**Fitness Application**

**Test Plan**

**Version 1.0 (Draft)**

**Created By**

**Kumar Yadav**

[1. Introduction 3](#_Toc140917722)

[1.1 Synopsis 3](#_Toc140917723)

[1.2 Scope 3](#_Toc140917724)

[1.3 Objectives 3](#_Toc140917725)

[2. Test Environment & Hardware, Scope 3](#_Toc140917726)

[2.1 Test Environments 3](#_Toc140917727)

[2.2 Hardware 3](#_Toc140917728)

[2.3 Testing Scope 4](#_Toc140917729)

[3. Assumptions 4](#_Toc140917730)

[4. Testing Types 4](#_Toc140917731)

[4.1 Functional Testing 4](#_Toc140917732)

[4.2 UI Testing 4](#_Toc140917733)

[4.3 Usability Testing 4](#_Toc140917734)

[4.4 Performance Testing 5](#_Toc140917735)

[4.5 Security Testing 5](#_Toc140917736)

[4.6 Automation Testing 5](#_Toc140917737)

[5. Use Cases coverage 5](#_Toc140917738)

[5.1 Functional Use Cases 5](#_Toc140917739)

[5.2 Performance Use cases 6](#_Toc140917740)

[5.3 Usability use cases 6](#_Toc140917741)

[6. Test Execution strategy 6](#_Toc140917742)

[6.1 Test Environment 6](#_Toc140917743)

[6.2 Hardware Requirements 6](#_Toc140917744)

[**6.3** **Manual Testing** 6](#_Toc140917745)

[**6.4 Automation Testing** 6](#_Toc140917746)

[7.0 Risk Assumptions 7](#_Toc140917747)

[8. Entry and Exit Criteria 7](#_Toc140917748)

[8.1 Entry Criteria: 7](#_Toc140917749)

[8.2 Exit Criteria: 7](#_Toc140917750)

[9.1 Test Summary 8](#_Toc140917751)

[9.2 Test Reporting 8](#_Toc140917752)

[10.0 Defect Tracking and Management 8](#_Toc140917753)

[11.0 Test Closure 8](#_Toc140917754)

# 1. Introduction

## 1.1 Synopsis

The intent of this test plan is to encapsulate and ensure that the fitness tracking mobile app features, functionalities, application business flows meet its requirements, and also it is user-friendly, Security and performs well under the expected user load.

## 1.2 Scope

The scope is to perform the testing of the Fitness application to cover the functional, usability, Security, responsiveness, and performance testing of the fitness application on the defined environments, devices, Different Operating Systems like iOS, Android

## 1.3 Objectives

The objectives of this test plan are to:

* Thoughtful and end-to-end testing of the application business flows and features
* Identify defects and have a regression to ensure the core & previously running functionalities of the app remain intact after fixing the defects
* Evaluate the application is user-friendly and easy to use
* Evaluating the application performance under the various defined user load
* Evaluating the Responsiveness of the application in Orientation positions, connecting with 3rd party apps, social media apps like Facebook etc.
* Evaluate the Security features like Data encryption, Data sharing etc.
* Provide a comprehensive report of the testing results
* Collaborate with the development team to prioritize, resolve defects, defect triage

# 2. Test Environment & Hardware, Scope

## 2.1 Test Environments

The fitness mobile application will be tested in the test environments using the testing tools like Appium, Android Studio using Emulators:

* Testing environment
* Integration/SIT environment
* Actual mobile device

## 2.2 Hardware

The fitness application mobile app will be tested on the following hardware:

* Android: version 9.0 or above
* iOS: version 11.0 and also on wearables devices
* Android Studio
* Xcode: version 12.0 or later
* Appium: version 1.20 or later
* Java: version 8 or later
* Physical mobile devices: iOS, Android
* Eclipse

## 2.3 Testing Scope

Test scope of the Fitness application is defined as below:

* Ability for users to log workouts, including exercises, sets, and reps.
* Ability for users to track progress over time and set goals for fitness and nutrition.
* Ability for users to connect with friends and share progress and achievements.
* Ability for users to access a library of exercises and workout plans.
* Ability for users to integrate with other fitness apps and devices.

# 3. Assumptions

The following are the assumptions considered at every cycle/every phase/code drop will be evaluated:

* All the requirements related of every code drop / phase are developed
* Testing environments are Up and running with latest application apk
* All required Test data are created well in advance
* Open end queries / gaps are clearly marked-out and communicated to customer
* Final list of Screen size to be tested are already communicated.

# 4. Testing Types

## 4.1 Functional Testing

Functional testing will be conducted to ensure that the Fitness application features, functionalities, business work-flow as per the requirements on Actual devices, Emulators. The following functional testing will be conducted:

* Validation of the log functionality of daily workouts, other various exercise performed with sets, reps.
* Validating the Sign-Up, Sign-In, Logout functionality.
* Validate the exercise data update in the application and tracking against the goal set for fitness & nutrition
* Validating the report publishing with defined intervals like weekly, monthly etc.
* Validating the Notification functionality
* Validating the Social network functions like Connecting with friends and sharing progress and achievements, Integration with cross platform connectivity.
* Accessing a library of exercises and workout plans
* Validate for any dead / disconnected pages in the flow.

## 4.2 UI Testing

In UI Testing will verify and validate the Look and Feel of the application in the various screen sizes and actual mobile devices also.

* Validate all the User details, Images, menus options are displayed properly as per the screen sizes
* Validate the scroll of the page
* Data encryption of password during login

## 4.3 Usability Testing

Usability testing will ensure that the app is user-friendly and easy to use on various parameters with various physical mobile devices. The following usability testing will be conducted:

* Touch response of the screens.
* Personalise customable of the application layout.
* Keyboard user-friendly
* Short-cut keys
* Navigation and ease of use for all features
* Clarity of instructions and labels
* User satisfaction with the overall experience

## 4.4 Performance Testing

Performance testing to ensure that the app performs well under the expected load with commonly used and critical actions. The following performance testing will be conducted:

* Load testing to simulate high user traffic
* Stress testing to simulate heavy user loads
* Response time testing to ensure timely responses
* Resource efficient availability and management when running out conditions like
  + Less battery
  + Less memory
  + Multiple activities at the same by separate users
  + Less Internet band width

## 4.5 Security Testing

Security testing to validate the security aspect of the app performs in below scenarios well under the expected load with commonly used and critical actions. The following performance testing will be conducted:

* Password is in encrypted format
* PII information is hide during sharing the progress, feedbacks with 3rd party of social media
* Unauthorized logins

## 4.6 Automation Testing

Automation testing to validate the regression / commonly used user activities on mobile app using the automation tools like Appium, Android Studio. Scripts written will be executed using the Eclipse integrated with automation tools. The following scenarios tested to be conducted:

* App login, logout functionality
* Update and saving of Set data in the app for iOS, Android mobile devices

# 5. Use Cases coverage

## 5.1 Functional Use Cases

* Users able to log workouts details on daily basis with exercises, sets, and reps details
* Users able to track progress in regular intervals and set goals for fitness and nutrition
* App notification in scenarios like Skipping exercise / goal, over exercise etc.
* Users able to connect with friends and share progress and achievements
* Users able to access a library of exercises and workout plans
* User able to integrate with other fitness apps and devices
* User able to navigate to all the menu options of the app
* validate all the error messages, notification messages, User alert messages

## 5.2 Performance Use cases

* Login in the application with 100 with set of 10 users in every 5 sec intervals
* Saving of data by 100 users at the same
* Internet speed of 2G, 4G, 5G and Mobile internet
* Resource management efficiency when less battery, memory

## 5.3 Usability use cases

* App is easy to navigate and use for all features by the user
* Display of the instructions, alerts, notification, fields and labels are clear and concise
* Switching in-between the screens
* Verify the aesthetic, look & feel every icons, images, fields, data display in the app
* Responsiveness of the application with what the user is expecting

# 6. Test Execution strategy

## 6.1 Test Environment

The fitness mobile app will be tested in the following environment:

* Operating System: Android 9.0 and above ; iOS 10 and above
* Emulator of Devices: Google Pixel 3, iPhone
* Appium Version: 1.20.2
* Java Development Kit: 1.8
* Integrated Development Environment: Eclipse
* Actual devices: Apple iOS device, Android devices

## 6.2 Hardware Requirements

* Mobile devices (smartphones and tablets) with Android Version 9.0
* A desktop or laptop computer running the appropriate development environment, such as Xcode for iOS or Android Studio for Android
* USB cables for connecting mobile devices to the computer for debugging and testing purposes
* Adequate storage space for storing test data and results

### **Manual Testing**

Manual testing will be performed manually by the Testing team using the Emulators and/or Actual mobile devices

* Execute the Test cases manually step-by-step and record the actual outputs
* Test the application on actual mobile devices like Android, iOS, Emulator
* Extensively test all the business functional flow, features
* Log the defects in the Test management tool and re-testing once fixed
* Verify the Data retrieval, data update in the App.

### **6.4 Automation Testing**

* Automated testing will be performed using Appium and the Java programming language.
* Test scripts will be developed using Appium, Android Studio and Java to automate the test cases.
* The automated tests will be executed on the devices listed in the Test Environment section.

### **6.5 Performance Testing**

* Measuring the Throughput of the application when users load increased gradually.
* Performing the multiple activities by multiple set of users parallel on various devices.
* Monitor the application performance when scale down of the resources.

# 7.0 Risk Assumptions

There are few commonly considered Risks come across during the testing phases. So, we have to take those it into consideration.

* Technical issues with the mobile app, OS, such as compatibility issues with certain devices, that could impact the testing process.
* Time constraints that limit the amount of testing that can be performed, leading to the possibility of undiscovered bugs.
* Risks associated with user data storing, sharing and handling, also the possibility of data breaches or unauthorized access.
* Integrating with 3rd party app or devices, which could lead to unexpected behaviour and errors.
* Users may interact with the app in unexpected ways, leading to scenarios that were not anticipated during testing.
* Performance issues with the app's, such as slow load times or crashes, that could impact the user experience.
* Environment unavailability, might stall the Testing activity till it get restored back.

# 8. Entry and Exit Criteria

# 8.1 Entry Criteria:

* The SRS document is approved by the client and shared with Teams
* All the required Hardware, Mobile Operating System, Mobile device with their screen size is clearly carved out and approved by respective authority
* Timelines of the code drops for testing are finalized
* Required test data and inputs are available
* Test team trained on the Fitness application usage

## 8.2 Exit Criteria

* All the planned test cases have been executed with all critical, major functional flows are test.
* The test results are analysed and all critical, high and medium severity defects are resolved.
* All the Test mobile devices are factory reset once all testing is completed.
* Publishing of the open defects and a well approved plan in place to fix and release it.
* The performance and usability benchmarks are met as per the defined requirements.
* The test report is generated and shared with the stakeholders.
* The fitness app is ready for release.

**9.0 Test Summary & Reporting**

# 9.1 Test Summary

At the end of the testing phase, a Test Summary Report will be created by the QA

Engineer to provide an overview of the testing process and results which will include the following details:

* Summary of the testing activities and their outcomes
* Metrics on the number of test cases executed, passed, failed, and blocked
* List of defects identified during testing along with their severity, priority, and status
* Recommendations for improving the quality and user experience of the fitness-tracking app
* Overall assessment of the fitness app's readiness for releases

The Test Summary Report will be shared with the development team, project stakeholders, and management for their review and feedback.

# 9.2 Test Reporting

Test results will be documented in a test report, which will include the following:

* Summary of test execution results, including number of tests executed, passed, failed, and blocked.
* Detailed description of any issues or defects found during testing, including steps to reproduce and severity.
* Suggestions for improvement and recommendations for addressing any defects or issues found during testing.
* Any additional feedback or comments related to the testing process.
* The test report will be shared with the development team on a regular basis to ensure that they are aware of the progress of testing and any issues that need to be addressed.

# 10 Defect Tracking and Management

* During the testing phase, defects and bugs will be identified by the QA Engineer and logged in the defect tracking system.
* The development team will be responsible for resolving the defects, also performing Unit testing. Status of each defect will be tracked in the defect tracking system and communicated to the project stakeholders.

# 11 Test Closure

* Once the testing phase is complete, the QA Engineer will review the testing process and results to identify areas for improvement.
* Test Closure Report will be created to summarize the testing activities and outcomes, and provide recommendations for future testing efforts. It will be shared with the development team and project stakeholders for their feedback and review.

**Revision history**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** | **Reviewer** | **Approver** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |