**INTRODUCTION**

This test plan outlines the testing methodology for a few crucial components of a crowdfunding website. The document introduces:

* Test strategy
* Execution strategy
* Test report and project schedule

The test plan will be based on an assumption, as below:

* Such features are in MVP state. For example, browsing campaigns feature should work with or without pagination support so that the user can browse the projects without problem.
* The feature support responsive on mobile devices
* Resources for responsive test will be not listed in the test task
* The testing environment is ready at the point of completing coding.
* JIRA will be used as the main management tool for the team.
* Test cases, tests result will be stored in .xlsx files.
* The bug severity is categorized to 4 level: critical, major, medium, minor.

**TEST OBJECTIVES**

The objectives of the test is to verify the functionality of new features of the crowdfunding platform, including:

* Browsing campaigns (Jira Ticket – User story) (CFD-01)
* Contributing to a campaign (CFD-02)
* Sharing campaigns on social media (CFD-03)
* Starting a campaign (CFD-04)
* Tracking its progress (CFD-05)
* Communicating with contributors (CFD-06)

to make sure all features work as described in the SRS:

* The user can browse campaigns and use filters to get more interesting project
* The user can see details of a project then contribute to it
* If the user finds the project interesting, he can share it via social media
* If the user is a creator, he can start a project
* The creator can track the process of the project and see details about the project
* The creator can leave comments, reaction to comments, mention contributor and so on

**TASKS**

Task: Review the SRS when it is still in draft and notice of changes in each version, update

When to perform: at the time of changes, notices from BA

Who will perform: Tester who is in charge of the related ticket

Definition of done: clearly understand about the new features and how to test those features  
  
Task: Review the SRS notices all the change to the final of the document, to propose testing type, technique, contribute to the test strategy.

When to perform: the same as above  
Who will perform: test manager or project manager

Definition of done: Testing Scope document is done

Task: Write the test cases  
When to perform: after final version of SRS and task “Review the SRS” is done carefully to make sure tester understand clearly about the feature to be tested  
Who will perform: Tester who is in charge of the related ticket

Definition of done: test cases for the feature including functional test cases and user acceptance test cases are ready for test execution

Task: Prepare the test data, accounts, related stuffs for the incoming test, including social media account, system accounts, database accounts..  
When to perform: after all the test cases are prepared  
Who will perform: Tester, engineer, devops, manager who is in charge of the related ticket

Definition of done: test data are available and ready for tests

Task: Execute the test cases  
When to perform: after DEV is done  
Who will perform: Tester

Definition of done: all the test cases are executed in a testing round

Task: Execute the test cases on mobile devices  
When to perform: after DEV is done  
Who will perform: Tester

Definition of done: all the test cases are executed in a testing round

Task: Retest and verify defect  
When to perform: after developer has done fixing bug  
Who will perform: Tester

Definition of done: all the defects are fixed

Task: Clean the test environment, junk data, database unused records,  
When to perform: after a testing round  
Who will perform: Tester

Task: Report defects if found by posting an issue in JIRA to describe as much details about the defect, how to reproduce with test data, screenshot  
When to perform: after a testing round  
Who will perform: Tester

Definition of done: bugs are report and assigned to dev for fixing

**SCOPE OF TESTING**

These types of testing will be performed

**Functional test**

To verify that each functionality works as expected and meets the requirements. This includes testing the user interface, the business logic, the data validation, and the integration with other components.

**Why functional test is used?**

The new feature will be widely used by many types of user who already have a business or will have a business, focus on the business requirement is a must. Functional tests only focus on the business requirement and the outcome of the feature no matter how it was done.

**Exploratory testing**

Exploratory testing is not really a type of testing but an approach and proved to be effectively uncover defects that are not easily covered in the scope of other testing type.

For example:

Tester are encouraged to perform these test cases because it was not mentioned in the requirement:

* Any links in those pages that re-direct to another page should be checked for proper re-directions
* If there is any notification system, it should be checked. For example, after contributing to a campaign, an email is sent to investor. The email should be sent successfully with correct content

**User Acceptance Testing**

This test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment

With the real usage by the end user, some validations which were not included in the functional test cases will be covered.

**Note:**

Test Scope is defined in (Testing Scope.xlsx)

Test cases are defined in (challenge1\_testcases.xlsx)

**Out Of Scope**

Load testing will not be performed  
End-to-end testing will not be performed due to lack of resource

**TEST EXECUTION**

**Test circles**

* There will be 2 round of testing and it’s flexible about having one more test round if there is any critical issue occur and product owner is the one who will make a decision for any changes.
* All the test cases will be executed in each round.
* The objective of the first round is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the test cases executed.
* The objective of the second round is to identify remaining high and medium defects, remove the work-around from the first cycle

**When to start a Testing round**

* All the test cases are ready.
* The team have prepare documents, ways for tracking defects and their statuses, % of finished work, tracking related issues with team

**Bug report guidelines**

At the very first round, if there is a blocking issue which prevents tester from test execution, the tester should inform immediately by any way (chat, call, speak) to test lead , test lead should then communicate with DEV team, provide with useful information to resolve the issue. The issue should be tracked by creating a JIRA ticket for future use as a case study.

During the test execution, all the defects should be carefully noted down by:

* Time of execution (this info maybe useful info if the developer needs)
* Number of attempts to reproduce (this info is also useful)
* The most important thing: how to reproduce it.

Then list as much as details such as screenshots, input data.. with steps and special steps to produce the bug, to a JIRA ticket with this format, for example:

• Summary: Unable to share a campaign on Facebook due to an invalid URL error

• Steps to reproduce:

1. Go to a campaign details page
2. Click on the Twitter icon
3. Observe

**Actual:**

Step3: Facebook pop-up is displayed

**Expected:**

Step 3: Elon Musk face is popped-up

The ticket then should be assigned to the corresponding developer.

Link the test case and the ticket by some means

**Summary of activities when a bug occurs**

* Tester found bug → Create bug ticket → Assign to Dev (round 1)
* Dev fixed the bug → Assigned the ticket to Tester → Tester retest → Verify bug has been fixed → Close the ticket (Round 2)

**TEST REPORT, TASKS SCHEDULE**

* There will be a (QA Report.xlsx) file that managed by the Test Lead/Manager to track for the current process for each test round.
* With this file, the test manager will be able to track team workload, current activity of each team member, pass rate of test cases, block issues..