# Section A

# Question 1

# Test Plan – Carry1st

|  |  |
| --- | --- |
| **General Information** | |
| Created by (Author) | Njabulo |
| Preparation date | 04/06/2024 |
| Version | 01 |
| Status | In-progress |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Revision History** | | | | | |
| **Version** | **Description** | **Authour** | **Date** | **Approved by** | |
| **Authour** | **Date** |
|  |  |  |  |  |  |

**Summary**

1. **Introduction**

**General information**

This document describes the methods and procedures that will be used by the Carry1st team in the functional testing process of the mobile Application

It is meant to be used as a manual during testing works. It describes the procedure of the testing process. The test plan is intended for project managers, product developers, and QA engineers.

The objective of the testing activities is to check functions and features of a software product developed for mobile applications Android and iOS devices.

**Purpose**

This Test Plan document for the Gamesupports the following objectives:

● Identify existing project information and software components to be tested.

● Recommendation and description of the testing strategies to be employed.

● Identify required resources and provide a test effort estimate

● List the test project deliverable elements. The results of test execution will be sent to the stakeholders as reports. All found bugs will be tracked using the Jira board.

1. **Scope of Project**

**Scope of web portal**

Testing of application is in the scope of this test plan. The following components and functions would be tested:

**Scope of mobile application**

Testing of mobile applications is in the scope of this test plan. The following components and functions would be tested:

Functional testing

UI testing

Performance testing

1. **Work plan**

The parties agreed to follow the next work plan:

1. Test plan preparation

2. Test plan approval

3. Functional testing and bugs reporting

4. Daily reports preparation

5. Final report preparation

1. **Test Plan and Strategy**

**Functional testing**

The objective of functional testing is to make sure that the whole software product works according to the requirements, and no significant errors appear in the application. Functional testing is the most substantial part of software testing. It involves checking different aspects of the system. A software product must pass all the planned tests. Only in this case its quality can be assured.

|  |  |
| --- | --- |
| **Test Objectives: Ensure proper target-of-test functionality** | |
| **Technique:** | Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following:  ● The expected results occur when valid data is used.  ● The appropriate error or warning messages are displayed when invalid data is used.  ● Each rule is properly applied |
| **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. |
| **Completion/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. |
| **Special Considerations:** | API services |

**Test Procedure**

Test procedure assumes the next points:

● Reporting of found software bugs.

Various aspects of the tested software should be checked; this requires executing of different testing types.

● Functional Testing

● UI Testing (along with the UI/UX design team)

● Usability Testing

● Compatibility Testing (**4** modern web browsers and devices)

● Regression testing

● Sanity testing

● Retesting (during the second round if needed)

It also will be checked how the software product is run on the browsers and devices that are supposed to support it, how it starts and stops, and how much time it needs to launch.

**Bug Reports**

Bug reports are created in order to provide the development team and the project managers with exhaustive information about the discovered defects. They must be helpful in determining causes of the errors and correcting them. Follow this guideline [Defect - Capturing Process](file:///C:\wiki\spaces\~470807482\pages\9090793505\Defect+-+Capturing+Process)

**Defect Severity can be classified into four categories: To be confirmed with the rest of the teams**

● **Critical** (blocker) defects are the failure of the complete software system or of a critical subsystem, and no work or testing can be carried out after the occurrence of the defect. It also applies to data loss failures and with processes that leave inconsistent data stored in the database.

● **Major/High** defects (and crashes) are those which also cause failure of the entire or part of the system, but there are some processing alternatives which allow further operation of the system. It also applies to the system crashing, or aborting, during normal operation of a non-critical flow.

● **Minor/Medium** defects do not result in failure but cause the system to show incorrect, incomplete, or inconsistent results.

● **Trivial/Low** defects are small errors that do not affect the functionality: typos, grammar mistakes, wrong terminology, etc.

**The information that is indicated in each bug report:**

● the software product name;

● version number of the software product (**for mobile**);

● the browser on which the tests were performed.

**Each report provides the next information about the defect:**

● summary, which is short description of the problem;

● location of the defect in the software product;

● steps to reproduce the error;

● frequency of the defect occurrence;

● severity of the defect;

● additional information about the defect in the form of attached screenshots or video records.

1. **Resources**

**Tools**

The following tools will be used for all projects:

|  |  |
| --- | --- |
| **Name of process** | **Tool** |
| Defect Tracking | Jira |
| Test cases | MS Excel/ALM |
| Screenshots / Videos capture | **Snagit** |

**The list of the devices**

|  |  |
| --- | --- |
| **Name of the Operating system** | **Version** |
| Android devices | All supported OS |
| iPhone devices | All supported OS |
| Huawei devices | All supported OS |

1. **The criteria of quality**

The product should operate in accordance with the requirements and the functional specification (if present). The product should not contain critical and blocking defects in the final version of the project

1. **Testing Process Risks**

The next issues may influence testing works:

● changes and modifications of the software product that were not planned and discussed with the test team before hand;

● changes in the software requirements that were not discussed with the test team before hand.

● delays in correcting/fixing errors;

● delays in delivering new builds to the test team;

● no sufficient test data;

1. **Test Team Expectations**

The test team must be provided with valid, updated documents during the whole testing process.

All the required devices and software must be acquired and prepared before the beginning of the testing process.

All show-stopping errors must be corrected as soon as possible.

Release notes should be added to each software release to the test team. The note must explain which elements, functions and features were added to the program and how these additions affect the software.

The developers should correct all the errors in the software modules before releasing a new version.

1. **Responsibilities of Test Team Members**

**Test Manager**

● Managing the whole testing process.

● Providing all the needed resources for the testing activities.

**QA Tech Lead**

● Managing the QA team from a technical perspective.

● Proposing best practices and tools for a project.

● Reviewing Test cases and ensuring quality of these.

**QA Engineer**

● [Test Process](file:///C:\wiki\spaces\~470807482\pages\9090826261\Test+Process) / logging found errors into the approved bug tracking system.

1. **Deliverables**

● Test Plan.

● Bug reports and reports regarding the testing progress.

**Test Cases**

1. Test Case 1 - Valid Search:
   * Description: Verify that users can search for products using valid search queries.
   * Steps:
     1. Navigate to the search bar.
     2. Enter a valid search query (e.g., product name).
     3. Press Enter or click on the search button.
     4. Verify that relevant products are displayed.
   * Expected Result: Relevant products matching the search query are displayed.
2. Test Case 2 - Invalid Search:
   * Description: Verify that appropriate feedback is provided when an invalid search query is entered.
   * Steps:
     1. Navigate to the search bar.
     2. Enter an invalid search query (e.g., gibberish).
     3. Press Enter or click on the search button.
     4. Verify that a message indicating no results or a suggestion to refine the search is displayed.
   * Expected Result: User is informed that the search query did not yield any results or is provided with suggestions to refine the search.

Feature: Add to Cart

Objective: To ensure that users can add products to their shopping cart.

Test Cases:

1. Test Case 1 - Add to Cart:
   * Description: Verify that users can add products to their shopping cart.
   * Preconditions: User is logged in and on the product page.
   * Steps:
     1. Navigate to the product page.
     2. Click on the "Add to Cart" button.
     3. Verify that the product is added to the cart.
   * Expected Result: Product is successfully added to the shopping cart.
2. Test Case 2 - Update Cart Quantity:
   * Description: Verify that users can update the quantity of products in their cart.
   * Preconditions: User is logged in and has items in the cart.
   * Steps:
     1. Navigate to the shopping cart page.
     2. Change the quantity of a product.
     3. Verify that the cart reflects the updated quantity and price.
   * Expected Result: Quantity of the product in the cart is updated accordingly.

Question 2

While testing the web application. Several bugs were identified which are as follows:

* Login screen is displaying both register and login. There are different page links for login and register. Login screen should be displayed on its own screen so is register.
* Search – When search for a product the search returns products with incorrect images’ Searching for an iPhone returns fridges instead on mobile devices. The correct images should be returned
* The cell phone number textbox on the registration screen allows the user to insert text instead of numbers. The textbox should be only allowing numbers
* Product details are missing under the iPad pro banner on the home page
* Blog on the home page is written in a different language than the rest of the website.
* Images and text are missing on HPS eadphonrHhead phone.
* When add the canon EOS SD camera there is a drop-down selection box that cannot be selected.