* **Test Plan for Mobile App Testing**

**Prepared by:**

Name: Kella Bhargav

Date :28 June,2022.

## 1. Introduction

**1.1 Overview**

**1.2 Scope**

**1.3 References**

**1.4 Acronyms and Abbreviations**

## 2. Test Plan and Strategy

**2.1 Unit Testing**

**2.2 System Testing**

**2.3 Pass/Fail Conditions**

**2.4 Test Report**

## 3. Schedules for Testing

## 4. Risks and Assumptions

**4.1 Risks**

**4.2 Assumptions**

## 5. Entry and Exit Criteria

**5.1 Entry Criteria**

**5.2 Exit Criteria**

## 6. Test Metrics

## 7. Logging Tests and Reporting

## 8. Roles and Responsibilities

**8.1 Project Manager**

**8.2 Test Lead**

**8.3 Test Engineer**

## 9. Deliverables

## 10. Approvals

## 1. Introduction

Mobile testing is required to ensure that a website or application runs perfectly,no matter the circumstances.

**1.1 Overview**

A Test Plan is a document that describes the scope of testing, test strategy, objectives, effort, schedule, and resources required. It serves as a guide to testing throughout the development process. Find an example of the Functional Test Plan .

**1.2 Scope**

The scope of work is defined at the beginning of the testing process. A project team should clearly understand what features and functions there are to be tested and which ones are out of scope. To determine the scope of testing, the project specification, budget, and customer’s requirements should be taken into account.

**1.3 References**

This section contains the documents which support the test plan and can be referred to during the testing process. Commonly, the list involves:

* the project plan
* system requirements
* specification design documentation

**1.4 Acronyms and Abbreviations**

All the acronyms used in the Test Plan require explanation. For example,

* DDD - Database design document
* QMS - Quality Management System
* SLC - Software life cycle

## 2. Test Plan and Strategy

**2.1 Unit Testing**

2.1.1 Objective

The main objective of unit testing is to verify whether every unit operates as intended. A function, procedure, method, or even the entire module can be considered a separate unit. Unit testing can be conducted manually, but automated testing is a more common practice.

**2.1.2 Entry Criteria**

* the planning phase has been finished
* testable units are available
* all functional requirements have been defined
* the unit testing environment has been set up

**2.1.3 Exit Criteria**

* all planned test cases have been covered
* all the bugs found have been reviewed
* the performance of key modules has been tested

**2.1.4 Logging Tests and Reporting**

Developers fix the defects found in each unit. If these defects are related to other modules and units, they must be reported.

**2.2 System Testing**

**2.2.1 Objective**

System testing is generally conducted after Unit Testing. The objective of System Testing is to evaluate compliance of an integrated application with its requirements.

**2.2.2 Testing Procedure**

* test cases preparation
* test executions
* bug reporting

**2.2.3 Types of System Testing**

**2.2.3.1 Performance Testing**

Performing testing is conducted to detect issues related to:

* memory consumption
* power utilization
* network connectivity
* operating in the background
* switching between applications
* memory leakage

**2.2.3.2 Interrupt Testing**

This testing type examines how an application reacts to interruption and resumes to its previous state. There are numerous reasons that can potentially interrupt the operation of an app, such as getting a phone call, messages, notifications, battery low, etc.

**2.2.3.3 Usability Testing**

Usability testing is performed to check whether the application is easy to use and understand for the end-user.

**2.2.3.4 Installation and Launch testing**

Installation testing aims to detect whether there are any issues during the installation, uninstallation, and updating process. Once the application has been installed, a QA engineer also checks the launching process.

**2.2.3.5 Functional Testing**

All the functions and features of the application are tested to verify whether they operate according to the specification.

**2.2.3.5 Security testing**

Security testing is conducted to find the application vulnerabilities and prevent data breaches.

**2.2.3.6 Regression testing**

Regression testing is a re-execution of tests that had been done before the code changes. Its purpose is to verify whether a new functionality has affected the existing one.

**2.3 Pass/Fail Conditions**

All the conditions when tests pass or fail are defined and described.

**2.5 Test Report**

Test Report helps to summarize testing activity in a formal way. It should contain:

* name and overview of an application
* testing hardware and software environment
* the number of test cases executed/passed/failed.

For each issue that has been encountered, the following information is provided:

* bug description
* bug status (open, fixed, etc.)
* bug location
* steps to reproduce an issue

## 3. Schedules for Testing

A test schedule, created by Project Managers, helps to monitor the testing process.

## 4. Risks and Assumptions

**4.1 Risks**

The following risks may occur during the mobile app testing process:

* availability of devices
* new features and modification which have not been planned in advance
* changes in requirements
* delays in schedule

**4.2 Assumptions**

* each release is accompanied by a note with information about implemented features and their impact on the system
* all blocker bugs receive the high priority status
* all the bugs found are fixed before the next software release
* all documents are up-to-date and delivered to the testing team in time
* all necessary equipment and tools are provided and ready for testing
* the test schedule is reviewed in case there are any obstacles for testing

## 5. Entry and Exit Criteria

**5.1 Entry Criteria**

* the development phase has been finished
* requirements have been defined and approved
* test design and tests plan have been created
* the test environment has been set up
* all necessary resources are available.

**5.2 Exit Criteria**

* tests cases are executed
* the rate of tests cases passed is satisfactory, e.g., 95%
* failed test cases are not related to crucial functionality
* tests results have been accepted
* critical defects have been fixed.

## 6. Test Metrics

Test metrics are quantitative measures that help to estimate the testing effort. The most common metrics are:

* requirement coverage
* test cases coverage
* number of tests executed
* number of defects found (taking into consideration their priorities and severities)
* tests design effort
* total test effort

## 7. Logging Tests and Reporting

Each detected issue should be properly reported using special tools and applications.

## 8. Roles and Responsibilities

Project roles and responsibilities should be clearly defined and divided among the project staff. Commonly, the roles are as follows:

**8.1 Project Manager**

* The Project Manager is responsible for managing the whole testing process. They approve all test documentation, considers budget and time terms, and provide necessary resources.

**8.2 Test Lead**

The Test Lead is responsible for collecting requirements, planning process, test activity monitoring, and project reporting.

**8.3 Test Engineer**

The Test Engineer is responsible for test case preparation and execution, as well as issue reporting.

## 9. Deliverables

The list of testing deliverables usually contains:

* test plan
* test cases documents
* test strategy
* test results
* test summary report

## 10. Approvals

Name(In Capital Letters) Signature Date

1.

2.

End.