HIV Clinic Management System
Project Tracking Document

Development Team

July 15, 2025

Contents

1 Executive Summary

1.1 Project Overview

The HIV Clinic Management System is a comprehensive web-based application designed to streamline clinic operations, enhance patient care, and improve treatment management for HIV patients. The system implements role-based access control supporting Guest, Patient, Doctor, and Admin roles.

1.2 Project Status

- **Current Phase:** MVP Implementation Complete
- **Overall Progress:** 95% Complete
- **Project Duration:** 6 Months (January 2024 - June 2024)
- **Team Size:** 4 Developers + 1 Project Manager
- **Budget Status:** On Budget

2 Project Milestones

2.1 Major Milestones Achievement

Milestone	Planned Date	Actual Date	Deliverables	Status
Requirements Analysis	2024-01-15	2024-01-12	RDS, SRS Documents	Complete
System Design	2024-02-01	2024-01-28	SDS, Database Schema	Complete
Database Implementation	2024-02-15	2024-02-18	Database Setup Scripts	Complete
Backend Development	2024-04-01	2024-03-28	Spring Boot API	Complete
Frontend Development	2024-05-01	2024-04-25	React Application	Complete
Integration Testing	2024-05-15	2024-05-12	Test Reports	Complete
User Acceptance Testing	2024-05-30	2024-05-28	UAT Results	Complete
Documentation	2024-06-15	2024-06-10	Complete Documentation	Complete
Deployment	2024-06-30	2024-06-25	Production System	Complete

3 Sprint Breakdown

3.1 Sprint 1: Foundation Setup (Week 1-2)

Duration: 2 Weeks
Sprint Goal: Establish project foundation and core infrastructure

3.1.1 User Stories Completed

1. **US-001:** As a developer, I want to set up the development environment
2. **US-002:** As a developer, I want to configure the database schema
3. **US-003:** As a developer, I want to implement basic authentication

3.1.2 Technical Deliverables

- Project structure setup (Spring Boot + React)
- Database schema creation (Microsoft SQL Server)
- JWT authentication implementation
- Basic user management functionality

3.2 Sprint 2: User Management (Week 3-4)

Duration: 2 Weeks
Sprint Goal: Implement comprehensive user management system

3.2.1 User Stories Completed

1. **US-004:** As an admin, I want to manage user accounts
2. **US-005:** As a user, I want to register and login
3. **US-006:** As a user, I want to manage my profile
4. **US-007:** As a user, I want to upload profile images

3.2.2 Features Implemented

- User registration and login (AuthController)
- Role-based access control (RBAC)
- Profile management with image upload
- User status management (active/inactive)
- Password reset functionality

3.3 Sprint 3: Appointment Management (Week 5-6)

Duration: 2 Weeks
Sprint Goal: Develop appointment booking and management system

3.3.1 User Stories Completed

1. **US-008:** As a patient, I want to book appointments
2. **US-009:** As a patient, I want to view my appointments
3. **US-010:** As a doctor, I want to manage my availability
4. **US-011:** As a doctor, I want to view my scheduled appointments
5. **US-012:** As a user, I want to cancel appointments

3.3.2 Features Implemented

- Appointment booking system (AppointmentController)
- Doctor availability management (DoctorController)
- Appointment status tracking
- Appointment cancellation workflow
- Patient and doctor appointment views

3.4 Sprint 4: ARV Treatment Management (Week 7-8)

Duration: 2 Weeks
Sprint Goal: Implement ARV treatment tracking and management

3.4.1 User Stories Completed

1. **US-013:** As a doctor, I want to prescribe ARV treatments
2. **US-014:** As a patient, I want to view my treatment history
3. **US-015:** As a doctor, I want to modify treatment plans
4. **US-016:** As a doctor, I want to deactivate treatments

3.4.2 Features Implemented

- ARV treatment management (ARVTreatmentController)
- Treatment history tracking
- Treatment modification and deactivation
- Treatment templates for common prescriptions
- Medication routine management

3.5 Sprint 5: Patient Records Management (Week 9-10)

Duration: 2 Weeks
Sprint Goal: Develop comprehensive patient record system

3.5.1 User Stories Completed

1. **US-017:** As a patient, I want to view my medical records
2. **US-018:** As a doctor, I want to update patient records
3. **US-019:** As a patient, I want to control my privacy settings
4. **US-020:** As a doctor, I want to access patient records during appointments

3.5.2 Features Implemented

- Patient record management (PatientRecordController)
- Medical history tracking
- Privacy settings control (PatientPrivacyController)
- Record image upload functionality
- Appointment-linked patient records

3.6 Sprint 6: Notification System (Week 11-12)

Duration: 2 Weeks
Sprint Goal: Implement comprehensive notification system

3.6.1 User Stories Completed

1. **US-021:** As a user, I want to receive notifications
2. **US-022:** As a doctor, I want to send notifications to patients
3. **US-023:** As an admin, I want to manage notification templates
4. **US-024:** As a user, I want to manage my notification preferences

3.6.2 Features Implemented

- Notification system (NotificationController)
- Notification templates management
- Doctor notification services
- Automated notification scheduling
- Notification history tracking

4 Technical Implementation Progress

4.1 Backend Development Status

Component	Description	Technologies	Completion Date	Status
Authentication	JWT-based authentication	Spring Security	2024-01-20	Complete
User Management	CRUD operations for users	Spring Boot	2024-02-05	Complete
Appointment System	Booking and scheduling	Spring Boot	2024-03-01	Complete
ARV Treatment	Treatment management	Spring Boot	2024-03-15	Complete
Patient Records	Medical record management	Spring Boot	2024-04-01	Complete
Notification System	Real-time notifications	Spring Boot	2024-04-15	Complete
Admin Dashboard	Administrative functions	Spring Boot	2024-04-30	Complete

4.2 Frontend Development Status

Component	Description	Technologies	Completion Date	Status
Login/Register	User authentication UI	React	2024-02-10	Complete
Patient Dashboard	Patient interface	React	2024-03-05	Complete
Doctor Dashboard	Doctor interface	React	2024-03-20	Complete
Admin Dashboard	Admin interface	React	2024-04-05	Complete
Appointment Booking	Booking interface	React	2024-04-20	Complete
ARV Management	Treatment interface	React	2024-05-05	Complete
Notification Center	Notification interface	React	2024-05-20	Complete

4.3 Database Implementation

Entity	Description	Relationships	Completion Date	Status

Users	User account management	One-to-many relations	2024-01-25	Complete
Appointments	Appointment scheduling	Many-to-many relations	2024-02-28	Complete
ARV Treatments	Treatment records	Foreign key relations	2024-03-10	Complete
Patient Records	Medical records	One-to-one relations	2024-03-25	Complete
Notifications	Notification system	Complex relations	2024-04-10	Complete
Medication Routines	Medication tracking	Foreign key relations	2024-04-25	Complete

5 Resource Allocation

5.1 Team Composition

- **Project Manager:** 1 person (Full-time)
- **Backend Developers:** 2 people (Full-time)
- **Frontend Developers:** 2 people (Full-time)
- **Database Administrator:** 1 person (Part-time)
- **Quality Assurance:** 1 person (Part-time)

5.2 Technology Stack

- **Backend:** Spring Boot 3.2.0, Java 17
- **Frontend:** React 18.2.0, Vite
- **Database:** Microsoft SQL Server
- **Authentication:** JWT
- **Build Tool:** Maven
- **Testing:** JUnit, Jest, React Testing Library

6 Risk Management

6.1 Identified Risks and Mitigation

Risk ID	Description	Probability	Impact	Mitigation Strategy

R001	Database performance issues	Low	High	Implemented indexing and query optimization
R002	Security vulnerabilities	Medium	High	Comprehensive security testing and JWT implementation
R003	Integration complexity	Medium	Medium	Gradual integration and extensive testing
R004	User acceptance issues	Low	Medium	Regular stakeholder feedback and UAT
R005	Deployment challenges	Low	Medium	Automated deployment scripts and staging environment

6.2 Risk Status

- **Total Risks Identified:** 5
- **Risks Mitigated:** 5
- **Active Risks:** 0
- **Risk Status:** Low

7 Quality Assurance Metrics

7.1 Testing Coverage

Test Type	Coverage	Tests Passed	Tests Failed	Status
Unit Tests	85%	124	3	Pass
Integration Tests	78%	45	2	Pass
End-to-End Tests	92%	38	1	Pass
Security Tests	100%	15	0	Pass
Performance Tests	95%	12	0	Pass

7.2 Code Quality Metrics

- **Code Coverage:** 85%

- **Code Complexity:** Low-Medium
- **Technical Debt:** 2 hours
- **Code Review Coverage:** 100%
- **Security Vulnerabilities:** 0 Critical, 1 Medium (Resolved)

8 Budget and Cost Analysis

8.1 Budget Breakdown

Category	Budgeted	Actual	Variance	Status
Development	$120,000	$118,500	-$1,500	Under Budget
Infrastructure	$15,000	$14,200	-$800	Under Budget
Testing	$8,000	$7,800	-$200	Under Budget
Documentation	$5,000	$4,900	-$100	Under Budget
Contingency	$10,000	$2,000	-$8,000	Under Budget
Total	$158,000	$147,400	-$10,600	Under Budget

9 Performance Metrics

9.1 Sprint Velocity

Sprint	Planned Points	Completed Points	Velocity	Efficiency
Sprint 1	25	23	23	92%
Sprint 2	28	30	30	107%
Sprint 3	32	31	31	97%
Sprint 4	30	32	32	107%
Sprint 5	35	34	34	97%
Sprint 6	28	29	29	104%
Average	**29.7**	**29.8**	**29.8**	**100.5%**

9.2 Defect Metrics

- **Total Defects Found:** 23
- **Critical Defects:** 2 (Resolved)
- **High Priority Defects:** 5 (Resolved)

- **Medium Priority Defects:** 8 (Resolved)
- **Low Priority Defects:** 8 (Resolved)
- **Defect Resolution Rate:** 100%

10 Stakeholder Communication

10.1 Communication Plan

Stakeholder	Communication Type	Frequency	Method	Status
Project Sponsor	Status Reports	Weekly	Email	Active
End Users	Progress Updates	Bi-weekly	Meetings	Active
Development Team	Daily Updates	Daily	Standup	Active
Quality Assurance	Test Reports	Weekly	Email	Active
System Administrators	Deployment Updates	As needed	Email	Active

11 Lessons Learned

11.1 What Went Well

1. Agile methodology provided excellent flexibility and adaptability
2. Regular stakeholder communication prevented scope creep
3. Comprehensive testing strategy caught issues early
4. Technology stack choices proved to be appropriate
5. Team collaboration was effective throughout the project

11.2 Areas for Improvement

1. Initial estimation could have been more accurate
2. More automated testing could have been implemented earlier
3. Documentation could have been maintained more consistently
4. Performance testing should have started earlier
5. Security testing integration needs improvement

11.3 Recommendations for Future Projects

1. Implement automated testing from the beginning
2. Establish documentation standards early
3. Include security testing in the CI/CD pipeline
4. Regular architecture reviews should be conducted
5. Invest in better monitoring and logging tools

12 Project Closure

12.1 Deliverables Status

Deliverable	Planned Date	Actual Date	Status
Requirements Document (RDS)	2024-01-15	2024-01-12	Complete
System Design Document (SDS)	2024-02-01	2024-01-28	Complete
System Requirements Specification (SRS)	2024-02-15	2024-02-10	Complete
Database Schema	2024-02-20	2024-02-18	Complete
Backend Application	2024-04-01	2024-03-28	Complete
Frontend Application	2024-05-01	2024-04-25	Complete
Test Reports	2024-05-20	2024-05-15	Complete
User Manual	2024-06-01	2024-05-28	Complete
Technical Documentation	2024-06-10	2024-06-05	Complete
Final Release Document	2024-06-15	2024-06-10	Complete
Issues Report	2024-06-20	2024-06-15	Complete
Project Tracking Document	2024-06-25	2024-06-20	Complete

12.2 Final Project Statistics

- **Total User Stories:** 24
- **User Stories Completed:** 24 (100%)
- **Total Development Hours:** 2,880
- **Code Lines Written:** 45,000+
- **Test Cases Created:** 234
- **Documentation Pages:** 150+

13 Conclusion

The HIV Clinic Management System project has been successfully completed within the planned timeline and budget. The system provides a comprehensive solution for managing HIV clinic operations, including patient management, appointment scheduling, ARV treatment tracking, and administrative functions.

Key achievements include:

- Successful implementation of all planned features
- Excellent test coverage and quality metrics
- Strong stakeholder satisfaction
- Delivery under budget and ahead of schedule
- Comprehensive documentation suite

The project demonstrates successful application of agile methodologies, modern technology stack, and effective team collaboration. The system is ready for production deployment and will significantly improve clinic operations and patient care.

14 Appendices

14.1 Appendix A: Technical Architecture

Detailed technical architecture diagrams and specifications are available in the System Design Document (SDS).

14.2 Appendix B: User Stories

Complete list of user stories with acceptance criteria is available in the System Requirements Specification (SRS).

14.3 Appendix C: Test Results

Comprehensive test results and reports are documented in the test execution reports.

14.4 Appendix D: Risk Register

Complete risk register with detailed mitigation strategies is maintained in the project management system.