CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342- GO2
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: DigiHealth: Digital Clinic & Appointment Booking System
Prepared By: [Your Name / System Name/ Role]

Date of Submission: [MM/DD/YYYY]

Version: 1.0

Table of Contents

1. Introduction

This Functional Requirements Specification (FRS) outlines the Minimum Viable Product (MVP) features for the DigiHealth: Digital Clinic & Appointment Booking System. Given the revised timeline with final presentation in the 1st week of December 2025, this document prioritizes core functionalities that can be realistically implemented, tested, and demonstrated within 7-8 weeks (October 14 - December 6, 2025).

Purpose: This document serves as the foundation for MVP development focusing on essential features that demonstrate the system's core value proposition for digital clinic management.

Scope Adjustment: The MVP approach prioritizes appointment booking, basic patient records, and authentication while deferring complex features like QR check-in, advanced analytics, and comprehensive inventory management to future phases.

Intended Audience:

- Development team for immediate implementation guidance
- Presentation evaluators for understanding deliverable scope
- Future development teams for post-presentation enhancements

2. Project Overview

Background

Campus and small community health clinics rely on manual processes that create inefficiencies. DigiHealth MVP will demonstrate core digital solutions within the constrained academic timeline.

Goals (MVP Focus)

- Primary Goal: Develop a functional demo showcasing digital appointment booking and basic patient record management
- Secondary Goal: Implement secure authentication and role-based access
- Presentation Goal: Demonstrate working mobile app and web dashboard by December 1st week

MVP Scope (What We WILL Build)

- Mobile Application (Android): Patient registration and appointment booking

- Web Dashboard: Staff access to patient records and appointments
- Backend APIs: Core data management and authentication
- Database: Patient, appointment, and basic user data storage
- Authentication: Google OAuth 2.0 integration

3. Team

Name	Role
William Bustamante	Lead Developer / Project Manager
Jessie Noel Lapure	Frontend/Backend Developer
Joel Verano	Frontend/Backend Developer
Matthew Rimar Martus	Mobile App Developer

4. Functional Requirements

FR-1: Patient Registration (Mobile App)

- Description: Basic patient account creation with essential information
- Priority: CRITICAL
- Implementation: Simple form with name, email, phone, basic medical info
- Timeline: 3-4 days development

FR-2: Appointment Booking (Mobile App)

- Description: Simple appointment scheduling with available time slots
- Priority: CRITICAL
- Implementation: Calendar view with preset time slots, basic appointment types
- Timeline: 4-5 days development

FR-3: Patient Record Management (Web Dashboard)

- Description: Staff access to view and edit patient information
- Priority: CRITICAL
- Implementation: Basic CRUD operations for patient data
- Timeline: 5-6 days development

FR-4: Authentication System

- Description: Google OAuth 2.0 for both mobile and web platforms

- Priority: CRITICAL
- Implementation: Secure login for patients and staff
- Timeline: 2-3 days development

HIGH PRIORITY FEATURES

FR-5: Basic Role-Based Access

- Description: Three user types - Patient, Staff, Admin
- Priority: HIGH
- Implementation: Simple permission levels for different interfaces
- Timeline: 3-4 days development

FR-6: Patient Search Function

- Description: Search patients by name or ID in web dashboard
- Priority: HIGH
- Implementation: Basic search with filter capabilities
- Timeline: 2-3 days development

OPTIONAL FEATURES (Nice to Have for Presentation)

FR-7: Basic Inventory Tracking

- Description: Simple manual entry of medicine inventory
- Priority: MEDIUM (Optional)
- Implementation: Basic add/edit medicine list (if time permits)
- Timeline: 3-4 days development

FR-8: Simple Dashboard Reports

- Description: Basic charts showing appointments and patient counts
- Priority: LOW (Demo Enhancement)
- Implementation: Simple Chart.js visualizations (if time permits)
- Timeline: 2-3 days development

5. Non-Functional Requirements

NFR-1: Platform Requirements

- Mobile: Android app (minimum viable functionality)
- Web: Modern browser compatibility (Chrome, Firefox primarily)

NFR-2: Technology Stack

- Backend: Spring Boot (REST APIs)
- Frontend: React (simplified dashboard)
- Mobile: Kotlin for Android
- Database: MySQL (single database, simplified schema)
- Authentication: Google OAuth 2.0

NFR-3: Performance Requirements

- Response Time: Under 5 seconds for basic operations
- Concurrent Users: Support for demo purposes (5-10 concurrent users)
- Uptime: Stable for presentation demonstration

NFR-4: Security Requirements

- Authentication: Google OAuth 2.0
- Data Protection: Basic encryption for sensitive data
- Access Control: Simple role-based permissions

NFR-5: Hosting

- Platform: Free tier hosting (Railway/Heroku/AWS Free Tier)
- Requirements: Stable during presentation period
- Backup: Basic data backup for demo protection

6. Use Cases

Use Case 1: Patient Registration and Login (CRITICAL)

Actor: Patient
Precondition: Patient has Android smartphone
Main Flow:

1. Patient downloads DigiHealth app
2. Patient taps "Register" and signs in with Google account
3. Patient completes basic profile (name, phone, emergency contact)
4. System creates patient account and profile
5. Patient can now access appointment booking

Postcondition: Patient account created and ready for appointment booking

Use Case 2: Book Appointment (CRITICAL)

Actor: Patient
Precondition: Patient is registered and logged in
Main Flow:

1. Patient opens appointment booking section
2. Patient selects appointment type (General Consultation, Check-up)
3. Patient chooses available date from calendar
4. Patient selects available time slot
5. Patient confirms appointment details
6. System saves appointment and sends confirmation

Postcondition: Appointment is booked and visible in both mobile app and web dashboard

Use Case 3: View Patient Records (CRITICAL)

Actor: Clinic Staff
Precondition: Staff member logged into web dashboard
Main Flow:

1. Staff member accesses web dashboard
2. Staff searches for patient by name or views appointment list
3. Staff selects patient to view profile and medical history
4. Staff can view appointment details and basic medical information
5. Staff can add simple notes or update contact information

Postcondition: Patient information is viewed and optionally updated

Use Case 4: Manage Daily Appointments (HIGH PRIORITY)

Actor: Clinic Staff
Precondition: Staff member has admin access
Main Flow:

1. Staff accesses appointment management section
2. Staff views daily appointment schedule
3. Staff can see patient details for each appointment
4. Staff can mark appointments as completed or cancelled
5. Staff can view patient count and basic statistics

Postcondition: Daily appointments are managed and tracked

7. System Interfaces

External Integrations (Simplified)

- Google OAuth 2.0 API: For authentication (both mobile and web)
- Cloud Database: MySQL hosted on cloud platform
- Hosting Platform: Railway/Heroku/AWS Free Tier APIs

Internal Interfaces

- Mobile App ↔ Backend: REST API calls for appointments and patient data
- Web Dashboard ↔ Backend: REST API calls for staff operations
- Backend ↔ Database: JPA/Hibernate for data persistence
-

8. Assumptions and Constraints

Revised Timeline Constraints

- Total Timeline: 7-8 weeks (October 14 - December 6, 2025)
- Development Phase: 4 weeks maximum
- Testing Phase: 1 week
- Presentation Prep: 1 week
- Final Presentation: December 1st week, 2025

Resource Constraints

- Team Size: 4 student developers with academic commitments
- Budget: Limited to free tier services and minimal costs
- Infrastructure: Free hosting platforms only

Feature Constraints (MVP Approach)

- No QR Code functionality in initial version
- No advanced analytics - basic charts only if time permits
- No complex inventory management - manual entry only
- No real-time notifications - basic email notifications at most
- No mobile iOS version - Android only

Assumptions

- Development Environment: Team has access to required development tools
- Presentation Setup: Demo will be conducted in controlled environment
- User Testing: Limited to team testing and basic functionality verification
- Data: Sample data will be used for demonstration purpose

9. Acceptance Criteria

MVP Acceptance Criteria

Core Functionality Demonstration:

- ✅ Android app successfully demonstrates patient registration
- ✅ Android app shows appointment booking with time slot selection
- ✅ Web dashboard displays patient list and appointment management
- ✅ Google OAuth authentication works for both platforms
- ✅ Basic patient record creation and editing functions

Technical Demonstration:

- ✅ Mobile app connects to backend APIs successfully
- ✅ Web dashboard retrieves and displays data from database
- ✅ System handles basic error scenarios gracefully
- ✅ Authentication flow works without major issues
- ✅ Data persistence demonstrated across sessions

Presentation Criteria:

- ✅ 10-15 minute demo showing end-to-end user journey
- ✅ Patient registration → appointment booking → staff viewing appointment
- ✅ System runs stably during presentation
- ✅ Key features explained clearly with live demonstration
- ✅ Future roadmap presented for deferred features

Documentation Criteria:

- ✅ Updated FRS reflecting MVP approach
- ✅ Basic user documentation for demo
- ✅ Technical architecture overview
- ✅ Lessons learned and next steps presentation

Post-Presentation Success Metrics

- Immediate: Successful demo presentation with working core features
- Academic: Meet course requirements and presentation criteria
- Future: Foundation established for continued development in subsequent phases

10. Approval Sign-off

	Full Name	Signature	Date
Prepared By:			
Developer			
Developer			
Lead Developer			
Reviewed By:			
Reviewed By:			

Approved By:	Mr. Frederick Revilleza		

10. Appendices

Appendix A: Revised Project Timeline (Realistic)

Phase	Duration	Start Date	End Date	Key Deliverables
Requirements & Design	2 weeks	Oct 14, 2025	Oct 27, 2025	FRS Document, UI Mockups, Database Schema

Core Development (MVP)	4 weeks	Oct 28, 2025	Nov 24, 2025	Android App, Web Dashboard, Backend APIs
Testing & Integration	1 week	Nov 25, 2025	Dec 1, 2025	Bug fixes, Integration testing
Presentation Preparation	1 week	Dec 2, 2025	Dec 6, 2025	Demo setup, Presentation materials

Total Timeline: 8 weeks
Presentation Date: December 6, 2025 (estimated)

Appendix B: MVP Development Priorities

Priority Level	**Features**	**Development Time**	**Status**
MUST HAVE	Patient registration, App Appointment booking, Staff dashboard, Authentication	14-18 days	Required for demo

SHOULD HAVE	Patient search, Role-based access	5-7 days	Important forcompleteness
COULD HAVE	Basic inventory, Simple reports	5-7 days	Time permitting
DEFERRED	QR check-in, Advanced analytics, Queue management	TBD	Post-presentation

Appendix C: Risk Mitigation (MVP Focus)

Risk	**Likelihood**	**Impact**	**Mitigation Strategy**
Timeline overrun	High	High	Focus on MVP features only, defer complex functionality

Technical integration issues	Medium	High	Start with simple authentication, avoid complex integrations
Team coordination challenges	Medium	Medium	Weekly team meetings, clear role assignments
Presentation demo failures	Medium	High	Prepare backup demo videos, test thoroughly before presentation

Appendix D: Technology Stack (Confirmed for MVP)

Backend:

- Spring Boot 2.7+ (REST APIs)
- MySQL 8.0+ (Database)
- Google OAuth 2.0 (Authentication)

Frontend:

- React 18+ (Web Dashboard)

- Tailwind CSS (Styling)
- Chart.js (Basic charts if implemented)

Mobile:

- Kotlin (Android Development)
- Android Studio (Development Environment)
- Retrofit (HTTP Client)

Hosting:

- Railway/Heroku (Backend)
- Cloud database service (MySQL)

Document Control:

- Version: 1.0 (MVP - Realistic Timeline)
- Last Updated: October 14, 2025
- Next Review: , 2025 (Start of development phase)
- Document Owner: William Bustamante
- Approval Authority: Mr. Frederick Revilleza
- Final Presentation: December 1st week, 2025