[Project Name]

**To put in a client name select all (ctrl A) then f9Code Star Academy – Kaopiz Software**

**Automation Test Plan**

**Created by: Hoan Tran**

**Version: 1.0**

**December, 01st , 2022**

**Table of Contents**

[1. INTRODUCTION 3](#_Toc201168091)

[2. TEST OBJECTIVES 3](#_Toc201168092)

[3. TEST STRATEGY 3](#_Toc201168093)

[*3.1.* *Scope of Testing* 3](#_Toc201168094)

[3.1.1. In scope testing 3](#_Toc201168095)

[3.1.2. Out of scope testing 3](#_Toc201168096)

[*3.2.* *Test Levels and Test Types* 4](#_Toc201168097)

[*3.2.1.* *Test levels* 4](#_Toc201168098)

[*3.2.2.* *Test types* 4](#_Toc201168099)

[*3.3.* *Risks* 5](#_Toc201168100)

[4. TEST CRITERIA 5](#_Toc201168101)

[*4.1.* *Entry Criteria* 5](#_Toc201168102)

[*4.2.* *Suspension Criteria* 6](#_Toc201168103)

[*4.3.* *Exit Criteria* 6](#_Toc201168104)

[5. RESOURCE PLANNING 6](#_Toc201168105)

[*5.1.* *System resource* 6](#_Toc201168106)

[*5.2.* *Test Resource* 6](#_Toc201168107)

[6. SCHEDULE AND ESTIMATION 7](#_Toc201168108)

[7. TEST DELIVERABLE 7](#_Toc201168109)

[8. AUTOMATION TESTING TOOL AND STRATEGY 9](#_Toc201168110)

[*8.1.* *Testing process* 9](#_Toc201168111)

[*8.2.* *Automation strategy* 9](#_Toc201168112)

[*8.3.* *Automation testing framework* 9](#_Toc201168113)

[9. LIFECYCLE OF BUG AND RISK 10](#_Toc201168114)

1. **INTRODUCTION**

The purpose of the project is to build the system that *[project description]*.

The Test Plan determine the scopes, objectives and risk of testing. Defining the overall test approach, the overall framework, environment that will support test activities, resolution for risk and contingency, test criteria, timeline and resource planning.

Build automation test scripts which using for regression test phase.

1. **TEST OBJECTIVES**

* Ensure that system functionality works as expected without any critical, high bugs
* Ensure that the response time when using system meets expectation
* Ensure that the integration between modules and systems work as expectation

1. **TEST STRATEGY**
   1. *Scope of Testing* 
      1. *In scope testing*

In scope items will be system testing as usability of bellow features.

The following is the list features to be tested in this project:

|  |  |  |
| --- | --- | --- |
| No. | UserStory/Function | Degree of complex |
| 1 | Bếp từ – Lọc theo giá | 2 |
| 2 | Bếp từ – Lọc theo số bếp | 2 |
| 3 | Bếp từ – Lọc theo phân loai | 2 |
| 4 | Bếp từ – Lọc theo hãng sx | 2 |
| 5 | Bếp từ – Lọc theo xuất xứ | 2 |
| 6 | Bếp từ – Lọc theo nhiều mục | 2 |
| 7 | Bếp từ – Phân loại theo tab giảm giá nhiều | 2 |
| 8 | Bếp từ – Phân loại theo tab giá thấp | 2 |
| 9 | Bếp từ – Phân loại theo tab giá cao | 2 |
| 10 | Bếp từ – Mua ngay | 1 |
| 11 | Bếp từ – Mua trả góp | 2 |

Degree: 1 – Complex, 2 – Medium, 3 – Low

* + 1. *Out of scope testing*

There are some out of scope items:

* Database test
* Mobile test
* Performance test
* Security test
* API test
  1. *Test Levels and Test Types*
     1. *Test levels*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test level** | **Method** | **People in charge** | **Note** |
| 1 | Integration test | Manual/Automation | Tester | * Testing the integration of systems and packages; testing interfaces to external organizations * Test to find defects in the interfaces and in the interactions between integrated components or systems |
| 2 | System test | Manual/Automation | Tester | Verify that the whole system meets with requirements |
| 3 | User acceptance test | Manual/Automation | Client | Client test the system to make sure that the system is met the requirement. To make sure the useablitiy, maintanceability, reliablity. |

* + 1. *Test types*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test type** | **Method** | **People in charge** | **Notes** |
| 1 | Review requirement (static testing) | Manual | Tester | To verify the correction and level of detail of User Story |
| 2 | Review code | Automated/Manual | Dev team | Self-review, Peer Review |
| 3 | Functional testing (dynamic testing) | Automated/Manual | Tester/Dev team | Verify that the system meets functional requirement |
| 4 | Non-functional testing | Manual | Tester | Performance test, API test, Security test |
| 5 | Re-testing | Manual | Tester/Dev team | Testing after fixing bug to confirm the bug is removed |

* 1. *Testing risks*

|  |  |  |
| --- | --- | --- |
| **No.** | **Risk** | **Mitigation** |
| 1 | May have high number of users | * Do load test, stress test to ensure system load |
| 2 | Project schedule is short, it may be hard to complete testing on time | * Set priority for test activities: focus testing on the core features that contains many critical bugs * Writing test checklist instead of test cases step by step if not enough time |

1. **TEST CRITERIA**
   1. *Entry Criteria*

The entry criteria refer to the conditions in order to start testing activities (test design, test execution):

1. The specification of a feature is approved and available on Jira before starting test design
2. Development completed (status must be Ready for test)
3. Features under test are already on QC/test environment
4. Test cases, test data are peer-review before test execution
   1. *Suspension Criteria*

The suspension criteria refer to the conditions that test activities will be suspended if suspension criteria are met during testing:

1. Any main flows are not covered
2. Main feature does not meet acceptance criteria
   1. *Exit Criteria*

The exit criteria are the targeted results of the test that need to be met to complete a test phase:

1. 100% of Test cases executed
2. No Critical, high, medium bug remaining
3. 5 % leakage (total bugs found by client / total bugs) : lọt lỗi sang khách hàng: khách hàng tìm được 5 bugs / 100 bugs team dự án tìm
4. **RESOURCE PLANNING**
   1. *System resource*

|  |  |  |
| --- | --- | --- |
| No. | Resources | Description |
| 1 | Server | * Servers: *[list out if have]* |
| 2 | Test tool | * Snagit: Provide evident for bugs * Jira, Zephyr: Task, bug tracking, Quality monitoring, Test case management, Test execution |
| 3 | Computer | * The PCs with hardware, software required |
| 4 | **Mobile device** | * *[list out iOS or android version if have]* |
| 5 | **Web browser** | * *[list out Browser version if have]* |

* 1. *Test Resource*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Start date** | **End date** |
| Tran Thi Hoan | Tester work 50% | 25-Nov, 2021 | 10-Jan, 2022 |
| Ho Tran Nhat Anh | Tester work 100% | 25-Nov, 2021 | 10-Jan, 2022 |
| Pham Thi Dinh | Tester work 100% | 25-Nov, 2021 | 10-Jan, 2022 |

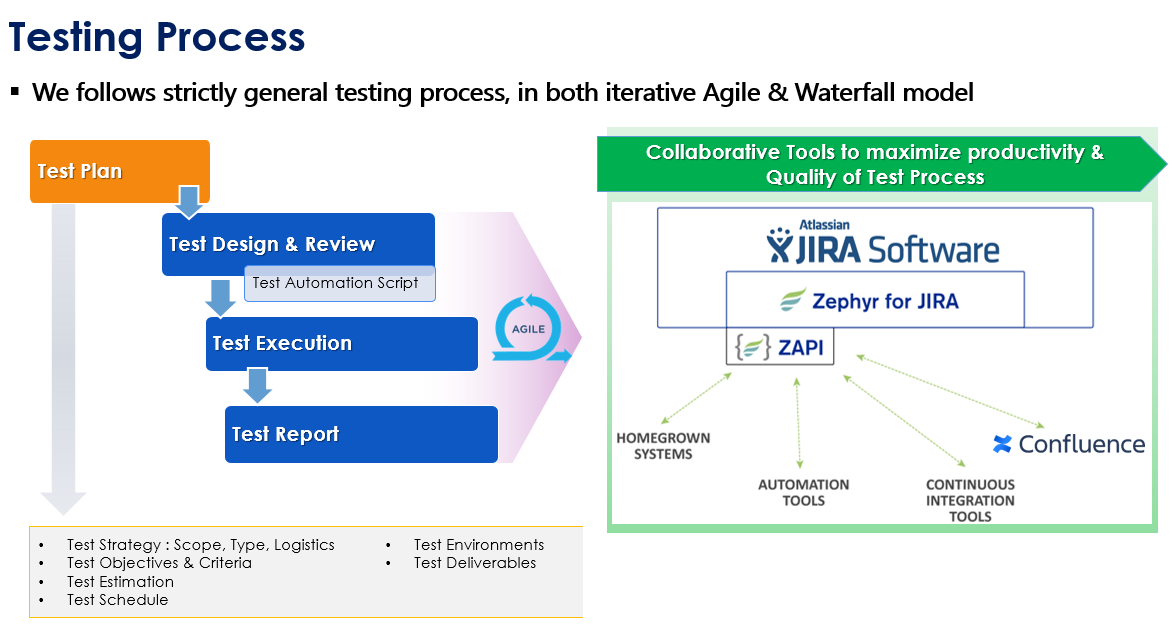
1. **SCHEDULE AND ESTIMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Task** | **Member** | **Estimation Effort**  **(Man-day)** |
| 1 | Create test plan | Team | 1 |
| 2 | Study & review requirements |  |  |
| 3 | Create test design, test cases |  |  |
| 4 | Test execution |  |  |
| 5 | Test report & Quality Control |  |  |

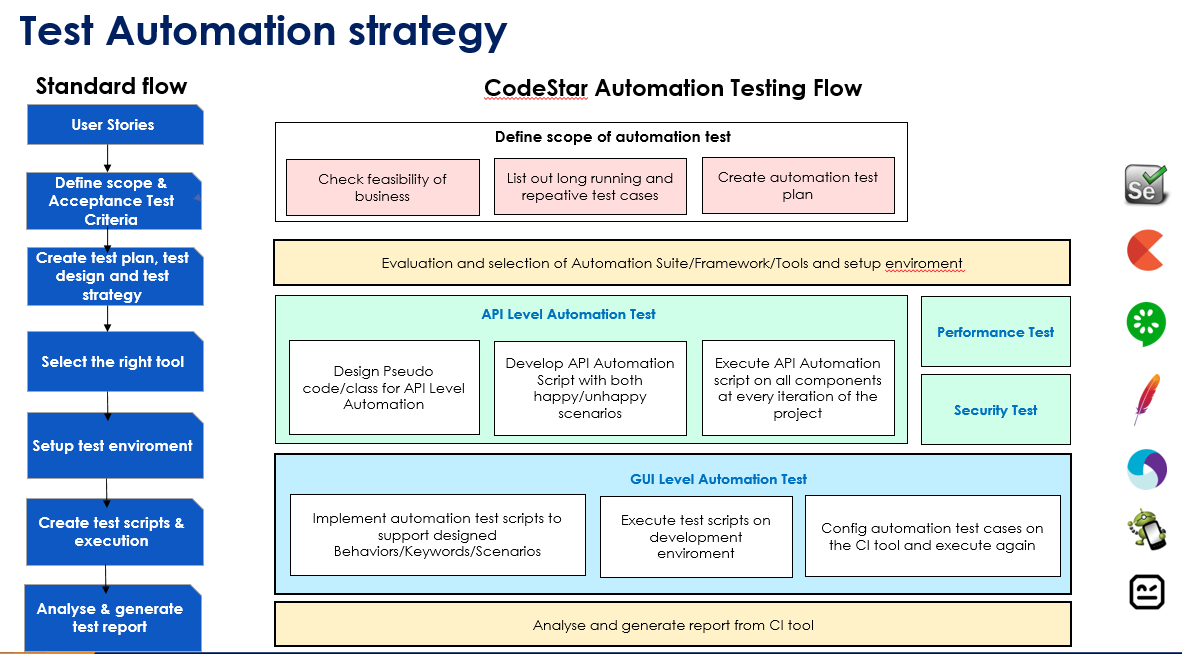
1. **TEST DELIVERABLE**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Deliverable Name** | **How to access** |
| **Before testing** | Test Plan | Test plan will be uploaded on Confluence  <https://conf.codestar.com/display/CDIS/Test+plan> |
| Test cases documents | All test cases will be uploaded on Confluence and Zephyr which is an Jira’s Add-On.  Link Confluence:  <https://conf.codestar.com/display/CDIS/Test+cases>  - Jira filter:  Open Advanced Search, then enter below JQL:  *project = CDIS AND issue type = Test* |
| **During testing** | Test cycles | Each Sprint will have at least 1 test cycle for executing sets of test cases.  Please follow this example to filter the test cycle: |
| **After testing** | Test reports | Test reports will be created after each sprint and published on Confluence: [https://conf. codestar.com/display/CDIS/Test+reports](https://conf.gemvietnam.com/display/CDIS/Test+reports) |
| Bugs | All bugs logged on Jira   * Sprint bugs are sub-tasks of the corresponding User story * Bugs/Improvement are linked to User stories, not add into current sprint |

1. **AUTOMATION TESTING TOOL AND STRATEGY**
   1. *Testing process*



* 1. *Automation strategy*



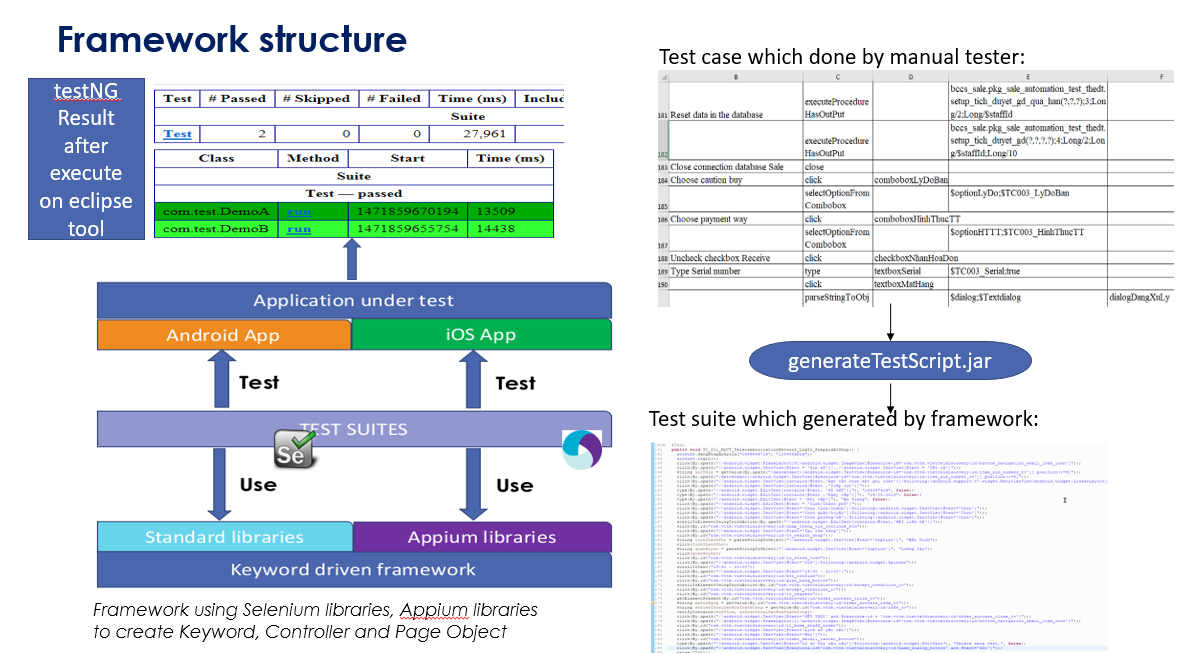
* 1. *Automation testing framework*

**Framework library/ tools:**

* Maven
* TestNG
* Github
* Selenium

**Language**:

* Java
* Python
* C#
* Javascript



1. **LIFECYCLE OF BUG AND RISK**

BUG LIFECYCLE

