

Software Test Plan for Mindful Life Companion (Group #2)

1. Introduction

1.1. Purpose

This document outlines the software testing methodologies and approaches to ensure the functionality, integration, and the performance of the Mindful Life Companion app, an android based phone app for mental health tracking and management.

1.2. Scope

The testing encompasses all critical functionalities of the Mindful Life Companion App, including create, read, update, and delete operations for the Mood Tracker and Journaling Features, database accuracy and reliability, and

1.3. References

[1] External API for Quote Generation

...

2. Test Strategy

2.1. Objective

To identify any defects within the Mindful Life Companion app, in order to ensure error-free user experience that promotes mental wellness.

...

2.2. Approach

A combination of manual and automated tests will be used to validate individual components, system tests, and apps performance under a large collection of user data.

2.3. Tools

-Android Studios IDE for android app development

-Firebase for database

-GitHub for version control and collaboration

...

2.4. Environments

Separate environments will be set up for the development, testing, staging, and production phases of the MindfulLifeCompanion app to ensure isolated and focused testing.

...

2.5. Entry and Exit Criteria

Entry:

- The test environment is set up with all necessary configurations specifically tailored for the MindfulLifeCompanion app.

- Sufficient test data is available to conduct the testing activities.

...

Exit:

- All test cases are executed and their results are documented.
- Any critical defects identified during testing are resolved and closed.
- The test summary report is reviewed and approved by relevant stakeholders.

...

Scope of Testing

3.1. Functional Testing

3.1.1 Unit Testing:

Developers perform unit testing on individual components of the MindfulLifeCompanion application in isolation to ensure they work as intended. This occurs continuously during each development sprint.

3.1.2 Integrated Unit Testing:

Developers collaborate to verify the interactions and interfaces between integrated components of MindfulLifeCompanion to ensure they work together correctly. This will be performed as each component from a feature is integrated with other components, towards the mid to end of each development sprint after individual unit tests.

3.1.3 System Integration Testing:

The development team verifies that the entire MindfulLifeCompanion system, comprising multiple components such as the Mood Tracking, Medication Reminder, and Journaling, functions correctly and meets the specified requirements. This is conducted as completed features are integrated, towards the end of a planning increment.

3.1.4 User Acceptance Testing (UAT):

The Product Owner identifies a group of users, including novice and expert individuals focusing on mental wellness, to perform UAT for the MindfulLifeCompanion app. This determines whether the system meets the set requirements and expectations. This is performed after system integration testing and before a release.

3.1.5 Regression Testing:

As new features are added or existing features are modified in the MindfulLifeCompanion app, testers ensure that recent changes have not negatively affected existing features. The regression test set will run as part of an automated Continuous Integration/Continuous Delivery (CI/CD) pipeline.

3.1.6 Smoke Testing:

Testers conduct a preliminary test to check the main functionalities of the MindfulLifeCompanion app. This is performed in the test environment at the beginning of each testing phase and in the production environment after each software release.

3.2 Non-Functional Testing

3.2.1 Coding Standard Testing:

Developers run these tests as they code the MindfulLifeCompanion app to ensure adherence to company coding guidelines and standards.

4. Consideration of Infrastructure

4.1. Server Configuration

Server configuration included with Firebase

...

4.2. Database

Firebase allows for app to interact with with cloud database through the app directly, eliminating need for server management

5. Risks or Mitigation Plan

5.1. Risks

Data accuracy during CRUD operations

Usage pricing in relation to firebase services

Reliance on Google's infrastructure and server up time

...

5.2. Mitigation

Evaluate trade offs between using Firebase and other database services to ensure app profit and longevity is worth it

Synchronization strategies to ensure data consistency in case of user internet disruption

Optimize data queries and minimizing unnecessary data transfers

...

6. Resourcing

6.1. Team Composition

Test Lead

Quality Assurance Analysts

Tools and infrastructure specialists

Performance Testers

...

7. Milestones and Deliverables

7.1 Milestones:

2/24-3/1 Sprint 24.1.2

- Review User stories and acceptance criteria
- Validate features are acceptable and testable

3/2-3/8 Sprint 24.1.3

- Test UI elements and its interactions with sprint 1 and 2s features
- Periodic database verification and maintenance
- Unit Testing

3/9-3/15 Sprint 24.1.4

- Regression Testing and Smoke testing for initial features developed up to this point
- Performance Testing

3/16-3/22 Sprint 24.1.5

- Periodic database verification and maintenance
- Functional testing

3/23-3/29 Sprint 24.1.6

- User Acceptance Training
- System Integration Testing
- Further Regression Testing and Smoke testing for all features
- Load Testing

3/30-4/5 Sprint 24.2.1

- Bug fixing and issue resolution based on feedback from previous testing phases
- Refinement of user interface based on user acceptance training
- Security testing and vulnerability assessment

4/6-4/12 Sprint 24.2.2

- Integration testing with external systems and APIs
- Accessibility testing to ensure compliance with accessibility standards
- Usability testing to gather user feedback on the application's ease of use
- Documentation updates to reflect any changes or additions made during the sprint

4/13-4/19 Sprint 24.2.3

- User acceptance testing with a focus on finalizing the application for release
- Performance tuning and optimization based on load testing results

4/20-4/26 Sprint 24.2.4

- Bug fixing and issue resolution based on user acceptance testing
- Regression testing to ensure no new issues have been introduced
- Finalizing documentation, including user guides and technical specifications

4/27-5/3 Sprint 24.2.5

- Final bug fixing and issue resolution based on user acceptance testing
- Conducting a final round of regression testing to ensure no new issues have been introduced
- Finalizing documentation, including user guides and technical specifications

...

7.2. Deliverables

- Bug reports documents with bug fixes documentation
- Reports of completed and accepted tests

...