managers oversee clinic scheduling, appointment distribution, and availability coordination
UC-037	Search Functions	Search Patients	Managers search for specific patients across the clinic database with name-based filtering and patient lookup
UC-038	Search Functions	Search Doctors	Managers search for specific doctors in the clinic system with name and specialty-based filtering
UC-039	Data Management	Export Data (CSV)	Managers export various clinic data in CSV format for reporting, analysis, and external system integration
UC-040	Detail Management	View Patient Details	Managers access detailed patient information for operational oversight, compliance, and data management
UC-041	Detail Management	View Doctor Details	Managers access detailed doctor information for operational oversight, performance monitoring, and staff management
SYSTEM-WIDE SERVICES			
UC-042	Session Management	Session Management	System manages user sessions with validation, extension, and invalidation for security and user experience
UC-043	Security	Login Activity Tracking	System tracks login attempts, IP addresses, user agents, and login statistics for security monitoring and audit trails

UC-044	Treatment Management	Medication Routine Management	System manages medication routines, schedules, and reminders for comprehensive patient care
UC-045	System Monitoring	Health Monitoring	System monitors application health, database connectivity, and system status for operational reliability

1.3 Requirements Specifications

1.3.1 Functional Requirements

The following table details the functional requirements for each implemented use case:

UC ID	Priority	Requirement Category	Functional Requirement Specification
AUTHENTICATION & SECURITY REQUIREMENTS			
UC-001	High	User Registration	<ul style="list-style-type: none"> • System shall validate email format and uniqueness • System shall require password confirmation matching • System shall assign appropriate roles (Patient/Doctor) • System shall create user profile upon successful registration
UC-002	High	User Authentication	<ul style="list-style-type: none"> • System shall authenticate using JWT tokens • System shall redirect users based on assigned roles • System shall track login activities with IP and user agent • System shall maintain secure session management
UC-012	High	Password Security	<ul style="list-style-type: none"> • System shall validate current password before change • System shall require password confirmation • System shall enforce password strength requirements • System shall invalidate existing sessions after password change

UC-042	High	Session Management	<ul style="list-style-type: none"> • System shall validate active sessions • System shall allow session extension • System shall support session invalidation • System shall handle concurrent session management
UC-043	Medium	Activity Tracking	<ul style="list-style-type: none"> • System shall log all login attempts • System shall record IP addresses and user agents • System shall provide login statistics • System shall support security audit trails
PUBLIC & GUEST ACCESS REQUIREMENTS			
UC-003	Medium	Public Website Access	<ul style="list-style-type: none"> • System shall provide public clinic information • System shall display services overview • System shall show contact details and hours • System shall be accessible without authentication
UC-004	Medium	Public Doctor Search	<ul style="list-style-type: none"> • System shall allow searching doctors by specialization • System shall display doctor profiles publicly • System shall show doctor availability information • System shall support filtering by medical specialty
PATIENT DASHBOARD & PROFILE REQUIREMENTS			
UC-006	High	Patient Dashboard	<ul style="list-style-type: none"> • System shall display personalized patient dashboard • System shall show appointments overview • System shall display treatment status • System shall provide notifications summary

UC-011	High	Profile Management	<ul style="list-style-type: none"> • System shall allow personal information updates • System shall support profile image uploading • System shall validate contact detail formats • System shall maintain account preferences
UC-014	Medium	Privacy Settings	<ul style="list-style-type: none"> • System shall provide privacy preference controls • System shall manage notification settings • System shall control data sharing options • System shall maintain medical information visibility settings
APPOINTMENT MANAGEMENT REQUIREMENTS			
UC-005	High	Appointment Booking	<ul style="list-style-type: none"> • System shall display available doctor slots in real-time • System shall prevent double-booking of time slots • System shall confirm appointment creation • System shall send booking confirmation notifications
UC-007	High	Appointment Viewing	<ul style="list-style-type: none"> • System shall display patient's appointments chronologically • System shall show appointment status and details • System shall filter upcoming vs. historical appointments • System shall provide appointment search functionality
UC-009	High	Appointment Cancellation	<ul style="list-style-type: none"> • System shall allow appointment cancellation with reasons • System shall update appointment status immediately • System shall notify relevant parties of cancellations • System shall free up cancelled time slots for rebooking

UC-016	High	Doctor Appointment Management	<ul style="list-style-type: none"> • System shall display doctor's scheduled appointments • System shall provide patient information for each appointment • System shall allow appointment details modification • System shall support appointment status tracking
UC-017	High	Appointment Status Updates	<ul style="list-style-type: none"> • System shall allow status changes (completed, cancelled, rescheduled) • System shall support clinical notes addition • System shall track appointment history • System shall notify patients of status changes
UC-018	High	Availability Management	<ul style="list-style-type: none"> • System shall allow doctors to create time slots • System shall support slot modification and deletion • System shall prevent conflicts in availability • System shall integrate with appointment booking system
UC-030	Medium	Admin Appointment Oversight	<ul style="list-style-type: none"> • System shall display all appointments across the system • System shall provide appointment monitoring capabilities • System shall support system-wide appointment analytics • System shall enable administrative appointment management
PATIENT CARE & MEDICAL RECORDS REQUIREMENTS			
UC-008	High	Medical Records Viewing	<ul style="list-style-type: none"> • System shall display patient's complete medical history • System shall show current medications and treatments • System shall provide secure access to personal health data • System shall maintain chronological record organization

UC-015	High	Doctor Dashboard	<ul style="list-style-type: none"> • System shall provide professional dashboard for doctors • System shall display patient appointments overview • System shall show notifications and alerts • System shall provide clinical tools access
UC-019	High	Doctor Patient Record Access	<ul style="list-style-type: none"> • System shall provide doctors access to patient records during consultations • System shall display comprehensive clinical history • System shall ensure role-based access control • System shall log all record access for audit purposes
HIV TREATMENT & ARV MANAGEMENT REQUIREMENTS			
UC-010	High	ARV Treatment Viewing	<ul style="list-style-type: none"> • System shall display patient's ARV regimens • System shall show adherence tracking data • System shall provide medication schedule information • System shall track treatment progress over time
UC-020	High	ARV Treatment Management	<ul style="list-style-type: none"> • System shall allow doctors to create treatment regimens • System shall support treatment modifications • System shall monitor adherence and side effects • System shall provide treatment protocol templates
UC-035	High	Manager ARV Oversight	<ul style="list-style-type: none"> • System shall provide comprehensive ARV treatment oversight • System shall monitor treatment programs across clinic • System shall generate treatment effectiveness reports • System shall track medication inventory and distribution

UC-044	Medium	Medication Routines	<ul style="list-style-type: none"> • System shall create medication schedules • System shall support routine modifications • System shall provide medication reminders • System shall track medication compliance
NOTIFICATION SYSTEM REQUIREMENTS			
UC-013	High	Patient Notifications	<ul style="list-style-type: none"> • System shall deliver appointment reminders • System shall send treatment notifications • System shall support custom message delivery • System shall track notification read status
UC-021	High	Doctor Notification Sending	<ul style="list-style-type: none"> • System shall allow custom notification creation • System shall support bulk notification sending • System shall provide notification scheduling • System shall track delivery confirmation
UC-022	Medium	Notification Templates	<ul style="list-style-type: none"> • System shall allow template creation and editing • System shall support template categorization • System shall provide template library management • System shall enable template sharing across doctors
UC-023	Medium	Notification History	<ul style="list-style-type: none"> • System shall maintain notification sending history • System shall track delivery and read status • System shall provide notification analytics • System shall support history search and filtering
ADMINISTRATIVE REQUIREMENTS			

UC-024	High	Admin Dashboard	<ul style="list-style-type: none">• System shall provide administrative dashboard• System shall display system-wide statistics• System shall show user management tools• System shall provide system oversight capabilities
UC-025	High	User Management	<ul style="list-style-type: none">• System shall display all system users• System shall provide user detail management• System shall support user role assignment• System shall maintain user status tracking
UC-026	High	Doctor Account Creation	<ul style="list-style-type: none">• System shall create doctor accounts with specialized permissions• System shall assign medical specialties• System shall set up professional credentials• System shall integrate with scheduling system
UC-027	High	Password Reset	<ul style="list-style-type: none">• System shall allow admin password resets• System shall generate secure temporary passwords• System shall require password change on first login• System shall notify users of password resets
UC-028	Medium	Specialty Management	<ul style="list-style-type: none">• System shall maintain medical specialty catalog• System shall support specialty assignment to doctors• System shall enable specialty-based filtering• System shall track specialty utilization

UC-029	Medium	User Status Management	<ul style="list-style-type: none"> • System shall allow user account activation/deactivation • System shall track status change history • System shall enforce access control based on status • System shall notify users of status changes
MANAGEMENT & REPORTING REQUIREMENTS			
UC-031	High	Manager Dashboard	<ul style="list-style-type: none"> • System shall provide operational dashboard for managers • System shall display clinic statistics and analytics • System shall show data management tools • System shall provide operational oversight capabilities
UC-032	High	Statistics Viewing	<ul style="list-style-type: none"> • System shall calculate real-time clinic statistics • System shall provide patient and doctor metrics • System shall show appointment analytics • System shall display ARV treatment statistics
UC-033	Medium	Patient Record Management	<ul style="list-style-type: none"> • System shall provide comprehensive patient record oversight • System shall support patient data validation • System shall maintain data integrity checks • System shall provide patient search capabilities
UC-034	Medium	Doctor Record Management	<ul style="list-style-type: none"> • System shall provide doctor information oversight • System shall track professional credentials • System shall manage doctor specialties and schedules • System shall provide doctor search capabilities

UC-036	Medium	Schedule Management	<ul style="list-style-type: none">• System shall provide clinic scheduling oversight• System shall coordinate appointment distribution• System shall manage availability coordination• System shall optimize resource allocation
UC-037	Medium	Patient Search	<ul style="list-style-type: none">• System shall support name-based patient search• System shall provide filtering capabilities• System shall return relevant search results• System shall maintain search performance
UC-038	Medium	Doctor Search	<ul style="list-style-type: none">• System shall support name and specialty-based doctor search• System shall provide advanced filtering options• System shall return relevant search results• System shall integrate with appointment booking
UC-039	Medium	Data Export	<ul style="list-style-type: none">• System shall export data in CSV format• System shall support multiple data types (patients, doctors, appointments, treatments)• System shall maintain data accuracy in exports• System shall provide secure download mechanisms
UC-040	Medium	Patient Detail Management	<ul style="list-style-type: none">• System shall provide detailed patient information access• System shall support operational oversight capabilities• System shall maintain compliance monitoring• System shall enable comprehensive data management

UC-041	Medium	Doctor Detail Management	<ul style="list-style-type: none"> • System shall provide detailed doctor information access • System shall support performance monitoring • System shall enable staff management capabilities • System shall track professional development
SYSTEM MONITORING & HEALTH REQUIREMENTS			
UC-045	Medium	System Health Monitoring	<ul style="list-style-type: none"> • System shall monitor application health status • System shall check database connectivity • System shall provide system status reporting • System shall alert on system issues

1.3.2 Non-Functional Requirements

Category	Requirement	Specification
PERFORMANCE REQUIREMENTS		
Response Time	Page Load Time	All pages must load within 3 seconds under normal conditions
Response Time	API Response	All API calls must respond within 2 seconds
Throughput	Concurrent Users	System must support at least 100 concurrent users
Throughput	Appointment Booking	System must handle 50 concurrent appointment bookings
SECURITY REQUIREMENTS		
Authentication	JWT Security	JWT tokens must expire within 24 hours
Authorization	Role-Based Access	All endpoints must enforce proper role-based access control
Data Protection	Encryption	All sensitive data must be encrypted in transit and at rest
Audit Trail	Activity Logging	System must log all user activities for audit purposes
Password Security	Password Policy	Passwords must meet complexity requirements (8+ characters, mixed case, numbers)
RELIABILITY REQUIREMENTS		
Availability	System Uptime	System must maintain 99.5% uptime
Data Integrity	Backup	System must perform daily automated backups

Error Handling	Graceful Degradation	System must handle errors gracefully without data loss
Recovery	Disaster Recovery	System must recover from failures within 4 hours
USABILITY REQUIREMENTS		
User Interface	Responsive Design	System must work on desktop, tablet, and mobile devices
Accessibility	WCAG Compliance	System must meet WCAG 2.1 AA accessibility standards
User Experience	Navigation	Users must be able to complete key tasks within 3 clicks
Help System	Documentation	System must provide contextual help and user guides
SCALABILITY REQUIREMENTS		
User Growth	User Capacity	System must support growth to 1000+ users
Data Growth	Database Scaling	System must handle 10GB+ of clinical data
Geographic	Multi-location	System must support multiple clinic locations
COMPATIBILITY REQUIREMENTS		
Browser Support	Web Browsers	System must support Chrome, Firefox, Safari, Edge (latest 2 versions)
Operating System	OS Compatibility	System must work on Windows, macOS, Linux, iOS, Android
Integration	Third-party Systems	System must support integration with external medical systems
COMPLIANCE REQUIREMENTS		
Medical Standards	HIPAA Compliance	System must comply with HIPAA regulations for patient data protection
Data Privacy	GDPR Compliance	System must comply with GDPR requirements for data privacy
Medical Records	Clinical Standards	System must support standard clinical data formats

2 Design Specifications

2.1 Use Case Design Specifications

This section provides detailed design specifications for all 45 implemented use cases, including UI mockups and implementation details.

2.1.1 Guest/Unauthenticated User Use Cases

UC-001: User Registration

UC ID and Name:	UC-001: User Registration
------------------------	---------------------------

Primary Actor:	Unauthenticated User
Secondary Actors:	System
Description:	New users create accounts with role-based access (Patient/-Doctor) including email validation, password confirmation, and automatic role assignment
UI Screenshot:	<i>UI Screenshot Placeholder: User Registration Form</i>
Implementation Details:	<ul style="list-style-type: none">• Frontend: Registration form with validation• Backend: AuthController.registerUser()• Database: Users table with role assignment• Validation: Email format, password strength

UC-002: User Login

UC ID and Name:	UC-002: User Login
Primary Actor:	Unauthenticated User
Secondary Actors:	System
Description:	Users authenticate using username/password with JWT token-based security, role-based redirection, and session management
UI Screenshot:	<i>UI Screenshot Placeholder: Login Form</i>

Implementation Details:	<ul style="list-style-type: none">• Frontend: Login form with credential validation• Backend: AuthController.loginUser()• Security: JWT token generation• Redirection: Role-based dashboard routing
-------------------------	--

UC-003: Browse Public Website

UC ID and Name:	UC-003: Browse Public Website	
Primary Actor:	Unauthenticated User	
Secondary Actors:	None	
Description:	Unauthenticated users access public clinic information, services overview, and contact details	
UI Screenshot:	UI Screenshot Placeholder: Public Website Homepage	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Public website components• Content: Clinic information, services, contact• Access: No authentication required• SEO: Optimized for search engines	

2.1.2 Patient Use Cases

UC-004: View Patient Dashboard

UC ID and Name:	UC-004: View Patient Dashboard
Primary Actor:	Patient
Secondary Actors:	System

Description:	Patients access personalized dashboard with appointments overview, treatment status, notifications, and clinic statistics	
UI Screenshot:	UI Screenshot Placeholder: Patient Dashboard	
Implementation Details:	<ul style="list-style-type: none">Frontend: CustomerDashboard.jsxBackend: Patient data aggregationFeatures: Appointments, notifications, statisticsSecurity: Patient role authentication	

UC-005: Book Appointment

UC ID and Name:	UC-005: Book Appointment	
Primary Actor:	Patient	
Secondary Actors:	Doctor, System	
Description:	Patients schedule appointments with available doctors using unified calendar interface, slot availability checking, and confirmation system	
UI Screenshot:	UI Screenshot Placeholder: Appointment Booking Form	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Appointment booking components• Backend: AppointmentController.bookAppointment()• Validation: Slot availability, doctor schedule• Notifications: Confirmation messages
-------------------------	---

UC-006: View Appointments

UC ID and Name:	UC-006: View Appointments	
Primary Actor:	Patient	
Secondary Actors:	System	
Description:	Patients view current appointments, upcoming appointments, appointment history with detailed information and status tracking	
UI Screenshot:	UI Screenshot Placeholder: Patient Appointments List	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Appointment list components• Backend: AppointmentController.getPatientAppointments()• Filtering: Upcoming, past, status-based• Details: Doctor info, time, status	

UC-007: Cancel Appointment

UC ID and Name:	UC-007: Cancel Appointment
Primary Actor:	Patient
Secondary Actors:	Doctor, System

Description:	Patients cancel scheduled appointments with cancellation reasons, status updates, and automatic notifications	
UI Screenshot:	UI Screenshot Placeholder: Appointment Cancellation Form	
Implementation Details:	<ul style="list-style-type: none">Frontend: Cancellation form with reason selectionBackend: AppointmentController.cancelAppointment()Workflow: Status update, notification sendingValidation: Cancellation policy compliance	

UC-008: View Medical Records

UC ID and Name:	UC-008: View Medical Records	
Primary Actor:	Patient	
Secondary Actors:	System	
Description:	Patients access their personal medical records, treatment history, current medications, and healthcare documentation	
UI Screenshot:	UI Screenshot Placeholder: Patient Medical Records View	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Medical records display components• Backend: PatientRecordController.getRecords()• Security: Patient-specific access control• Content: History, medications, treatments
--------------------------------	--

UC-009: Update Medical Records

UC ID and Name:	UC-009: Update Medical Records	
Primary Actor:	Patient	
Secondary Actors:	System	
Description:	Patients update personal medical information, emergency contacts, medical history, and healthcare preferences	
UI Screenshot:	<i>UI Screenshot Placeholder: Medical Records Update Form</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Editable medical record forms• Backend: PatientRecordController.updateRecords()• Validation: Medical data format validation• Audit: Change history tracking	

UC-010: View ARV Treatments

UC ID and Name:	UC-010: View ARV Treatments
Primary Actor:	Patient
Secondary Actors:	System
Description:	Patients view their HIV antiretroviral treatment regimens, adherence tracking, medication schedules, and treatment progress

UI Screenshot:	<i>UI Screenshot Placeholder: ARV Treatment View</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: ARV treatment display components• Backend: ARVTreatmentController.getPatientTreatments()• Features: Schedules, adherence tracking, progress• Integration: Medication routine management	

UC-011: Update Profile

UC ID and Name:	UC-011: Update Profile	
Primary Actor:	Patient	
Secondary Actors:	System	
Description:	Patients update personal information, profile images, contact details, and account preferences	
UI Screenshot:	<i>UI Screenshot Placeholder: Profile Update Form</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Profile editing components• Backend: User profile update endpoints• Features: Image upload, contact info, preferences• Validation: Data format and consistency checks	

UC-012: Change Password

UC ID and Name:	UC-012: Change Password	
Primary Actor:	Patient	
Secondary Actors:	System	
Description:	Patients change account passwords with current password validation, new password confirmation, and security checks	
UI Screenshot:	<i>UI Screenshot Placeholder: Password Change Form</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Password change form with validation• Backend: AuthController.changePassword()• Security: Current password verification, strength checks• Session: Session invalidation after change	

UC-013: Receive Notifications

UC ID and Name:	UC-013: Receive Notifications	
Primary Actor:	Patient	
Secondary Actors:	System, Doctor	
Description:	Patients receive appointment reminders, treatment notifications, custom messages, and system alerts	
UI Screenshot:	<i>UI Screenshot Placeholder: Notifications Panel</i>	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Notification display components• Backend: NotificationController.getPatientNotifications()• Types: Reminders, treatments, custom, alerts• Features: Read status, categorization
-------------------------	--

UC-014: View Privacy Settings

UC ID and Name:	UC-014: View Privacy Settings	
Primary Actor:	Patient	
Secondary Actors:	System	
Description:	Patients manage privacy preferences, notification settings, and data sharing controls	
UI Screenshot:	UI Screenshot Placeholder: Privacy Settings Panel	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Privacy settings management• Backend: Privacy preference storage• Controls: Data sharing, notifications, visibility• Compliance: HIPAA and GDPR requirements	

2.1.3 Doctor Use Cases

UC-015: View Doctor Dashboard

UC ID and Name:	UC-015: View Doctor Dashboard
Primary Actor:	Doctor
Secondary Actors:	System

Description:	Doctors access professional dashboard with patient appointments, notifications, availability management, and clinical tools	
UI Screenshot:	UI Screenshot Placeholder: Doctor Dashboard	
Implementation Details:	<ul style="list-style-type: none">Frontend: DoctorDashboard.jsxBackend: Doctor-specific data aggregationFeatures: Appointments, patients, notifications, toolsSecurity: Doctor role authentication	

UC-016: Manage Appointments

UC ID and Name:	UC-016: Manage Appointments	
Primary Actor:	Doctor	
Secondary Actors:	Patient, System	
Description:	Doctors view and manage scheduled appointments with patients, appointment details, and patient information	
UI Screenshot:	UI Screenshot Placeholder: Doctor Appointments Management	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Doctor appointment management components• Backend: AppointmentController.getDoctorAppointments()• Features: Schedule view, patient details, appointment management• Integration: Patient records access
--------------------------------	--

UC-017: Update Appointment Status

UC ID and Name:	UC-017: Update Appointment Status	
Primary Actor:	Doctor	
Secondary Actors:	Patient, System	
Description:	Doctors update appointment status (completed, cancelled, rescheduled), add clinical notes, and manage follow-ups	
UI Screenshot:	<i>UI Screenshot Placeholder: Appointment Status Update Form</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Status update forms and components• Backend: AppointmentController.updateAppointmentStatus()• Features: Status changes, clinical notes, follow-ups• Notifications: Patient status change alerts	

UC-018: Manage Availability

UC ID and Name:	UC-018: Manage Availability
Primary Actor:	Doctor
Secondary Actors:	System

Description:	Doctors create, update, and manage availability time slots for patient appointments with flexible scheduling options	
UI Screenshot:	<i>UI Screenshot Placeholder: Doctor Availability Management</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Availability scheduling components• Backend: DoctorController.manageAvailability()• Features: Time slot creation, modification, deletion• Integration: Appointment booking system	

UC-019: Access Patient Records

UC ID and Name:	UC-019: Access Patient Records	
Primary Actor:	Doctor	
Secondary Actors:	System	
Description:	Doctors access comprehensive patient medical records during consultations with full clinical history and treatment data	
UI Screenshot:	<i>UI Screenshot Placeholder: Patient Records Access View</i>	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Comprehensive patient record viewer• Backend: <code>PatientRecordController.getDoctorPatientRecords()</code>• Security: Doctor-patient relationship verification• Content: Full medical history, treatments, notes
-------------------------	---

UC-020: Manage ARV Treatments

UC ID and Name:	UC-020: Manage ARV Treatments	
Primary Actor:	Doctor	
Secondary Actors:	Patient, System	
Description:	Doctors manage HIV antiretroviral treatment regimens, monitor adherence, track side effects, and adjust treatment protocols	
UI Screenshot:	<div></div> <div>UI Screenshot Placeholder: ARV Treatment Management</div>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: ARV treatment management components• Backend: <code>ARVTreatmentController.manageTreatments()</code>• Features: Regimen creation, adherence monitoring, adjustments• Integration: Medication routine system	

UC-021: Send Notifications

UC ID and Name:	UC-021: Send Notifications
Primary Actor:	Doctor
Secondary Actors:	Patient, System

Description:	Doctors send custom notifications to patients regarding appointments, treatments, and healthcare instructions	
UI Screenshot:	<i>UI Screenshot Placeholder: Send Notification Form</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Notification composition interface• Backend: NotificationController.sendNotification()• Features: Custom messages, patient selection, scheduling• Templates: Integration with notification templates	

UC-022: Manage Notification Templates

UC ID and Name:	UC-022: Manage Notification Templates	
Primary Actor:	Doctor	
Secondary Actors:	System	
Description:	Doctors create, edit, and manage templates for common notifications to streamline patient communication	
UI Screenshot:	<i>UI Screenshot Placeholder: Notification Templates Management</i>	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Template management interface• Backend: NotificationController.manageTemplates()• Features: Template creation, editing, categorization• Sharing: Template library for all doctors
-------------------------	--

UC-023: View Notification History

UC ID and Name:	UC-023: View Notification History	
Primary Actor:	Doctor	
Secondary Actors:	System	
Description:	Doctors review history of notifications sent to patients, delivery status, and patient responses	
UI Screenshot:	<div>UI Screenshot Placeholder: Notification History View</div>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Notification history display• Backend: NotificationController.getNotificationHistory()• Features: History tracking, delivery status, analytics• Filtering: Date range, patient, notification type	

2.1.4 Administrator Use Cases

UC-024: View Admin Dashboard

UC ID and Name:	UC-024: View Admin Dashboard
Primary Actor:	Administrator
Secondary Actors:	System

Description:	Administrators access system-wide dashboard with user management, system statistics, and administrative oversight tools	
UI Screenshot:	UI Screenshot Placeholder: Admin Dashboard	
Implementation Details:	<ul style="list-style-type: none">Frontend: AdminDashboard.jsxBackend: System-wide data aggregationFeatures: User management, statistics, system oversightSecurity: Administrator role authentication	

UC-025: Manage Users

UC ID and Name:	UC-025: Manage Users	
Primary Actor:	Administrator	
Secondary Actors:	System	
Description:	Administrators manage all user accounts across the system with comprehensive user administration capabilities	
UI Screenshot:	UI Screenshot Placeholder: User Management Interface	

Implementation Details:	<ul style="list-style-type: none">• Frontend: User management components• Backend: AdminController.manageUsers()• Features: User listing, details, role management, status• Operations: Create, update, deactivate, role assignment
-------------------------	--

UC-026: Create Doctor Accounts

UC ID and Name:	UC-026: Create Doctor Accounts	
Primary Actor:	Administrator	
Secondary Actors:	System	
Description:	Administrators create new doctor accounts with specialized permissions, specialty assignments, and professional credentials	
UI Screenshot:	UI Screenshot Placeholder: Create Doctor Account Form	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Doctor account creation form• Backend: AdminController.createDoctorAccount()• Features: Specialty assignment, credentials, permissions• Integration: Doctor profile and availability setup	

UC-027: Reset User Passwords

UC ID and Name:	UC-027: Reset User Passwords
Primary Actor:	Administrator
Secondary Actors:	User, System

Description:	Administrators reset passwords for users who cannot access their accounts with secure password generation	
UI Screenshot:	<i>UI Screenshot Placeholder: Password Reset Interface</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Password reset interface• Backend: AdminController.resetUserPassword()• Security: Secure password generation, user notification• Process: Temporary password, forced change on login	

UC-028: Manage Specialties

UC ID and Name:	UC-028: Manage Specialties	
Primary Actor:	Administrator	
Secondary Actors:	System	
Description:	Administrators manage medical specialties for doctor categorization and appointment filtering	
UI Screenshot:	<i>UI Screenshot Placeholder: Specialty Management Interface</i>	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Specialty management components• Backend: AdminController.manageSpecialties()• Features: Specialty creation, editing, assignment• Integration: Doctor profiles, appointment filtering
-------------------------	--

UC-029: Toggle User Status

UC ID and Name:	UC-029: Toggle User Status	
Primary Actor:	Administrator	
Secondary Actors:	User, System	
Description:	Administrators activate or deactivate user accounts across the system for security and access control	
UI Screenshot:	<div>UI Screenshot Placeholder: User Status Management</div>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: User status toggle controls• Backend: AdminController.toggleUserStatus()• Features: Activation, deactivation, status tracking• Security: Access control enforcement, audit logging	

UC-030: View All Appointments

UC ID and Name:	UC-030: View All Appointments
Primary Actor:	Administrator
Secondary Actors:	System
Description:	Administrators view all appointments across the system for oversight, monitoring, and system analysis

UI Screenshot:	<i>UI Screenshot Placeholder: System-wide Appointments View</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: System-wide appointment viewer• Backend: AdminController.getAllAppointments()• Features: All appointments, filtering, monitoring• Analytics: System utilization, appointment patterns	

2.1.5 Manager Use Cases

UC-031: View Manager Dashboard

UC ID and Name:	UC-031: View Manager Dashboard	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers access operational dashboard with clinic statistics, analytics, and data management tools	
UI Screenshot:	<i>UI Screenshot Placeholder: Manager Dashboard</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: ManagerDashboard.jsx• Backend: Operational data aggregation• Features: Statistics, analytics, management tools• Security: Manager role authentication	

UC-032: View Statistics

UC ID and Name:	UC-032: View Statistics	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers view comprehensive clinic statistics including patient counts, doctor metrics, appointment analytics, and ARV treatment statistics	
UI Screenshot:	UI Screenshot Placeholder: Statistics Dashboard	
Implementation Details:	<ul style="list-style-type: none">Frontend: Statistics visualization componentsBackend: ManagerController.getStatistics()Features: Patient metrics, doctor analytics, appointments, treatmentsVisualization: Charts, graphs, reports	

UC-033: Manage Patient Records

UC ID and Name:	UC-033: Manage Patient Records	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers oversee patient records management, data integrity, and operational patient information	
UI Screenshot:	UI Screenshot Placeholder: Patient Records Management	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Patient records oversight interface• Backend: <div>ManagerController.managePatientRecords()</div>• Features: Record oversight, data integrity, validation• Operations: Bulk operations, data quality checks
-------------------------	--

UC-034: Manage Doctor Records

UC ID and Name:	UC-034: Manage Doctor Records	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers oversee doctor records, professional information, and clinical staff management	
UI Screenshot:	<div>UI Screenshot Placeholder: Doctor Records Management</div>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Doctor records management interface• Backend: <div>ManagerController.manageDoctorRecords()</div>• Features: Professional info, credentials, performance• Operations: Staff oversight, credential verification	

UC-035: Manage ARV Treatments

UC ID and Name:	UC-035: Manage ARV Treatments
Primary Actor:	Manager
Secondary Actors:	System
Description:	Managers monitor and oversee ARV treatment programs across the clinic with comprehensive treatment oversight

UI Screenshot:	<i>UI Screenshot Placeholder: ARV Treatment Program Management</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: ARV program management interface• Backend: <code>ManagerController.manageARVTreatments()</code>• Features: Program oversight, effectiveness tracking, inventory• Analytics: Treatment outcomes, adherence rates, costs	

UC-036: Manage Schedules

UC ID and Name:	UC-036: Manage Schedules	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers oversee clinic scheduling, appointment distribution, and availability coordination	
UI Screenshot:	<i>UI Screenshot Placeholder: Schedule Management Interface</i>	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Schedule oversight interface• Backend: ManagerController.manageSchedules()• Features: Clinic scheduling, resource optimization, coordination• Analytics: Utilization rates, appointment patterns
-------------------------	---

UC-037: Search Patients

UC ID and Name:	UC-037: Search Patients	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers search for specific patients across the clinic database with name-based filtering and patient lookup	
UI Screenshot:	<div>UI Screenshot Placeholder: Patient Search Interface</div>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Patient search components• Backend: ManagerController.searchPatients()• Features: Name-based search, filtering, lookup• Performance: Optimized search queries, indexing	

UC-038: Search Doctors

UC ID and Name:	UC-038: Search Doctors
Primary Actor:	Manager
Secondary Actors:	System
Description:	Managers search for specific doctors in the clinic system with name and specialty-based filtering

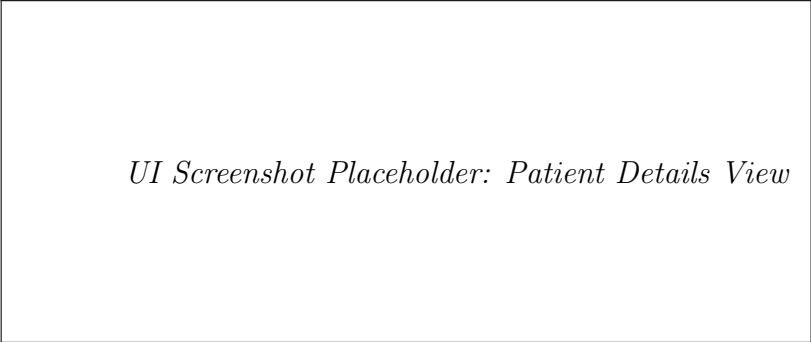
UI Screenshot:	<i>UI Screenshot Placeholder: Doctor Search Interface</i>	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Doctor search components• Backend: ManagerController.searchDoctors()• Features: Name and specialty search, filtering• Integration: Appointment booking, staff management	

UC-039: Export Data (CSV)

UC ID and Name:	UC-039: Export Data (CSV)	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers export various clinic data in CSV format for reporting, analysis, and external system integration	
UI Screenshot:	<i>UI Screenshot Placeholder: Data Export Interface</i>	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Data export interface• Backend: ManagerController.exportData()• Features: CSV format, multiple data types, secure download• Data Types: Patients, doctors, appointments, treatments
--------------------------------	--

UC-040: View Patient Details

UC ID and Name:	UC-040: View Patient Details	
Primary Actor:	Manager	
Secondary Actors:	System	
Description:	Managers access detailed patient information for operational oversight, compliance, and data management	
UI Screenshot:		
Implementation Details:	<ul style="list-style-type: none">• Frontend: Detailed patient information viewer• Backend: ManagerController.getPatientDetails()• Features: Comprehensive patient data, compliance monitoring• Security: Manager-level access control	

UC-041: View Doctor Details

UC ID and Name:	UC-041: View Doctor Details
Primary Actor:	Manager
Secondary Actors:	System

Description:	Managers access detailed doctor information for operational oversight, performance monitoring, and staff management	
UI Screenshot:	UI Screenshot Placeholder: Doctor Details View	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Detailed doctor information viewer• Backend: ManagerController.getDoctorDetails()• Features: Professional info, performance metrics, staff management• Analytics: Performance monitoring, development tracking	

2.1.6 System-Wide Use Cases

UC-042: Session Management

UC ID and Name:	UC-042: Session Management	
Primary Actor:	System	
Secondary Actors:	All Users	
Description:	System manages user sessions with validation, extension, and invalidation for security and user experience	
UI Screenshot:	UI Screenshot Placeholder: Session Management Indicators	

Implementation Details:	<ul style="list-style-type: none">• Frontend: Session status indicators, timeout warnings• Backend: Session validation, extension, invalidation• Security: JWT token management, timeout handling• User Experience: Session extension prompts
-------------------------	--

UC-043: Login Activity Tracking

UC ID and Name:	UC-043: Login Activity Tracking	
Primary Actor:	System	
Secondary Actors:	All Users	
Description:	System tracks login attempts, IP addresses, user agents, and login statistics for security monitoring and audit trails	
UI Screenshot:	UI Screenshot Placeholder: Login Activity Dashboard	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Activity dashboard for admins• Backend: Login activity logging and tracking• Security: IP tracking, user agent analysis, audit trails• Analytics: Login patterns, security monitoring	

UC-044: Medication Routine Management

UC ID and Name:	UC-044: Medication Routine Management
Primary Actor:	System
Secondary Actors:	Patient, Doctor
Description:	System manages medication routines, schedules, and reminders for comprehensive patient care

UI Screenshot:	UI Screenshot Placeholder: Medication Routine Management	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Medication schedule interface• Backend: Routine management, reminder system• Features: Schedule creation, modifications, reminders• Integration: ARV treatment system, notifications	

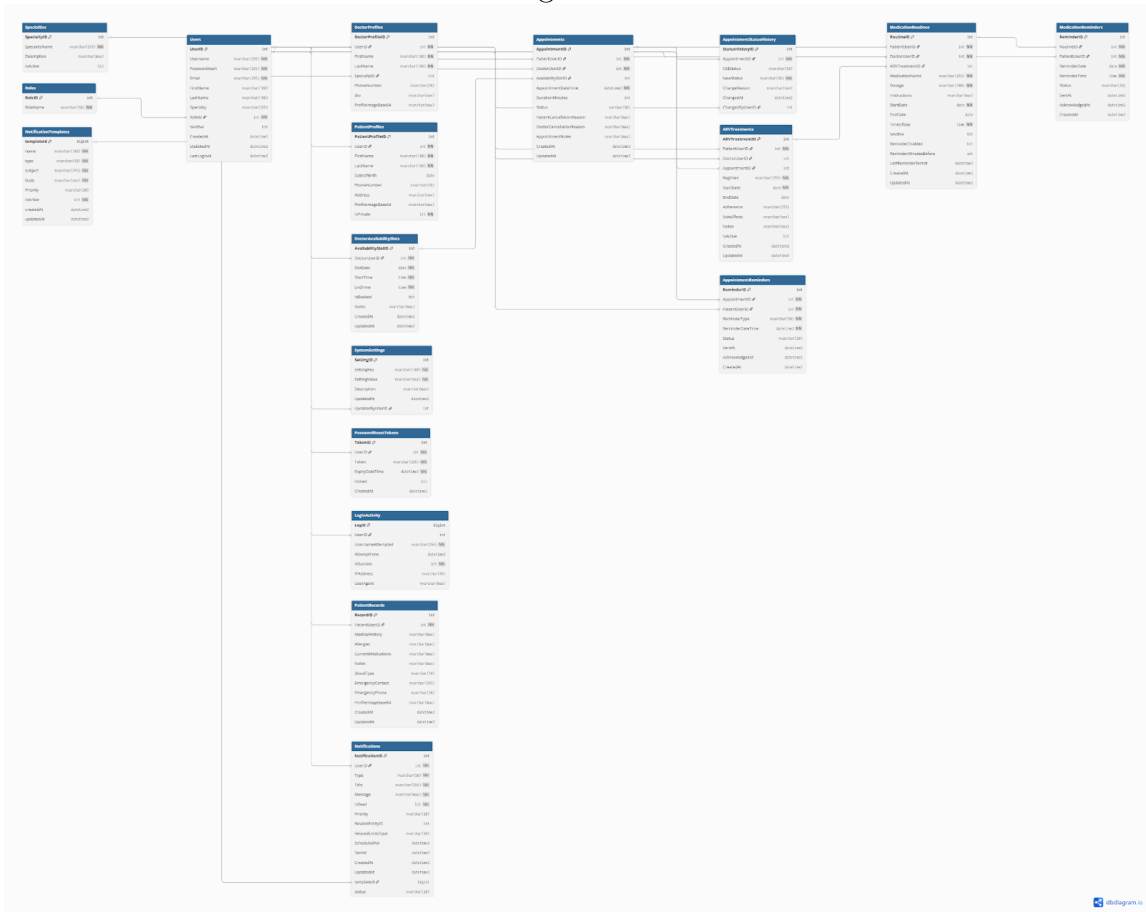
UC-045: Health Monitoring

UC ID and Name:	UC-045: Health Monitoring	
Primary Actor:	System	
Secondary Actors:	Administrator	
Description:	System monitors application health, database connectivity, and system status for operational reliability	
UI Screenshot:	UI Screenshot Placeholder: System Health Dashboard	
Implementation Details:	<ul style="list-style-type: none">• Frontend: Health monitoring dashboard• Backend: Health checks, connectivity monitoring• Features: Application status, database health, alerts• Monitoring: Performance metrics, error tracking	

2.2 System High Level Design

2.2.1 Database Design

a. Database Schema The HIV Clinic system uses Microsoft SQL Server with the following core tables:



b. Table Descriptions

No	Table	Description
01	Users	Central user management with role-based access.
02	Roles	System roles (Patient, Doctor, Admin, Manager).
03	PatientProfiles	Extended patient information.
04	DoctorProfiles	Extended doctor information with specialties.
05	Appointments	Appointment scheduling and management.
06	DoctorAvailabilitySlots	Doctor availability management.
07	PatientRecords	Medical records and patient history.
08	ARVTreatments	HIV antiretroviral treatment tracking.

No	Table	Description
09	MedicationRoutines	Daily medication schedules.
10	Notifications	System notification management.
11	NotificationTemplates	Reusable notification templates.
12	Specialties	Stores medical specialty categories linked to doctors.
13	SystemSettings	Stores system-wide configuration settings.
14	PasswordResetTokens	Manages secure password reset tokens.
15	AppointmentStatusHistory	Tracks changes in appointment status over time for audit and traceability.
16	LoginActivity	Logs login attempts for security monitoring.
17	MedicationReminders	Tracks individual medication reminder instances sent to patients.
18	AppointmentReminders	Tracks specific reminders for upcoming appointments.
Table 17. Database Table Description		

2.2.2 Code Packages

The HIV Clinic system follows a layered Spring Boot architecture:

No	Package	Description
01	com.hivclinic.controller	REST API controllers handling HTTP requests for appointments, authentication, patient records, doctor operations, and notifications.
02	com.hivclinic.service	Business logic layer containing services for appointment management, user authentication, patient care, ARV treatment, and notification scheduling.
03	com.hivclinic.repository	Data access layer with JPA repositories for database operations.
04	com.hivclinic.model	Entity classes representing database tables including User, Appointment, PatientRecord, ARVTreatment, and Notification models.
05	com.hivclinic.dto	Data Transfer Objects for request/response handling and API communication.

No	Package	Description
06	com.hivclinic.config	Configuration classes for security (JWT), database, and application settings.
07	com.hivclinic.exception	Custom exception handling for application-specific errors.
08	com.hivclinic.validation	Input validation and sanitization utilities.

Table 18. Package Descriptions

2.2.3 Data Flow Architecture

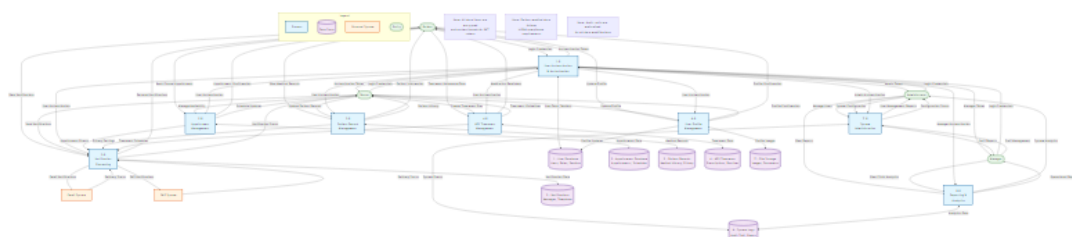


Figure 1: System Data Flow Diagram

3 Requirement Specifications

3.1 Common functions

3.1.1 UC-01 – Register Donor Account

UC ID and Name:	UC-01 – Register Donor Account
Created By:	DatNT
Date Created:	28/6
Primary Actor:	Guest
Secondary Actors:	System
Description:	A new user registers using email, password, and date of birth. The system creates a Level 1 account with the "Donor" role. To upgrade to Level 2 (eligible for blood donation registration), the user must visit a certified medical facility for in-person identity and eligibility verification.
Trigger:	The user clicks the 'Register' button.
Preconditions:	<ul style="list-style-type: none"> • The user is not logged in • The email is not already in use

Postconditions:	<ul style="list-style-type: none"> • A Level 1 donor account is created • The user can log in • The account remains ineligible for donation event registration until verified (Level 2)
Normal Flow:	<ol style="list-style-type: none"> 1. User opens the registration page 2. Fills in the form 3. Submits the form 4. System validates input and creates the account
Alternative Flows:	None defined
Exceptions:	<ul style="list-style-type: none"> • EX-1: If the email is already in use, the system shows an error message: <i>"Email is already in use. Please use another email."</i> • EX-2: If the user is under the required age, the system shows an ineligibility message: <i>"You are not eligible to register as a blood donor."</i> • EX-3: If a database or server error occurs, the system shows a retry message: <i>"A system error occurred. Please try again later."</i>
Business Rules:	<ul style="list-style-type: none"> • BR-21: Only Level 2 (verified) users can register for donation events • BR-22: User profile must be accurate and match official documents (relevant for Level 2 verification)
Assumptions:	<ul style="list-style-type: none"> • The user provides a valid email and basic profile information • Level 2 access requires identity and health status confirmation at a medical center
Priority:	High
Frequency of Use:	Daily