

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 | 41 |
| 1.4 | System High Level Design | 42 |
| 1.4.1 | Database Design | 42 |
| 1.4.2 | Code Packages | 44 |
| 1.4.3 | Data Flow Architecture | 45 |
| 2 | Requirement Specifications | 45 |
| 2.1 | Common functions | 45 |
| 2.1.1 | UC-01 – Register Donor Account | 45 |
| 2.1.2 | UC-02 – Log In | 47 |
| 2.1.3 | UC-03 – View Hospital Information | 48 |
| 2.1.4 | UC-04 – Read Blogs | 49 |
| 2.1.5 | UC-05 – Manage Personal Profile | 50 |
| 3 | Design Specifications | 51 |
| 3.1 | Authentication System | 51 |
| 3.1.1 | User Login | 51 |
| 3.2 | Appointment Management | 52 |
| 3.2.1 | Appointment Booking | 52 |
| 3.3 | Patient Care System | 53 |
| 3.3.1 | Patient Records Management | 53 |
| 4 | Appendix | 55 |
| 4.1 | Assumptions & Dependencies | 55 |
| 4.2 | Limitations & Exclusions | 55 |
| 4.3 | Business Rules | 56 |
| 4.4 | Technical Specifications | 56 |

1 Overview

1.1 User Requirements

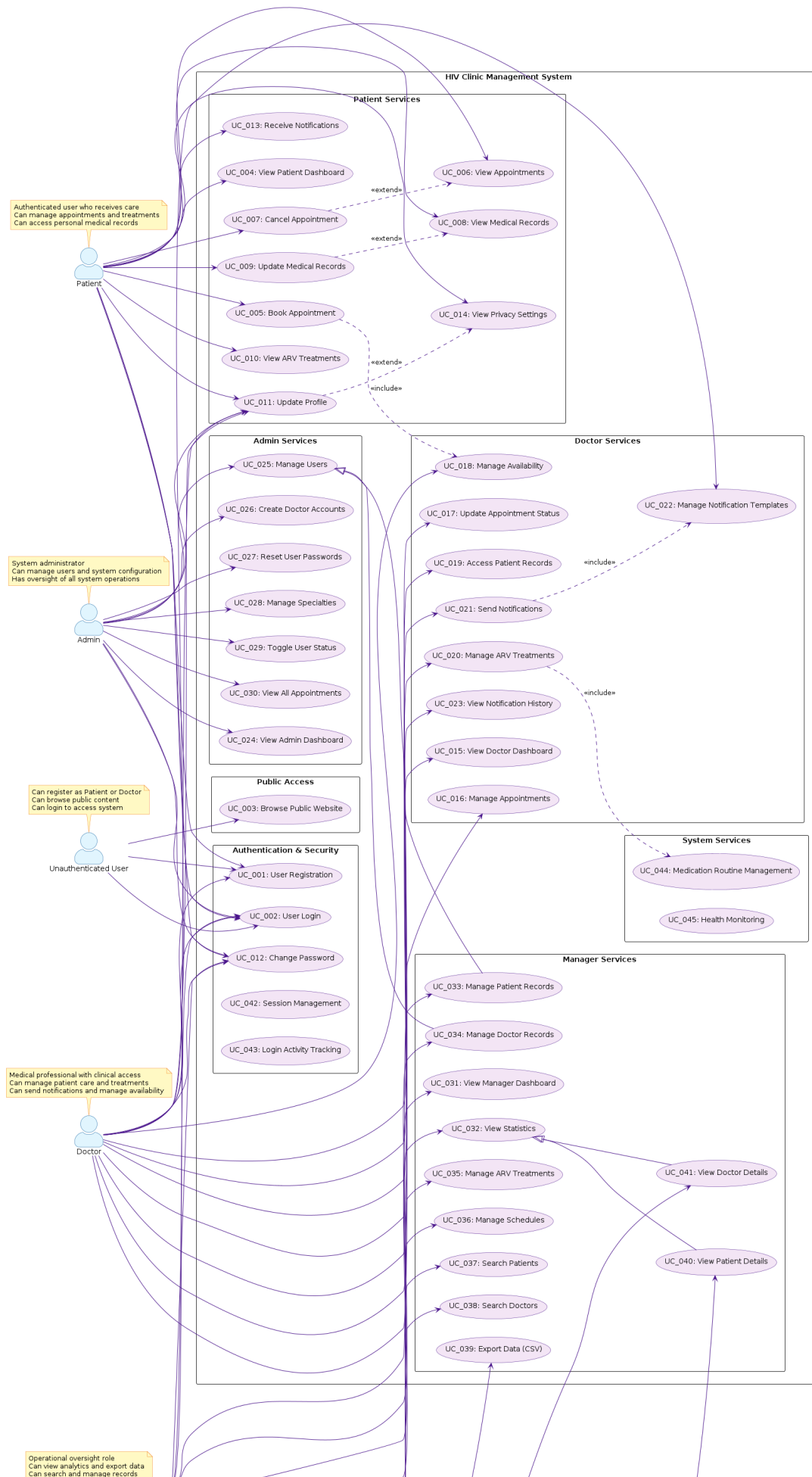
1.1.1 Actors

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 |

1.4 System High Level Design

1.4.1 Database Design

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

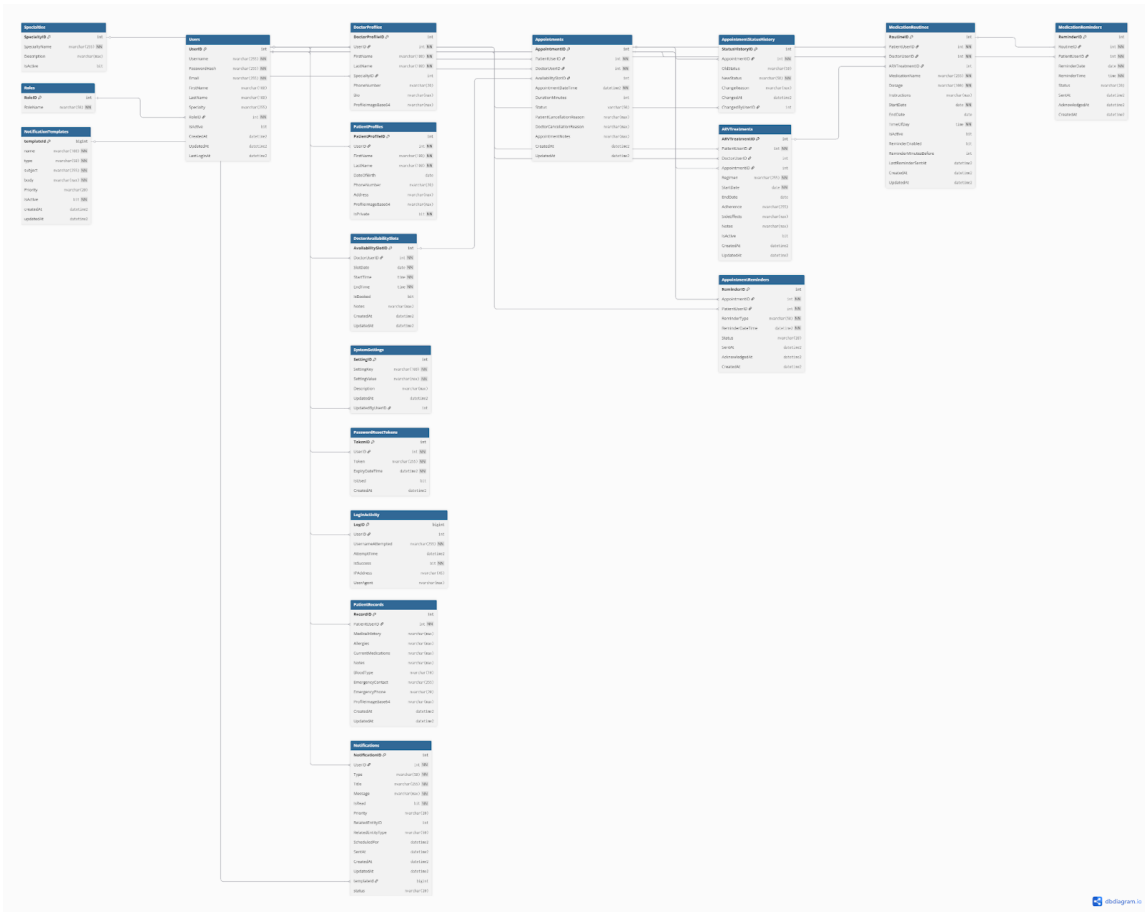


Table 16. Database Design Overview

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. |
| 09 | MedicationRoutines | Daily medication schedules. |
| 10 | Notifications | System notification management. |
| 11 | NotificationTemplates | Reusable notification templates. |

| No | Table | Description |
|---|--------------------------|--|
| 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 | | |

1.4.2 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. |
| 06 | com.hivclinic.config | Configuration classes for security (JWT), database, and application settings. |
| 07 | com.hivclinic.exception | Custom exception handling for application-specific errors. |

| No | Package | Description |
|----|--------------------------|--|
| 08 | com.hivclinic.validation | Input validation and sanitization utilities. |

Table 18. Package Descriptions

1.4.3 Data Flow Architecture

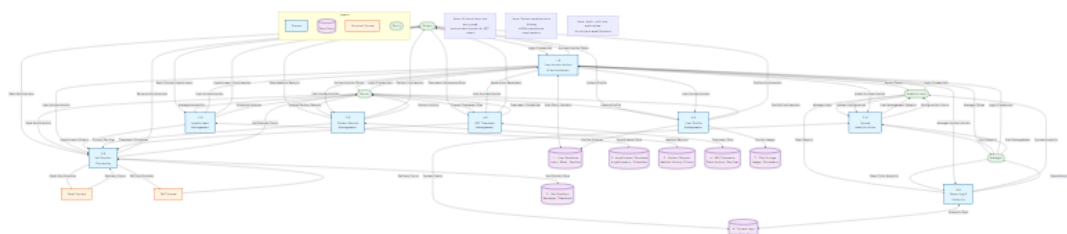


Figure 2: System Data Flow Diagram

2 Requirement Specifications

2.1 Common functions

2.1.1 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 |

2.1.2 UC-02 – Log In

| | |
|---------------------------|---|
| UC ID and Name: | UC-02 – Log In |
| Created By: | KhoaDD |
| Date Created: | 28/6 |
| Primary Actor: | Guest, Donor |
| Secondary Actors: | System |
| Description: | The user logs into the system using their email and password. Upon successful login, the system redirects the user to their personal dashboard. Account access may be limited based on verification level (Level 1 or Level 2). |
| Trigger: | The user submits the login form. |
| Preconditions: | <ul style="list-style-type: none"> • A registered account exists • The user is not currently logged in |
| Postconditions: | <ul style="list-style-type: none"> • The user is authenticated • The user is redirected to their personal dashboard |
| Normal Flow: | <ol style="list-style-type: none"> 1. User opens the login form 2. Enters email and password 3. Submits the form 4. The system validates the credentials and logs the user in 5. User is redirected to the dashboard with access based on account verification level |
| Alternative Flows: | None defined |
| Exceptions: | <ul style="list-style-type: none"> • EX-1: If the email or password is incorrect, show an error message: <i>"Invalid email or password."</i> • EX-2: If the system is unavailable (e.g., server error), show an error message: <i>"Unable to connect. Please try again later."</i> |

| | |
|--------------------------|---|
| Business Rules: | <ul style="list-style-type: none"> • BR-24: Email must follow institutional format (e.g., *@gmail.com) |
| Assumptions: | <ul style="list-style-type: none"> • The account is active and verified • The user provides correct credentials |
| Priority: | High |
| Frequency of Use: | Daily |

2.1.3 UC-03 – View Hospital Information

| | |
|---------------------------|---|
| UC ID and Name: | UC-03 – View Hospital Information |
| Created By: | AnPP |
| Date Created: | 28/6 |
| Primary Actor: | Guest |
| Secondary Actors: | None |
| Description: | The user views information about the hospital, including its services, location, and contact details. |
| Trigger: | The user accesses the homepage or selects the "About Us" section. |
| Preconditions: | <ul style="list-style-type: none"> • The system is online and accessible |
| Postconditions: | <ul style="list-style-type: none"> • Hospital information is displayed to the user |
| Normal Flow: | <ol style="list-style-type: none"> 1. User opens the website 2. Clicks on "About Us" 3. The system displays hospital information |
| Alternative Flows: | None defined |
| Exceptions: | <ul style="list-style-type: none"> • EX-1: If hospital data is unavailable, the system displays a default message or an error: <i>"Hospital information is currently unavailable."</i> |

| | |
|--------------------------|--|
| Business Rules: | <ul style="list-style-type: none"> • BR-19: Only admins can make changes to system about hospital information |
| Assumptions: | <ul style="list-style-type: none"> • Hospital information is properly maintained and updated in the system |
| Priority: | Medium |
| Frequency of Use: | Occasional |

2.1.4 UC-04 – Read Blogs

| | |
|---------------------------|---|
| UC ID and Name: | UC-04 – Read Blogs |
| Created By: | AnPP |
| Date Created: | 29/6 |
| Primary Actor: | Guest |
| Secondary Actors: | System |
| Description: | The user reads educational blog posts related to blood donation. |
| Trigger: | The user clicks on the "Blog" section. |
| Preconditions: | <ul style="list-style-type: none"> • At least one blog post has been published |
| Postconditions: | <ul style="list-style-type: none"> • A list of blogs is displayed • The user can read individual posts |
| Normal Flow: | <ol style="list-style-type: none"> 1. User clicks "Blog" 2. System displays list of available posts 3. User selects and reads a blog |
| Alternative Flows: | AF-1: No blog posts available → Show message: <i>"No blog content is currently available."</i> |
| Exceptions: | <ul style="list-style-type: none"> • EX-1: System error → Show message: <i>"Unable to load blogs. Please try again later."</i> |

| | |
|--------------------------|--|
| Business Rules: | <ul style="list-style-type: none"> • BR-17: Only blog authors or admins can edit blog articles |
| Assumptions: | <ul style="list-style-type: none"> • Blog content is reviewed and approved before being published |
| Priority: | Medium |
| Frequency of Use: | Frequent |

2.1.5 UC-05 – Manage Personal Profile

| | |
|---------------------------|--|
| UC ID and Name: | UC-05 – Manage Personal Profile |
| Created By: | TuanTM |
| Date Created: | 28/6 |
| Primary Actor: | Donor |
| Secondary Actors: | System |
| Description: | The donor views and updates their personal profile, including contact information. |
| Trigger: | The user accesses the "My Profile" section. |
| Preconditions: | <ul style="list-style-type: none"> • The user is logged in to the system |
| Postconditions: | <ul style="list-style-type: none"> • Profile data is updated and saved successfully |
| Normal Flow: | <ol style="list-style-type: none"> 1. User navigates to profile section 2. Edits personal information 3. Clicks "Save" 4. System validates and saves changes |
| Alternative Flows: | AF-1: Invalid input (e.g., phone number format) → Show error: <i>"Please correct the highlighted fields."</i> |
| Exceptions: | <ul style="list-style-type: none"> • EX-1: If update fails due to system error → Show error: <i>"Failed to update profile. Please try again."</i> |

| | |
|--------------------------|---|
| Business Rules: | <ul style="list-style-type: none"> BR-22: User profile information must be accurate and match official documents |
| Assumptions: | <ul style="list-style-type: none"> Profile data is accurate and editable |
| Priority: | High |
| Frequency of Use: | Occasional |

3 Design Specifications

3.1 Authentication System

3.1.1 User Login

This screen allows users to authenticate into the system with role-based access to appropriate functionalities.

Related use cases: UC-002 User Login

UI Design

| Field Name | Field Type | Description |
|------------------|--------------|---|
| Username* | Text Box | User enters registered username or email address for authentication |
| Password* | Password Box | User enters password (masked input for security) |
| Login | Button | Submits authentication request to server |
| Register | Hyperlink | Redirects to user registration page for new users |
| Forgot Password? | Hyperlink | Initiates password reset process |

Database Access

| Table | CRUD | Description |
|---------------|------|--|
| Users | R | Verify username/email and password hash for authentication |
| Roles | R | Retrieve user role information for authorization |
| LoginActivity | C | Log login attempt for security audit |

— 1. *Authenticate user credentials*

```
SELECT u.UserID, u.Username, u.Email, u.IsActive, r.RoleName
FROM Users u
INNER JOIN Roles r ON u.RoleID = r.RoleID
WHERE (u.Username = ? OR u.Email = ?) AND u.IsActive = 1
```

— 2. *Log login activity*

```
INSERT INTO LoginActivity
(UserID, UsernameAttempted, AttemptTime, IsSuccess, IPAddress, UserAgent)
VALUES (?, ?, GETDATE(), ?, ?, ?)
```

3.2 Appointment Management

3.2.1 Appointment Booking

This screen enables patients to book appointments with available doctors by selecting from available time slots.

Related use cases: UC-004 Book Appointment

UI Design

| Field Name | Field Type | Description |
|-----------------------|---------------|---|
| Doctor Selection* | Dropdown | List of available doctors with specialties |
| Appointment Date* | Date Picker | Calendar widget for selecting appointment date |
| Available Time Slots* | Radio Buttons | Dynamic list of available time slots for selected doctor/date |
| Appointment Notes | Text Area | Optional notes about appointment purpose or concerns |
| Book Appointment | Button | Submit appointment booking request |
| Cancel | Button | Return to previous screen without booking |

Database Access

| Table | CRUD | Description |
|-------------------------|------|---|
| Users | R | Retrieve available doctors with their specialties |
| DoctorAvailabilitySlots | RS | Query available slots and mark as booked |

| | | |
|---------------|---|---|
| Appointments | C | Create new appointment record |
| Notifications | C | Schedule appointment reminder notifications |

— 1. *Get available doctors*

```
SELECT u.UserID, u.FirstName, u.LastName, dp.Bio, s.SpecialtyName
FROM Users u
INNER JOIN DoctorProfiles dp ON u.UserID = dp.UserID
LEFT JOIN Specialties s ON dp.SpecialtyID = s.SpecialtyID
WHERE u.RoleID = (SELECT RoleID FROM Roles WHERE RoleName = 'Doctor')
AND u.IsActive = 1
```

— 2. *Get available time slots*

```
SELECT AvailabilitySlotID, SlotDate, StartTime, EndTime
FROM DoctorAvailabilitySlots
WHERE DoctorUserID = ? AND SlotDate = ? AND IsBooked = 0
ORDER BY StartTime
```

— 3. *Create appointment*

```
INSERT INTO Appointments
(PatientUserID, DoctorUserID, AvailabilitySlotID, AppointmentDateTime,
Status, AppointmentNotes, CreatedAt, UpdatedAt)
VALUES (?, ?, ?, ?, 'Scheduled', ?, GETDATE(), GETDATE())
```

— 4. *Update availability slot*

```
UPDATE DoctorAvailabilitySlots
SET IsBooked = 1, UpdatedAt = GETDATE()
WHERE AvailabilitySlotID = ?
```

3.3 Patient Care System

3.3.1 Patient Records Management

This screen provides comprehensive medical record management for HIV patients including treatment history and current medications.

Related use cases: UC-007 Manage Patient Records

UI Design

| Field Name | Field Type | Description |
|-----------------|------------|---|
| Medical History | Text Area | Comprehensive medical history including HIV diagnosis details |

| | | |
|---------------------|-----------|--|
| Current Allergies | Text Area | Known allergies and adverse reactions |
| Current Medications | Text Area | List of current medications including ARV regimens |
| Blood Type | Dropdown | ABO blood type classification |
| Emergency Contact | Text Box | Emergency contact person name |
| Emergency Phone | Text Box | Emergency contact phone number |
| Clinical Notes | Text Area | Doctor's clinical observations and notes |
| Save Record | Button | Save medical record updates |
| View ARV Treatments | Button | Access HIV treatment management screen |

Database Access

| Table | CRUD | Description |
|--------------------|------|---|
| PatientRecords | R,U | Retrieve and update patient medical records |
| ARVTreatments | R | Access HIV treatment history |
| MedicationRoutines | R | View current medication schedules |
| Users | R | Verify doctor access permissions |

— 1. *Retrieve patient record*

```

SELECT RecordID, PatientUserID, MedicalHistory, Allergies,
        CurrentMedications, BloodType, EmergencyContact,
        EmergencyPhone, Notes, UpdatedAt
FROM PatientRecords
WHERE PatientUserID = ?

```

— 2. *Update patient record*

```

UPDATE PatientRecords
SET MedicalHistory = ?, Allergies = ?, CurrentMedications = ?,
    BloodType = ?, EmergencyContact = ?, EmergencyPhone = ?,
    Notes = ?, UpdatedAt = GETDATE()
WHERE PatientUserID = ?

```

— 3. *Get ARV treatment history*

```

SELECT ARVTreatmentID, Regimen, StartDate, EndDate,
        Adherence, SideEffects, IsActive
FROM ARVTreatments

```

```
WHERE PatientUserID = ?  
ORDER BY StartDate DESC
```

4 Appendix

4.1 Assumptions & Dependencies

- **AS-1:** Microsoft SQL Server database is available and properly configured for healthcare data storage
- **AS-2:** SMTP email service is configured for sending appointment and medication reminders
- **AS-3:** System users have basic computer literacy and internet access
- **AS-4:** Clinic staff will receive training on HIV patient management workflows
- **DE-1:** Integration with existing hospital information systems may be required
- **DE-2:** HIPAA compliance requirements must be met for patient data protection
- **DE-3:** System depends on reliable internet connectivity for real-time operations

4.2 Limitations & Exclusions

- System does not include billing or insurance processing capabilities
- Laboratory result integration is not included in current scope
- Telemedicine or video consultation features are excluded
- Mobile application development is not part of initial release
- Integration with pharmacy systems for prescription management is excluded
- Advanced analytics and reporting dashboards are limited in scope

4.3 Business Rules

| ID | Category | Rule Definition |
|--------|--------------------------|---|
| BR-016 | Data Security | All patient data must be encrypted at rest and in transit using AES-256 encryption |
| BR-017 | Access Control | Role-based access ensures patients can only view their own records unless explicitly shared |
| BR-018 | Appointment Scheduling | No overlapping appointments allowed for any doctor or patient |
| BR-019 | Medication Adherence | ARV medication reminders are mandatory for all HIV patients unless opted out |
| BR-020 | Record Retention | Patient medical records must be retained for minimum 7 years per healthcare regulations |
| BR-021 | Emergency Access | Emergency override allows authorized medical staff to access any patient record |
| BR-022 | Notification Preferences | Patients must be able to opt-out of non-critical notifications |
| BR-023 | Data Backup | Daily automated backups of all patient data with 30-day retention |

4.4 Technical Specifications

- **Backend Technology:** Spring Boot 3.x with Java 17
- **Frontend Technology:** React 18 with modern JavaScript (ES6+)
- **Database:** Microsoft SQL Server with T-SQL stored procedures
- **Authentication:** JWT (JSON Web Tokens) with BCrypt password hashing
- **API Architecture:** RESTful APIs with JSON data exchange
- **Security:** HTTPS/TLS encryption, CORS configuration, input validation
- **Deployment:** Containerized deployment ready (Docker compatible)