MedRhythms Mobile App Testing Report

Version 1.0

Prepared by

Name	Date Signed
Ifeanyi Ineh	04/04/2025
Yiran Zhao	04/04/2025
Yoga Srinivas Kasireddy	04/04/2025
Chaoyi Jiang	04/04/2025

Group Name: Fantastic Four

Instructor: Dr. Gary Cantrell

Course: Foundations of Software Engineering

Teaching Assistant: Sam Morris

April 4, 2025

Contents

1 Revision History

Version	1.0
Date	April 4, 2025
Document Status	Draft
Document Owner	Testing Team
Primary Author	Yiran Zhao
Co-Author	Yoga Srinivas Kasireddy

2 Project Description

This project is developed for MedRhythms, a company specializing in therapeutic solutions for individuals with neurological conditions. The Android application helps patients with neurological diseases, Parkinson's disease, and stroke survivors monitor their walking performance. The app synchronizes music tempo with the user's walking rhythm, leveraging both technology and music for therapeutic benefits. Built using Flutter, the application features IMEI-based authentication, walking session tracking, record displays, and Spotify integration to create a personalized therapeutic experience.

3 Overall Testing Plan

This testing plan outlines a comprehensive strategy to ensure the MedRhythms application meets all functional and non-functional requirements while maintaining high quality standards. The testing approach includes:

- **Static Testing:** Code review to evaluate code quality, structure, and adherence to best practices before execution.
- **Unit Testing:** Testing individual components and functions in isolation.
- **Integration Testing:** Testing interactions between components.
- **Validation Testing:** Testing user workflows to validate requirements.
- **System Testing:** Testing the application in real-world scenarios.

The testing process will follow these stages:

1. Planning and test case preparation
2. Test execution
3. Defect reporting and tracking
4. Retesting and regression testing
5. Final review

Testing will be performed exclusively on Android devices. Test results are documented in the Excel sheet.

4 Testing Sections

Due to the extensive number of test cases, each testing section is provided in detail in the attached Excel document. Below is an outline of the testing IDs under each testing type:

4.1 Static Testing

- ST0001 – Check naming consistency in route files
- ST0002 – Check UI code organization
- ST0003 – Review health module code quality
- ST0004 – Check security in session handling
- ST0005 – Evaluate record service implementation
- ST0006 – Check code documentation quality
- ST0007 – Check error handling approach

Test ID	Requirement ID	Description and Procedure
ST0001	RQ-4.1 - Route Management	Tester: Yoga Environment: Flutter project codebase Goal: Check naming consistency in route files **Procedure:** 1. Review route files (createroutes.dart, deleteroutes.dart, modifyroutes.dart, readroutes.dart) 2. Check if route functions follow similar naming patterns 3. Make sure route parameters use consistent names 4. Check if route paths are structured consistently 5. Look for adequate documentation on route functions Expected: All route files should use consistent naming for functions, parameters, and have proper documentation. Comments: MedRhythms Testing Status

ST0002	RQ-4.1 - Code Structure	Tester: Yoga Environment: Flutter project codebase Goal: Check if UI code is properly organized **Procedure:** 1. Review UI pages (bottombar.dart, home_page.dart, loginpage.dart, medrhythmslogo.dart, records_page.dart, sessions_page.dart) 2. Check if data logic is separate from display code 3. Make sure state management works correctly 4. Verify UI components can be reused 5. Check for consistent styling across pages 6. Look at widget organization Expected: UI pages should separate display from logic, use consistent styling, and organize widgets properly. Comments: MedRhythms Testing Status
ST0003	RQ-4.2 - Health Data Management	Tester: Yoga Environment: Flutter project codebase Goal: Review health module code quality **Procedure:** 1. Check workout.dart in the health directory 2. Look for good documentation in health functions 3. Make sure health data models are clearly defined 4. Check for input validation 5. Verify health data privacy measures 6. Check for proper unit conversions Expected: Health code should have good documentation, clear data models, validate inputs, and follow privacy standards. Comments: MedRhythms Testing Status

ST0004	RQ-4.2 - User Session Management	Tester: Yoga Environment: Flutter project codebase Goal: Check security in session handling **Procedure:** 1. Review sessions.dart in the userappactions directory 2. Check how user credentials are stored 3. Make sure session timeouts are handled 4. Check logout and session clearing functions Expected: Session management should store data securely, handle timeouts properly, and clear sessions on logout. Comments: MedRhythms Testing Status
ST0005	RQ-4.3 - Service Layer Design	Tester: Yoga Environment: Flutter project codebase Goal: Evaluate record service implementation **Procedure:** 1. Check record_service.dart in the services directory 2. Make sure it's separate from UI code 3. Check for error handling 4. Verify resource management is efficient Expected: Service layer should be independent from UI, handle errors properly, and manage resources efficiently. Comments: MedRhythms Testing Status

ST0006	RQ-4.1 - Code Documentation	Tester: Yoga Environment: Flutter project codebase Goal: Check code documentation quality **Procedure:** 1. Look at documentation in all Dart files 2. Check if classes and methods have comments 3. Make sure complex code is explained 4. Check for consistent documentation style 5. Verify documentation matches current code Expected: Code should have clear comments for classes and methods with consistent style. Comments: MedRhythms Testing Status
ST0007	RQ-4.1 - Error Handling	Tester: Yoga Environment: Flutter project codebase Goal: Check error handling approach **Procedure:** 1. Review error handling throughout the code 2. Check if exceptions are caught properly 3. Make sure error messages make sense to users 4. Check for recovery mechanisms 5. Verify null safety implementation Expected: Code should handle errors consistently with clear messages and proper exception handling. Comments: MedRhythms Testing Status

4.2 Unit Testing

- UT0001 – Test route functionality
- UT0002 – Test user login functionality
- UT0003 – Test workout data calculations
- UT0004 – Test user session handling
- UT0005 – Test record retrieval operations

- UT0006 – Test bottom navigation bar functionality
- UT0007 – Test records page functionality
- UT0008 - Test health data synchronization

Test ID	**Requirement ID**	**Description and Procedure**
UT0001	RQ-4.1 - Route Creation	Tester: Yoga Environment: Flutter project test environment Goal: Test route functionality **Procedure:** 1. Create test for createroutes.dart, deleteroutes.dart, modifyroutes.dart, and readroutes.dart 2. Test route creation, modification, read, delete with valid parameters 3. Test route creation, modification, read, delete with invalid parameters 4. Verify error handling for edge cases Expected: Route functions should successfully create, modify, read, and delete routes with valid parameters and handle errors properly for invalid inputs. Comments: MedRhythms Testing Status
UT0002	RQ-4.2 - User Authentication	Tester: Ineh Environment: Flutter project test environment Goal: Test user login functionality **Procedure:** 1. Create test for login functions in loginpage.dart 2. Mock authentication service 3. Test login with valid credentials 4. Test login with invalid credentials 5. Test login with empty fields 6. Verify error messages for failed logins Expected: Login function should authenticate users with valid credentials and show appropriate error messages for invalid inputs. Comments: MedRhythms Testing Status

UT0003	RQ-4.2 - Workout Data Processing	Tester: Yoga Environment: Flutter project test environment Goal: Test workout data calculations **Procedure:** 1. Create test for data processing functions in workout.dart 2. Prepare sample workout data 3. Test session data created 4. Verify calculations match expected values Expected: Workout functions should calculate accurate results and store them in the database. Comments: MedRhythms Testing Status
UT0004	RQ-4.2 - Session Management	Tester: Yoga Environment: Flutter project test environment Goal: Test user session handling **Procedure:** 1. Create test for sessions.dart functions 2. Test session creation 3. Test session storage Expected: Session management should correctly create and save user sessions. Comments: MedRhythms Testing Status
UT0005	RQ-4.3 - Data retrieval	Tester: Yiran Environment: Flutter project test environment Goal: Test record retrieval operations **Procedure:** 1. Create test for readroutes.dart 2. Mock data storage dependencies 3. Test record retrieval function Expected: Record service should correctly retrieve records with proper error handling. Comments: MedRhythms Testing Status

UT0006	RQ-4.1 - Navigation Bar	Tester: Ineh Environment: Flutter project test environment Goal: Test bottom navigation bar functionality **Procedure:** 1. Create test for bottombar.dart 2. Test tab selection mechanism 3. Test navigation between different screens 4. Test active tab indication 5. Verify correct screen is displayed for each tab Expected: Bottom navigation bar should correctly handle tab selection and navigate to the appropriate screens. Comments: MedRhythms Testing Status
UT0007	RQ-4.3 - Records Page	Tester: Chaoyi Environment: Flutter project test environment Goal: Test records page functionality **Procedure:** 1. Create test for records_page.dart 2. Test records loading mechanism 3. Test records display formatting 4. Test record filtering functionality 5. Test record sorting options 6. Verify empty state handling when no records exist Expected: Records page should correctly load, display, filter, and sort records, and properly handle navigation to record details. Comments: MedRhythms Testing Status

UT0008	RQ-4.2 - Health Data Sync	Tester: Ineh Environment: Flutter project test environment Goal: Test health data synchronization **Procedure:** 1. Create test for health data sync functions 2. Mock device health data sources 3. Test data extraction from source 4. Test data transformation 5. Test data storage after sync 6. Test handling of duplicate data 7. Verify error handling for sync failures Expected: Health data sync should correctly extract, transform, and store health data with proper error handling. Comments: MedRhythms Testing Status

4.3 Integration Testing

- IT0001 – IMEI Authentication Flow
- IT0002 – Workout Cycle
- IT0003 – Spotify Integration
- IT0004 – Records Management
- IT0005 – Navigation and UI Flow
- IT0006 – Device Health Integration

Test ID	**Requirement ID**	**Description and Procedure**

IT0001	RQ-4.1 - IMEI Authentication Flow	Tester: Yiran Environment: Flutter integration test environment Goal: Test IMEI login process **Procedure:** 1. Launch the app in test mode 2. Enter valid 15-digit IMEI number 3. Verify successful authentication 4. Test with invalid IMEI number 5. Verify appropriate error handling Expected: Users should be able to login with valid IMEI and receive appropriate errors for invalid IMEI. Comments: MedRhythms Testing Status
IT0002	RQ-4.2 - Workout Cycle	Tester: Yiran Environment: Flutter integration test environment Goal: Test complete workout flow from start to finish **Procedure:** 1. Login with test IMEI 2. Navigate to workout screen 3. Start a new workout 4. Complete workout 5. Sync workout results 6. Verify workout data appears in records 7. Check workout records Expected: Users should be able to complete the entire workout cycle with data properly recorded. Comments: MedRhythms Testing Status

IT0003	RQ-4.3 - Spotify Integration	Tester: Yiran Environment: Flutter integration test environment with Spotify account Goal: Test Spotify music integration with workout **Procedure:** 1. Login with test IMEI 2. Start a new workout 3. Verify Spotify automatically plays music 4. Check if music tempo matches workout pace 5. Change workout pace and verify music adapts 6. Complete workout and stop 7. Verify music record is saved with workout data 8. Check record shows correct music information Expected: Spotify should play music that matches workout pace and music details should be recorded with workout data. Comments: MedRhythms Testing Status
IT0004	RQ-4.3 - Records Management	Tester: Yiran Environment: Flutter integration test environment Goal: Test records viewing and manipulation **Procedure:** 1. Login with IMEI containing pre-populated records 2. Navigate to records page 3. Test filtering and sorting records 4. View detailed record information Expected: Users should be able to view and filter workout records. Comments: MedRhythms Testing Status

IT0005	RQ-4.1 - Navigation and UI Flow	Tester: Yiran Environment: Flutter integration test environment Goal: Test overall app navigation and UI interactions **Procedure:** 1. Launch app and login with IMEI 2. Test navigation between all main screens 3. Verify bottom navigation bar functionality 4. Verify UI element interactions and feedback Expected: App navigation should be intuitive and user-friendly. Comments: MedRhythms Testing Status
IT0006	RQ-4.2 - Device Health Integration	Tester: Yiran Environment: Flutter integration test environment Goal: Test integration with device health sensors **Procedure:** 1. Login with test IMEI 2. Start a workout session 3. Verify app correctly reads step count from device 4. Test GPS tracking during workout 5. Compare app measurements with device health app data 6. Verify data accuracy and consistency Expected: App should correctly integrate with device health kit. Comments: MedRhythms Testing Status

4.4 Validation Testing

- VT0001 – User Authentication Success
- VT0002 – User Authentication Failure
- VT0003 – Start a User Workout Session
- VT0004 – Pause a User Workout Session
- VT0005 – Cancel a User Workout Session

- VT0006 – User Workout Session Progress
- VT0007 – User Workout Session Completion
- VT0008 – User Workout Session Music
- VT0009 – User Workout Session Music Pausing
- VT0010 – User Workout Session Music Cancelling

Test ID	**Requirement ID**	**Description and Procedure**
VT0001	RQ-4.1 - User Authentication Success	Tester: Chaoyi Environment: Android Goal: User enters a valid IMEI and logs in with valid credentials **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter a 15-digit IMEI 4. Click the Login button Expected: Redirected to Home Screen. Comments: MedRhythms Testing Status
VT0002	RQ-4.1 - User Authentication Failure	Tester: Chaoyi Environment: Android Goal: User enters an invalid IMEI and credentials **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter a 14-digit IMEI 4. Click the Login button Expected: Warning shown; no redirection to Home Screen. Comments: MedRhythms Testing Status

VT0003	RQ-4.2 - Start a User Workout Session	Tester: Chaoyi Environment: Android Goal: User starts a successful workout session **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Navigate to Home screen 5. Press Yes to start workout Expected: Redirected to Workout Screen. Comments: MedRhythms Testing Status
VT0004	RQ-4.2 - Pause a User Workout Session	Tester: Chaoyi Environment: Android Goal: User pauses a workout session **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Start a workout session 5. Press Pause Expected: Workout pauses successfully. Comments: MedRhythms Testing Status
VT0005	RQ-4.2 - Cancel a User Workout Session	Tester: Chaoyi Environment: Android Goal: User cancels a workout session **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Start a workout session 5. Press Cancel Expected: Workout canceled successfully. Comments: MedRhythms Testing Status

VT0006	RQ-4.2 - User Workout Session Progress	Tester: Chaoyi Environment: Android Goal: Track time and progress during workout **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Start a workout session 5. Observe time and progress Expected: User sees workout status and remaining time. Comments: MedRhythms Testing Status
VT0007	RQ-4.2 - User Workout Session Completion	Tester: Chaoyi Environment: Android Goal: Complete workout and return to workout page **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Complete the workout session Expected: Return to workout page after completion. Comments: MedRhythms Testing Status
VT0008	RQ-4.3 - User Workout Session Music	Tester: Chaoyi Environment: Android Goal: Start music during workout session **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Start a workout session Expected: Music plays during workout. Comments: MedRhythms Testing Status

VT0009	RQ-4.3 - Music Pausing During Workout	Tester: Chaoyi Environment: Android Goal: Pause music during workout session **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Start a workout session 5. Press Pause Expected: Music pauses as workout is paused. Comments: MedRhythms Testing Status
VT0010	RQ-4.3 - Music Cancelling During Workout	Tester: Chaoyi Environment: Android Goal: Cancel music when workout is cancelled **Procedure:** 1. Open MedRhythms app 2. Enable all permissions 3. Enter IMEI 4. Start a workout session 5. Press Cancel Expected: Music stops when workout is cancelled. Comments: MedRhythms Testing Status

4.5 System Testing

- ST0001 – User Workout
- ST0002 – Offload Data Upload
- ST0003 – User Workout Data Sync
- ST0004 – Battery Consumption & Usage
- ST0005 – Device Compatibility Testing

Test ID	**Requirement ID**	**Description and Procedure**

ST0001	RQ-4.2 - User Workout	Tester: Ineh Environment: Android Goal: Test application under background load **Procedure:** 1. Open heavy background apps 2. Launch MedRhythms app 3. Enable permissions 4. Enter 15-digit IMEI 5. Navigate to Home screen 6. Start workout by pressing Yes Expected: App is usable and responsive. Comments: MedRhythms Testing Status
ST0002	RQ-4.2 - User Workout	Tester: Ineh Environment: Android Goal: Test offline data upload **Procedure:** 1. Open MedRhythms app 2. Enable permissions 3. Enter IMEI 4. Navigate to Home screen 5. Start workout 6. Disable WiFi and Cellular Expected: Workout completes, data uploads once connection restores. Comments: MedRhythms Testing Status
ST0003	RQ-4.2 - User Workout Data Sync	Tester: Ineh Environment: Android Goal: Sync workout data **Procedure:** 1. Start workout 2. Complete workout session 3. Press Sync for past hour's health data Expected: Data syncs correctly and is recorded. Comments: MedRhythms Testing Status

ST0004	RQ-4.2 - User Workout	Tester: Ineh Environment: Android Goal: Monitor battery usage **Procedure:** 1. Record battery level 2. Start workout session 3. Complete workout 4. Sync health data 5. Observe battery consumption Expected: Battery usage remains moderate. Comments: MedRhythms Testing Status
ST0005	RQ-4.2 - User Workout	Tester: Ineh Environment: Android Goal: Test compatibility across Android versions **Procedure:** 1. Build the APK 2. Install on Android 13 and 14 devices 3. Launch app and verify full functionality Expected: App functions correctly on supported versions. Comments: MedRhythms Testing Status

5 Notes

All actual results, statuses, and comments for each test are maintained in the Excel sheet attached with this document for detailed reference.