

Quality Assurance Plan

Project Name: HealthTracker

TABLE OF CONTENTS

1. Quality Assurance Strategy
 - a. Overview
 - b. Testing Methodologies
 - c. Automated vs. Manual Testing
2. Quality Factors & Metrics
 - a. Performance
 - b. Flexibility
 - c. Efficiency
 - d. Maintainability
3. Test Plan
 - a. Test Cases
 - b. Bug Tracking

LIST OF CONTRIBUTORS

- Mehmet ESKİ
- Ece EKER

1. Quality Assurance Strategy

a. Overview

The HealthTracker application helps users manage their health by tracking medications, doctor appointments, fitness routines, and water consumption. The goal of this QA plan is to ensure the HealthTracker application functions as intended and is free of critical bugs or vulnerabilities. This will be achieved by implementing testing protocols throughout the software development lifecycle. The focus will be on ensuring both the technical functionality of the app and the usability experience of the end-user.

b. Testing Methodologies

Various testing methodologies will be employed to ensure that all aspects of the application are thoroughly tested:

- i. **Unit Testing:** Each function, method, and module within the application will be tested in isolation to confirm that it behaves as expected. Unit tests will help identify issues early in development.
- ii. **Integration Testing:** This phase will verify that the different modules of the application (e.g., fitness tracker, medication tracker, doctor appointment scheduler) work seamlessly together.
- iii. **System Testing:** After unit and integration tests, the entire application will undergo system testing to ensure it functions as intended in a real-world environment.
- iv. **Usability Testing:** User experience will be tested to ensure the application is easy to navigate, intuitive, and useful. Feedback from potential users will be gathered during beta testing.

c. Automated vs Manual Testing

i. Automated Testing

Unit tests and regression tests will be automated to ensure fast and repeatable execution across various environments.

ii. Manual Testing

Usability testing will be conducted manually by a group of end-users to assess the user interface and overall experience.

2. Quality Factors & Metrics

a. Performance

Description: The application's ability to handle a large number of simultaneous users while maintaining a responsive interface.

Measurement Metric: Average response time in milliseconds (ms) for key actions such as submitting health data, scheduling appointments, or updating fitness goals.

b. Flexibility

Description: Flexibility is the ability of the application to adapt to changing requirements or evolving business needs without requiring extensive rework.

Measurement Metric: Number of Features or Updates Implemented with Minimal Code Changes

c. Efficiency

Description: Efficiency refers to the optimal use of system resources (CPU, memory, bandwidth) while performing tasks.

Measurement Metric: Memory Usage (MB)

d. Maintainability

Description: The ease with which the application code can be updated, maintained, or debugged, ensuring smooth operation and scalability.

Measurement Metric: Code complexity score that reflects how easy or difficult the code is to maintain and enhance.

3. Test Plan

a. Test Cases

i. Test Case 1: User Login and Authentication

Description: Verify that the user can log into the application securely using valid credentials and is rejected for invalid credentials.

Expected Outcome: The system should allow login with valid credentials and reject invalid login attempts.

Test Steps:

- Navigate to the login page.
- Enter valid credentials and verify successful login.
- Enter invalid credentials and verify the error message displayed.

ii. Test Case 2: Medication Reminder Functionality

Description: Test if the application sends medication reminders based on the schedule set by the user.

Expected Outcome: The system should send notifications or alerts to the user at the scheduled time.

Test Steps:

- Set a medication reminder for a specific time.
- Verify that the application sends the reminder at the set time.

iii. **Test Case 3: Fitness Goal Tracking**

Description: Test whether the fitness tracking feature correctly records exercise data and updates the user's progress.

Expected Outcome: The application should correctly track exercise input and display updated progress on the user's dashboard.

Test Steps:

- Log a new workout session with exercise data.
- Check if the data appears correctly in the user dashboard.

iv. **Test Case 4: Water Consumption Tracker**

Description: Test if the water consumption tracker accurately logs the daily intake.

Expected Outcome: The application should log each water intake entry and display an updated total for the day.

Test Steps:

- Add a water consumption entry for a given time.
- Verify that the entry updates the total consumption for the day.

b. Bug Tracking

Bugs will be reported and tracked using GitHub Issues. The process will follow these steps:

- Bug Reporting:** Testers will report bugs through the bug tracking tool, classifying each issue by severity (Critical, Major, Minor).

- ii. **Bug Verification:** The development team will resolve reported issues, after which QA will verify the fixes.
- iii. **Bug Closure:** Once verified, the bug will be closed, and the issue will be marked as fixed.

Task Matrix

Ece EKER	Quality Assurance Strategy Quality Factors and Metrics
Mehmet ESKİ	Test Plan