OneHaven QA Challenge – Test Plan

This document outlines the test strategy and plan for validating the OneHaven Mock API.

The goal is to ensure all endpoints function correctly for both positive (happy path) and negative scenarios.

Objectives

- Verify that each API endpoint performs as expected according to business requirements.
- Validate authentication, device linking, rule creation, and reporting flows.
- Ensure consistent response structures, status codes, and error handling.

Scope

In Scope:

- Signup and Login endpoints
- Device linking and retrieval
- Rule creation/update and retrieval by member
- Daily reports endpoint

Out of Scope:

- UI testing
- Database verification
- Performance or load testing

Test Approach

Testing will be performed using Postman for request creation and validation, with Newman used for command-line execution and reporting.

Testing Types:

- Functional Testing: Validate that each endpoint works as intended.
- Negative Testing: Confirm proper handling of invalid inputs and unauthorized requests.
- Automation Testing: Execute API tests using Newman to ensure repeatability and consistency.

Test Environment

Environment: Postman Mock Server

Base URL: {{baseUrl}} (resolved to mock server URL)

Authentication: Bearer token used for authorized endpoints

Data Source: Mock data as defined in Postman examples

Tools and Frameworks

- Postman – for creating and managing API test cases

- Newman – for executing Postman tests and generating reports
- Excel Tracker – for documenting and tracking test cases
- GitHub – for version control and submission of deliverables

Deliverables

- Postman Collection (.json) containing all API endpoints with test scripts
- Environment File (.json) with variables such as baseUrl, auth_token, and deviceId
- Test Tracker (.xlsx) documenting test cases and results
- Newman HTML Report showing test execution results
- Test Plan (.md) document detailing the strategy and approach

Entry and Exit Criteria

Entry Criteria:
- All API endpoints are defined and accessible via the mock server.
- Environment variables are properly configured in Postman.

Exit Criteria:
- All happy path tests pass successfully.
- All negative tests return expected error responses.
- Newman report is successfully generated and reviewed.

Risks and Assumptions

- The mock API may not reflect all real-world or production behaviors.
- Network or mock server issues may temporarily prevent execution.

Schedule

Activities and Status:
- Create test cases: Completed
- Build Postman collection: Completed
- Automate tests in Postman: Completed
- Execute tests with Newman: Completed
- Prepare and submit deliverables via GitHub: In progress

References

- OneHaven QA Challenge Document
- Postman and Newman Documentation