**Homeppl Test Strategy & Plan**

# Introduction:

The goal of this project is to build services that enable users to search for computer by name and view the number computers found during the operation. And, user will also be able to edit the device details as required and save and cancel the opreation. And, following the completion of the tasks, user shall be able to view the success messages.

# Testing Goals:

This application is new and critical, so we must ensure its functional quality. To validate application functionality, all the features will be tested to ensure all functions provide the expected output. Functional testing will be complete when all features have passed all associated test cases, with no exceptions.

## 2.1. Types of testing to be completed:

The project team members will run testing consecutively to ensure the correctness of the application, covering the list of testing techniques as stated in this document and based on the needs of the projects. The implementation of test execution may vary, depends on the testing objectives. The goal is to follow the best practice and constantly improving the quality of the product under testing including enabling a quick release cycle to the business.

## 2.2. Unit Testing:

This type of testing will be performed to ensure that the smallest testable parts of this application are independently and individually tested to ensure its correctness during the software development lifecycle.

## System Integration Testing:

Testing will be performed to detect bugs and prevent bugs during the integration of different components of the software under test. This type of testing will be used to expose faults in the interaction between integrated modules.

## System testing:

This type of testing will be conducted to verify and validate the fully integrated software product under test to evaluate end-to-end application specifications. The detailed test case and test suites will be created to test each aspect of the application based on use cases and the customer's view to ensure the product meets the need of the business as specified in the requirements.

### Regression Testing:

The regression testing will be performed to test that existing features, changes, or additional features are all working as expected during the software development lifecycle. The testing team will re-execute this type of testing to prevent bugs and improve the quality of the application under test (AUT).

## Sanity Testing:

This type of testing will be executed to check that bugs have been successfully fixed in the build, immediately after the regression. This will be done manually and undocumented. And, it should be completed after the regression testing.

## User Acceptance Testing:

Acceptance testing is very crucial and may be considered as needed by the business. The goal is to test that the newly developed systems meet the need of the customers as specified in the business requirement documents. However, optional as more clarification may be needed to proceed with the workload estimate.

## Smoke Testing:

This type of testing will be conducted after the system have successfully passed the User Acceptance Testing (UAT) to ensure more bug are uncover before releasing the products to production. Automated regression testing may be used to perform the test to measures the stability of the application under test (AUT) covering end-to-end testing of all the functions of the products.

# Team Member Assignment:

The following resource will be completely and partially allocated and dedicated to the testing effort including the role each will play in the testing phases.

Table 1:

|  |  |
| --- | --- |
| **Name** | High-High-Level Testing AssignmentsLevel Testing Assig |
|  |  |
|  |  |
|  |  |

Table 2: Testing Team Member Assignment

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Test Approach & Tools:

The test will be executed with the aid of Python & PyCharm automated tool. All requirements, test cases, and project-related communication will be stored in the test management tool for monitoring and controlling purposes across the project.

Test Case template will be used as the template for creating test cases. And, all testing-related documents should be stored and archived in the project share folder to provide visibility, traceability, and knowledge share across the team.

The automated testing design model implemented should be documented and stored in a shared folder.

Table 3:

|  |
| --- |
| **Tools & Environment:**   1. Python Framework (UI) 2. Python (API) 3. GitHub (CI/CD pipeline) 4. Jenkins/TeamCity (Continuous Integration Build Tool) 5. AWS Amazon cloud |

The templates listed in the table below will be used by the testing team for creating, documenting, and communicating test cases and test results.

Table 4:

|  |  |  |
| --- | --- | --- |
| **Template Name** | **Purpose** | **Location (file path)** |
| Test case Design Template |  |  |
| Exploratory Test Case Design  Template |  |  |
| Defect Report Template |  |  |
| Test Report Template |  |  |
| Release Document |  |  |
| Traceability Matrix Template |  |  |

# Scope:

This section details the features that will be included in the functional testing phase(s) and those that will be excluded.

Table6. Features included in Testing

|  |  |
| --- | --- |
| **Feature ID** | **Name / Description** |
| 1 | The user shall be able to view the following details:.   * Computer name * Introduce date * Discontinued date * Company |
| 2 | User shall be able to search for the computer by name and view all the model of the same brand. |
| 3 | User shall be able to perform the following tasks:   * Search for computer by name * Update/edit the details save the changes * Cancel the task during the update * View the success message. |
| 4 | Application shall only allowed the date format as shown: yyyy/mm/dd (**To be completed**) |
| 5 | All mandatory field shall be completed to successfully submit the form (**To be completed**).  Note: Failure to completed the steps will result in error message (**To be determined**). |
| 6 |  |
|  |  |

Table 7: Features NOT included in Testing

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Name / Description** | **Reason for Exclusion** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Functional Requirement:

1. The user shall be able to complete and submit a form.
2. The user shall receive the first email notification message(s) after they have successfully submitted the form.
3. The user shall get a second notification message when the item has been shipped.
4. The user shall receive only one item per order (**Unclear requirement**).
5. The user should be able to order more than one product at once (**Unclear requirement**).
6. Each user shall receive the same item (**Unclear requirement**).

### Non-Functional Requirement:

* Pending clarification from the stakeholders/customers.

# Metrics

The following information will be collected from test results to develop testing metrics.

* Functional Metrics: Number of requirements verified, broken down into test phase, component, QA Engineer
* Issue Metrics:

1. Problem found per component
2. Problems found per day

* Schedule Metrics:

1. Percentage of tests completed
2. Estimated days to completion
3. Time to complete testing by component.

# Functional Test Cases

The following test cases will be created and executed against the application.

Using the template below, complete the details for each test case. Copy the template each time a new test case is created. This section can be moved to a separate document and referenced here.

|  |  |
| --- | --- |
| Test Case ID | Details |
| **Test Case ID:** | **0001** |
| Test case name: | As a user I want to search for computer by name and view the list of related brands |
| Purpose | To search for computer by name and view the list of related brands |
| Initiation Criteria | Test URL must be provided before testing and test data must be cleared from the server after the test. |
| Execution Steps | Step1 :  1. Navigate to to **http://computer-database.herokuapp.com/computers**  2. Enter the computer name  3. Click on search button  4. Verify that the related brands are displayed  5. Verify that the item count is displayed |
| Expected Results | • User should see all the available brands from the same company  • User should see the item count on the top of the page. |
| Pass/Fail |  |
| Assumption |  |

Table 8:

|  |  |
| --- | --- |
| Test Case ID | Details |
| **Test Case ID:** | **0002** |
| Test case name: | As user a user I want to view the below details:.   * Computer name * Introduce date * Discontinued date * Company |
| Purpose | To view the computer name, Introduce date, Discontinued date and the Company. |
| Initiation Criteria | Test URL must be provided before testing and test data must be loaded in the database ready for testing. |
| Execution Steps | Step1 :  1. Navigate to **http://computer-database.herokuapp.com/computers**  2. Verify that the following columns are present on the page and populated with computer data.   * Computer name * Introduce date * Discontinued date * Company |
| Expected Results | • The columns are present on the page and populated with computer data. |
| Pass/Fail |  |
| Assumption |  |

|  |  |
| --- | --- |
| Test Case Field | Details |
| Test Case ID: | 0003 |
| Test Case Name: | As a user I want to search for computer by name, edit the details and save the changes and view the success message.  And, as a user I want cancel the operation during the update. |
| Description: | As a user I want to search for computer by name, edit the details , save or cancel the operation and view the success message.  . |
| Initiation Criteria: | Test URL must be provided before testing. And, test data must be cleared from the server after the test. |
| Execution Steps: | Step1 :  1. Navigate to to **http://computer- database.herokuapp.com/computers**  2. Enter the brand name.  3. Click on the computer name..  4. Verify that you are on edit page.  5. Edit the details  6. click on save this computer button.  6. Verify that a success message is displayed.  Step 2:  1. Navigate to **http://computer- database.herokuapp.com/computers**  2. Enter the brand name.  3. Click on the computer name..  4. Verify that you are on edit page.  5. Edit the details  6. click on Cancel button.  6. Verify that a success message is displayed. |
| Expected Results | 1. Users should be able to search, edit and save the changes  2. User should be able to cancel the operation |
| Pass/Fail |  |
| Assumption: |  |

# Traceability Matrix

The traceability matrix relates the test cases to their respective requirements tested. Not all test cases will have a direct relationship, but all the use cases for the features in the scope should have associated test cases.

Traceability matrix (Link included).

**Automated Testing Frameworks**

Python Framework (To be continued)

Python API DESIGN MODEL (To be continued)