Test Plan

(Full Body Workout at Home)

Objective

The objective of this test plan is to ensure that the Full Body Workout Android App meets all the functional requirements, provides a user-friendly experience, is secure, and performs well under various load conditions.

- Android Studio IDE
- Java
- XML
- SQLite Database
- Google Firebase

Scope

The scope of this test plan includes the following areas:

Navigation Drawer functionality
Various 21 Days Challenge module functionality which is available on Home Screen
Unity Ads implementation functionality
BMI (Body Mass Index) calculator functionality
User workout history functionality
Workout activity functionality

The criteria that will be used to evaluate the success of the testing, such as the number of defects found, the time taken to complete the testing, and user satisfaction ratings.

The roles and responsibilities of the team members involved in the testing, such as the test lead, testers, and developers.

The schedule and milestones for the testing, including the start and end dates, and the planned testing activities.

The tools and equipment that will be used for testing, such as testing software, hardware, and documentation templates.

Inclusions

The following items are included in this test plan:

Test strategy document
Test cases document
Test execution report
Defect report
Performance test report

Test Environments

The following test environments will be used:

- Development environment
- Test environment
- Production environment

The **Android Operating Systems** and versions that will be used for testing, such as Android 12, Android 10, Oreo.

The device types and screen sizes that will be used for testing, such as Samsung, Oppo, Redmi and Oneplus

The network connectivity and bandwidth that will be available for testing, such as Wi-Fi, cellular, or wired connections.

The hardware and software requirements for running the test cases, such as a specific processor, memory, or storage capacity.

The security protocols and authentication methods that will be used to access the test environment, such as passwords, tokens, or certificates.

The access permissions and roles of the team members who will be using the test environment, such as testers, developers, or stakeholders.

Name	Environment
QA	Android Studio IDE
Pre Prod	Android Studio IDE
UAT	Android Studio IDE
Prod	Android Studio IDE

- Android Mobile OS – Chrome

Defect Reporting Procedure

The criteria for identifying a defect, such as deviation from the requirements, user experience issues, or technical errors.

The **steps for reporting a defect**, such as using a designated template, providing detailed reproduction steps, and attaching screenshots or logs.

The **process for triaging and prioritizing defects, s**uch as assigning severity and priority levels, and assigning them to the appropriate team members for investigation and resolution.

The **tools and systems** that will be used for tracking and managing defects, such as a defect tracking software or a project management tool.

The **roles and responsibilities of the team members** involved in the defect reporting process, such as testers, developers, and the test lead.

The **communication channels a**nd frequencies for updating stakeholders on the progress and status of defects.

The metrics and metrics that will be used to measure the effectiveness of the defect reporting process, such as the number of defects found, the time taken to resolve them, and the percentage of defects that were successfully fixed.

Defect Process	**POC**
Frontend	Rahul
Backend	Rahul
Mobile Ads Integration	Rahul

Tools - MS EXCEL

Test Strategy

Component	**Description**
Objectives	1. Ensure App Functionality and Usability 2. Validate UI/UX and User Interactions 3. Validate Compatibility on Different Devices 4. Evaluate User Reminders and Notifications 5. Validate Offline Functionality 6. Validate 21 Days Challenge Functionality 7. Evaluate Video Trainer Functionality 8. Validate App Stability and Robustness

Test Levels	Unit Testing, Integration Testing and System Testing.
Test Types	Functional Testing, UI/UX Testing, Performance Testing, Security Testing, Compatibility Testing and Stability Testing.
Test Techniques	Black-box Testing, White-box Testing, Exploratory Testing, Usability Testing, Load Testing, Device Fragmentation Testing, Video Playback Testing and Offline Testing.
Test Deliverables	Test Cases, Test Scripts, Test Data, Test Execution Report
Test Environment	Emulators, Physical Devices, Android Studio IDE, Various Android Versions, Different Devices and Various Network Conditions
Test Schedule	Parallel with development cycle.
Resource Allocation	Rahul Kawatghare as Android Developer, Software Tester.

Risk Management	Changes in Requirements, Technical Issues, Compatibility Issue, Offline Mode Failures and Unexpected App Behaviour.
Test Exit Criteria	No Critical and High Priority Defects, No Critical GUI Issues, App Works on Major Devices and Issues, App Functions Smoothly in Offline Mode, Smooth Video Playback and Consistent Behavior in All Scenarios.

The first **step** is to create test scenarios and test cases for the various features in Scope.

While developing test cases, we'll use a number of test design techniques.

- Equivalence Class Partition
- Boundary Value Analysis
- Decision Table Testing
- State Transition Testing
- Use Case Testing
- Exploratory Testing
- Risk-Based Testing
- Regression Testing
- Usability Testing
- Security Testing

We also use our expertise in creating Test Cases by applying the below:

- Error Guessing
- Exploratory Testing
- We prioritize the Test Cases

Step 2: Our testing procedure when we receive a request for testing:

• First, we'll conduct smoke testing to see if the various and important functionalities of the application are working.

• We reject the build, if the Smoke Testing fails and will wait for the stable build before performing in depth testing of the application functionalities.

• Once we receive a stable build, which passes Smoke Testing, we perform in depth testing using the Test Cases created.

• Multiple Test Resources will be testing the same Application on Multiple Supported Environments simultaneously.

We then report the bugs in bug tracking tool and send dev. management the defect found on that day in a status end of the day email.

As part of the Testing, we will perform the below types of Testing:

- Smoke Testing and Sanity Testing
- Regression Testing and Retesting
- Usability Testing, Functionality & UI Testing
- We repeat Test Cycles until we get the quality product.

Step3 – We will follow the below best practices to make our Testing better:

• **Context Driven Testing** – We will be performing Testing as per the context of the given application.

• **Shift Left Testing** – We will start testing from the beginning stages of the development itself, instead of waiting for the stable build.

• **Exploratory Testing** – Using our expertise we will perform Exploratory Testing, apart from the normal execution of the Test cases.

• **End to End Flow Testing** – We will test the end-to-end scenario which involve multiple functionalities to simulate the end user flows.

Test Schedule

Following is the test schedule planned for the project –
Task Time Duration

Task	Dates
▪ Creating Test Plan	10 Dec 2023
▪ Test Case Creation	12 Dec 2023
▪ Test Case Execution	15 Dec 2023
▪ Summary Reports Submission Date	26 Dec 2023

2 Sprints to Test the Application

Test Deliverables.

The following are to be delivered:		
Deliverables	**Description**	**Target Completion Date**
Test Plan	Details on the scope of the project, test strategy, test schedule, resource requirements, test deliverables and schedule	10 Dec 2023
Functional Test Cases	Test cases created for the scope defined	12 Dec 2023
Defect Report	Detailed description of the defect identified along with screenshots and steps to reproduce on a daily basis	22 Dec 2023
Test Summury Report	Summery Reports- Bugs by bug Bugs by Functional area Bugs by Priority	26 Dec 2023

Entry and Exit Criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:
Requirement Analysis

Entry Criteria:

• Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

• List of Requirements are explored and understood by the Testing team
• Doubts are cleared

Test Execution

Entry Criteria:

• Test Scenarios and Test Cases Documents are signed-off by the Client
• Application is ready for Testing

Exit Criteria:

• Test Case Reports, Defect Reports are ready

Test Closure

Entry Criteria:

• Test Case Reports, Defect Reports are ready

Exit Criteria:

• Test Summary Reports

Tools

The following are the list of Tools we will be using in this Project:

- Android Studio
- Emulators
- Physical Devices
- Appium
- TestNG
- UI Automator
- Load Testing Tools
- Crashlytics
- Word and Excel documents

Risks and Mitigations

The following are the list of risks possible and the ways to mitigate them:

Risk: Significant changes in requirements impact ongoing testing.
Mitigations: Regular communication with the development team, early involvement in requirement discussions, and maintaining flexibility in test cases.

Risk: UI/UX issues may impact user satisfaction.
Mitigations: Regular collaboration with UI/UX designers, early involvement in design reviews, and comprehensive exploratory testing.

Risk: Performance bottlenecks may affect user experience.
Mitigations: Early performance testing, continuous monitoring, and collaboration with development to address bottlenecks promptly.

Risk: Compatibility issues on specific devices.
Mitigations: Comprehensive compatibility testing, maintaining a device matrix, and prompt bug resolution for specific devices.

Risk: App not functioning correctly in offline mode.
Mitigations: Thorough offline testing, syncing mechanisms, and quick resolution of issues affecting offline functionality.

Risk: Video playback problems affecting user guidance.
Mitigations: Comprehensive testing of video playback, collaboration with video playback libraries, and quick resolution of issues.

Risk: Unpredictable app behavior in specific scenarios.
Mitigations: Scenario-based testing, exploratory testing, and quick resolution of issues affecting app behavior.

Risk: App crashes and instability negatively impact user experience.
Mitigations: Rigorous stability testing, continuous monitoring, and prompt resolution of crashes and instability issues.

Approvals

Team will send different types of documents for Client Approval like below:

- Test Plan
- Test Scenarios
- Test Cases
- Reports

Testing will only continue to the next steps once these approvals are done