**Propose and Scope:**

This document provides a detail and high level comprehensive, twitter data visualization of testing and technical knowledge and including brief documentation of testing. The testing is including selected items based according to the unique specification, here it is meant that the items which is most applicable in the given project.

This project is oriented based user experience and as a result most of the testing part is done by hand in testing to the user experience, and another test case is performed by some logic and analysed based. However, the unit testing to high level testing is performed to correct functionality of the user experience. As testing steps consume 30 to 50% of the project efforts. This document setup and monitoring and run the test program and summaries their result.

**Principles and Guiding:**

The main principle is that idea of each developer member is considered for the testing and quality of items, what they developed. For the testing purpose, we should take the depth knowledge of the use cases and all the components including in the project, exactly that ones we should use of the testing, and approach of the user, how they are using for the system and these approaches we need to consider for highly recommended for the testing.

**System Description:**

The “Map Based Social Network Visualisation” is twitter data visualization web application. This application allows user to analyze twitter data to perform zoom in and zoom out interactive map. Map is capable of simulation of numerous functionalities including:

* **Followers and Following**
* **Details of Followers and Following**
* **Relationship between followers and following**
* **Search twitter keywords**

**Objective:**

The main objective of testing for project is to ensure all the perquisite requirement is fulfilled or not and firstly check the customers requirement and match with the project final output and second objective to find the faults or issues occurring in the application before it is submitted to the user end, final step to evaluate performance and security of the project.

**Software quality factors**

**1.Product operation factor:** Correctness, reliability, efficiency, Integrity and usability

1. **Correctness:** Requirements deals with proper output of the application.

**Target:**

* The complete accuracy in the project without any negative and incorrect value.
* The completeness of the proper timeline .
* Specific standard of the codes and documentation

**Priority:** Must have

1. **Reliability:** Requirements deal with failure service.

**Target:**

* This requirement provides service to failure system.

**Priority:** Should have

1. **Efficiency:** Requirement deal with hardware components that perform the function of the application.

**Target:**

* Processing capabilities, storage and data capabilities.
* It deals with time between two units such as portable and meteorological units

**Priority:** Should have

1. **Integrity:** Requirement deal with security system of the main application.

**Target:**

* System security from unauthorized people access.
* Deals with cyber, internet, network security

**Priority:** Must have

1. **Usability:** Requirements deal with scope of the resource for the new employee.

**Target:**

* Operate for the new employees and trained them.

**Priority:** Should have

**2.Product revision factor:** Maintainability, Flexibility and testability

1. **Maintainability:** Requirement deals with proper maintenance

of the system

**Target:**

* Corrective maintenance
* Identify the cause of the failure and verify the success of the correctness.

**Priority:** Must have

1. **Flexibility:** Factor deals with adaptive activities of the software

**Target:**

* Change the resource and maintenance for the customer for more robust environment.
* Change the service with proper maintenance of the software.

**Priority:** Should have

1. **Testability:** Requirement deals with testing of the operation

**Target:**

* Deals with log files and obtain the result of the all the working components, to find it work properly.
* Automatic diagnostic check that is detected by the technicians.

**Priority:** Must have

3.**Product Transition factor:** Portability, reusability and Interoperability

I. **Portability :** Requirement deals with software system environment.

**Target:**

* Adaption of the software for the other environment and operating system and soon,
* Software continue with the different software diverse condition.

**Priority:** Must have

II. **Reusability:** Factor deals with software modules currently developed and make use for future application.

**Target:**

* Reuse the software such as quality modules and development period.

**Priority:** Should have

**III. Interoperability:** Factor interface with new components resource.

**Target:**

* These software’s can be interface platforms and database.
* Interface with new operating system.

**Priority:** Should have

**4.Test Scope:** The scope of the testing for the visualization of the twitter data. This project focus on the functionality part of the project and integration of the system and the user experience ability is the part of the testing, as a result testing procedure for the scope of the project. Some components which is not tested in this process due to some constraints or techanical ….. uses cases of the application is not involved in the system for large number of users, performance of the system under high load process for the scope of testing.

1. **In Scope:**

Functional testing are the main modules in the in-scope.

1.

2.

3.

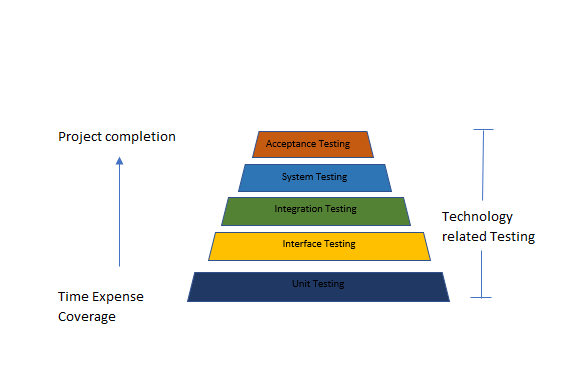
4.

1. **Out Scope:**
2. **Item not tested:**

**5.Test Approach:**

Software testing are two types functional and non- functional testing, functional testing using to the features and the functionality of the software. In our project, we are looking forward to functional testing and there are different types include:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Definition** | **Test Method** | **Execution** |
| **Unit** | Techniques using for every module to detect the issues. | Black and white testing | Apply to the Git individual commit code. |
| **Interface** | Techniques evaluation the components and system , the interface between hardware and software. | selenium | Apply to the changes made in the system/ impact on the user. |
| **Integration** | Techniques evaluation the functional, performance and reliability. | Manual testing | Apply to the modules of the codes that are integrated. |
| **System** | Techniques evaluate the complete system against the specific requirement. |  | Apply to the end to end modules for whole system. |
| **Acceptance** | Techniques using to the software requirement meet. |  | Apply to when meeting with customer. |



**Test Preparation:**

**Environments:**

The twitter data visualization test is used to very simplistic set of development and testing environment based on the Git Hubs.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Tools | Test |
| Development | Source code version control, development new codes, collaborative of codes and synchronised in all machines | GitHub | Unit and Informal Test |

The GitHub is cloud based platform, developers deploy their codes and fetch any times , First step to create a branch in the Project and developers can free to do changes in that branch without affect to master branch. ….

Test Execution:

Test Cases:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Application Testing | | | | | | |
| Tested by |  | | Application Name | | |  |
| Procedure | | Expected Result | | Pass/Fail | Results/Review | |
| Start the web application when other instances are running | | Yes | | Pass | No errors while running more applications | |
| Change screen resolution from (640x480, 800x600) | | Yes | | Pass | No effect on the result to see marker. | |
| Import all files twitter data(csv) | | NA | | Fail | Only support Geo-json files. | |
|  | |  | |  |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Application Testing | | | | | | |
| Tested by |  | | Application Name | | |  |
| Procedure | | Expected Result | | Pass/Fail | Results | |
|  | |  | |  |  | |

Test Case 1: Run Server

|  |  |
| --- | --- |
| Requirement |  |
| Description |  |
| System Environment |  |
| Hardware Environment |  |
| Procedure |  |
| Expected Result |  |
| Testing Result |  |
| Pass/Fail |  |
| Features |  |

Test Case 2:

|  |  |
| --- | --- |
| Requirement |  |
| Description |  |
| System Environment |  |
| Hardware Environment |  |
| Procedure |  |
| Expected Result |  |
| Testing Result |  |
| Pass/Fail |  |
| Features |  |

Test Case 3:

|  |  |
| --- | --- |
| Requirement |  |
| Description |  |
| System Environment |  |
| Hardware Environment |  |
| Procedure |  |
| Expected Result |  |
| Testing Result |  |
| Pass/Fail |  |
| Features |  |

Test Case 4:

|  |  |
| --- | --- |
| Requirement |  |
| Description |  |
| System Environment |  |
| Hardware Environment |  |
| Procedure |  |
| Expected Result |  |
| Testing Result |  |
| Pass/Fail |  |
| Features |  |