Rise automation test project

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

**Automation Test Plan**

**Created by: Team2**

**Version: 1.0**

**Mar, 24th, 2024**

**Table of Contents**

[1. INTRODUCTION 3](#_Toc162183528)

[2. TEST OBJECTIVES 3](#_Toc162183529)

[3. TEST STRATEGY 3](#_Toc162183530)

[*3.1.* *Scope of Testing* 3](#_Toc162183531)

[*3.2.* *Test Levels and Test Types* 4](#_Toc162183532)

[*3.2.1.* *Test levels* 4](#_Toc162183533)

[*3.2.2.* *Test types* 4](#_Toc162183534)

[*3.3.* *Risks* 5](#_Toc162183535)

[4. TEST CRITERIA 6](#_Toc162183536)

[*4.1.* *Entry Criteria* 6](#_Toc162183537)

[*4.2.* *Suspension Criteria* 6](#_Toc162183538)

[*4.3.* *Exit Criteria* 7](#_Toc162183539)

[5. RESOURCE PLANNING 7](#_Toc162183540)

[*5.1.* *System resource* 7](#_Toc162183541)

[*5.2.* *Test Resource* 7](#_Toc162183542)

[6. SCHEDULE AND ESTIMATION 7](#_Toc162183543)

[7. TEST DELIVERABLE 8](#_Toc162183544)

[8. AUTOMATION TESTING TOOL AND STRATEGY 8](#_Toc162183545)

[*8.1.* *Testing process* 8](#_Toc162183546)

[*8.2.* *Automation strategy* 9](#_Toc162183547)

[*8.3.* *Automation testing framework* 9](#_Toc162183548)

[9. LIFECYCLE OF BUG AND RISK 10](#_Toc162183549)

1. **INTRODUCTION**

The purpose of the project is to build a system that, when new features are added, will not affect the current features of the system. The Test Plan determine the scopes, objectives and risk of testing. Defining the overall test approach, the overall framework, enviroment 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 and medium 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*

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 | Filter by Quick filters | 10 |
| 2 | Filter by Related to | 11 |
| 3 | Filter by Project | 1 |
| 4 | Filter by Milestone | 2 |
| 5 | Filter by Team member | 1 |
| 6 | Filter by Priority | 4 |
| 7 | Filter by label | 1 |
| 8 | Filter by deadline | 6 |
| 9 | Filter by status | 5 |
| 10 | Add new filter (have icon, don’t have icon) | 2 |
| 11 | Edit filter | 1 |
| 12 | Delete filter | 1 |
| 13 | Add task | 2 |
| 14 | Edit task | 2 |
| 15 | Delete task | 2 |
| 16 | Add multiple tasks | 1 |
| 17 | Search by keyword | 3 |
| 18 | Hide by ID | 2 |
| 19 | Hide by Title | 2 |
| 20 | Hide by Start date | 2 |
| 21 | Hide by Deadline | 2 |
| 22 | Hide by Milestone | 2 |
| 23 | Hide by Related to | 2 |
| 24 | Hide by Assigned to | 2 |
| 25 | Hide by Collaboration | 2 |
| 26 | Hide by Status | 2 |

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

* 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 |

* + 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) | Manual | Tester/Dev team | Verify that the system meets functional requirement |

* 1. *Risks*

|  |  |  |
| --- | --- | --- |
| **No.** | **Risk** | **Mitigation** |
| 1 | Requirement is in complete and mention a huge amount of distributed actor, leads tester to difficult to design test cases. | * Testers need to review user story/requirement before writing test cases (usually the first 2 days of each Sprint) * Improve communication in the team * Peer review testcase |
| 2 | * Technology Compatibility: The selected technology may not be compatible with certain features, leads to integration difficulties. * Performance Issues: The application may experience performance issues, such as slow loading times or unresponsive user interfaces, especially when filtering large datasets. * Data Breach: There's a risk of unauthorized access to sensitive user data stored within the application, especially if security measures are not adequately implemented. * Server Downtime: The application may experience server downtime due to technical issues or maintenance, leading to interruptions in service availability. | * Planning and resources for environmental restoration * Do load test, stress test to ensure system load |
| 3 | **Schedule Risks:**  - Project schedule is tight and short, it may be hard to complete testing on time  -Resource Constraints: Limited availability of resources, such as developers or testing environments, may result in delays or bottlenecks in the project workflow. | * Set priority for test activities: focus testing on the features that contains many critical bugs * Writing test checklist instead of test cases step by step if not enough time |
| 4 | System is complex so deployment may be has potential issue | In sprint planning, team need to clarify deployment schedule |
| 5 | **User Acceptance Risks:**   * User Adoption: Users may not fully adopt or engage with the application if the user interface is not intuitive or if the filter option does not meet their needs. * Feature Overload: The filter option may include too many criteria or be overly complex, leading to confusion and dissatisfaction among users. | * Prioritize user experience (UX) design by conducting user research, usability testing, and gathering feedback throughout the development process. * Incorporate user feedback and iterate on the design to address usability issues and improve the overall user experience. * Offer interactive tutorials, walkthroughs, or video guides to demonstrate how to use the filter option effectively and maximize its benefits. * Regularly analyze usage metrics, user feedback, and support tickets to identify pain points, usability issues, and areas for enhancement.   Iterate on the filter option based on user feedback and data-driven insights to ensure it meets users' evolving needs and expectations over 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 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: |
| 2 | Test tool |  |
| 3 | Computer | * The PCs with hardware, software required |
| 5 | **Web browser** | * Chrome, Microsoft Edge, Firefox, |

* 1. *Test Resource*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Start date** | **End date** |
| Dang Thi Quynh | Tester work 100% | 22-Mar, 2024 | 31-Mar, 2024 |
|  | Tester work 100% | 22-Mar, 2024 | 31-Mar, 2024 |
|  | Tester work 100% | 22-Mar, 2024 | 31-Mar, 2024 |

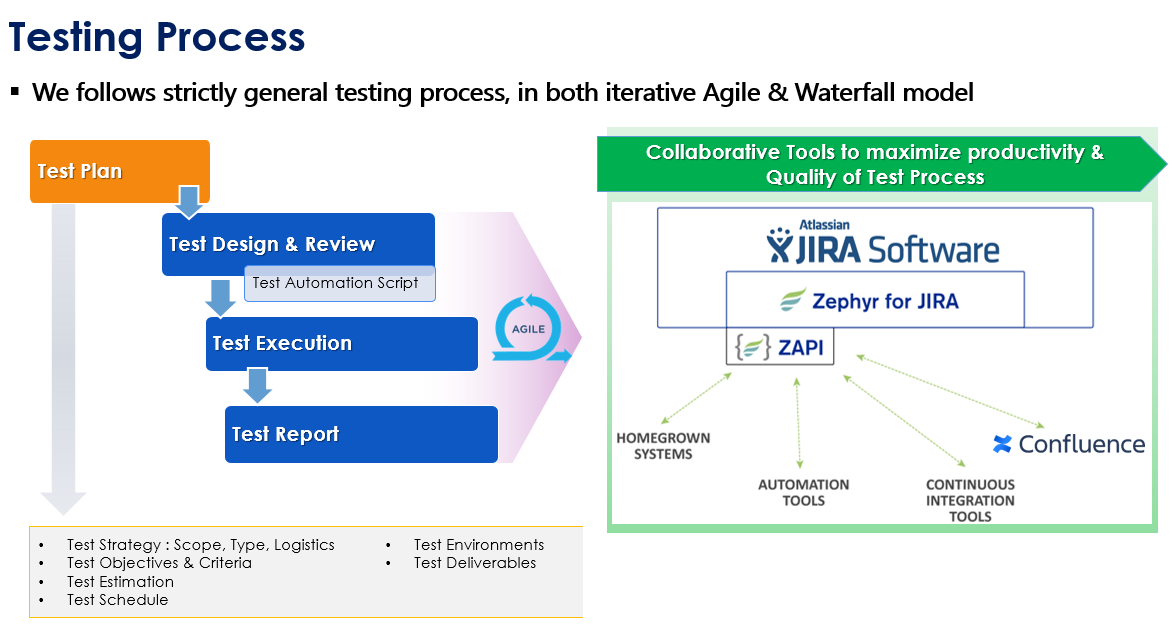
1. **SCHEDULE AND ESTIMATION**

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

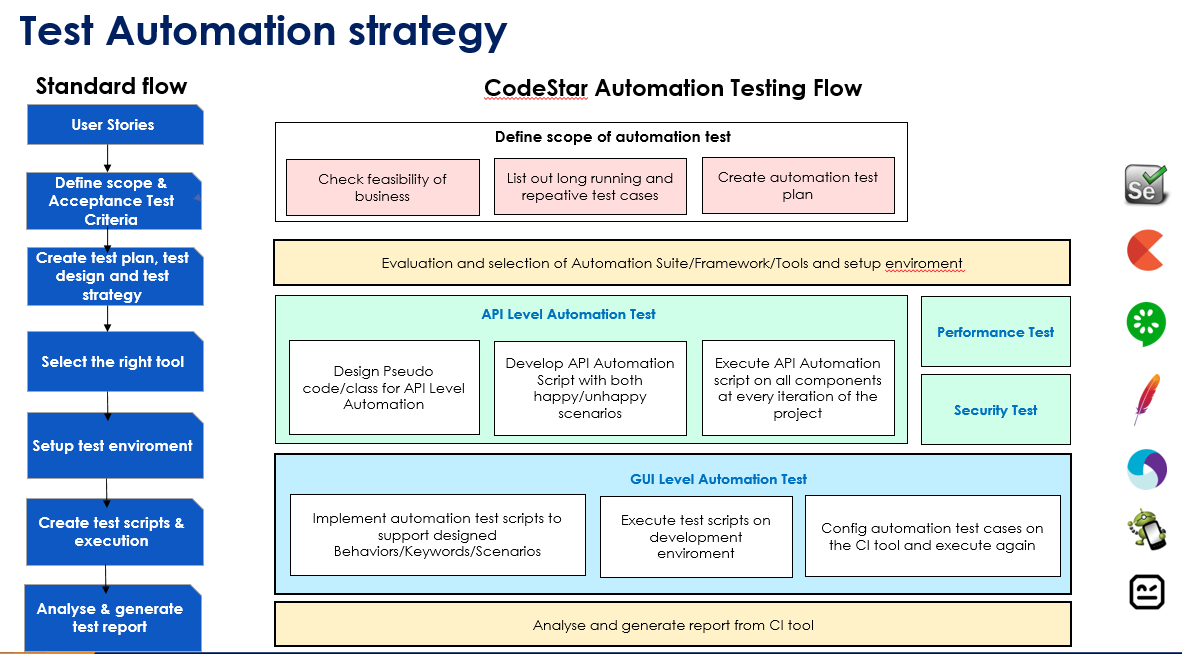
1. **TEST DELIVERABLE**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Deliverable Name** | **How to access** |
| **Before testing** | Test Plan | Test plan will be uploaded on github  Team2­\_TestPlan.docx |
| Test cases documents | All test cases will be uploaded on github. |
| **During testing** | Test cycles | Each Sprint will have at least 1 test cycle for executing sets of test cases. |
| **After testing** | Test reports | Test reports will be created after each sprint. |
| 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*

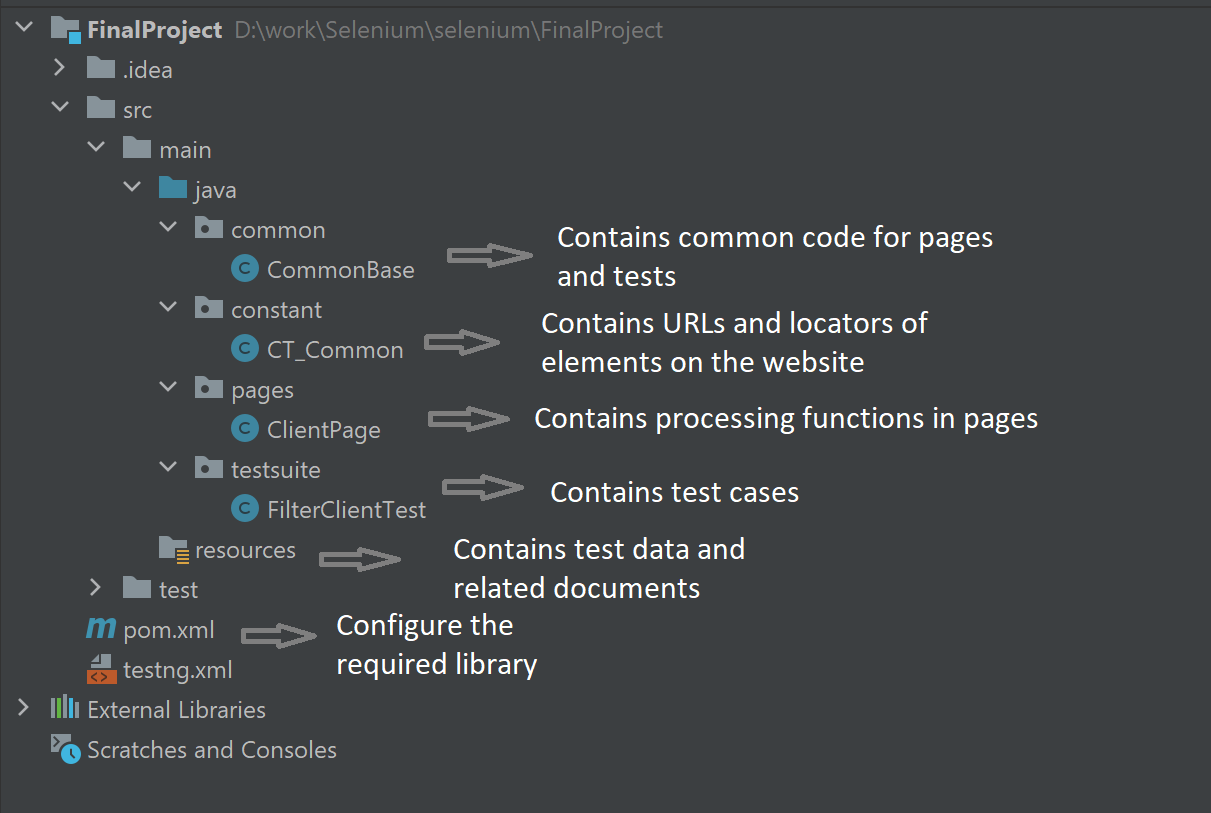
**Framework library/ tools:**

* Maven
* TestNG
* Github
* Selenium

**Language**:

* Java

**Test Structure**



1. **LIFECYCLE OF BUG AND RISK**

BUG LIFECYCLE

