**Master Test Plan**

**TEST PLAN IDENTIFIER**

ID: MTP1.0

**REFERENCES**

IEEE test plan outline

**Introduction**

This is the master test plan for Decide project.

The main focus of the plan is to ensure the correctness of the functional and non-functional requirements stated in software requirement specification.

The project will have three levels of testing, Unit, System/Integration and Acceptance. The details for each level are addressed in the approach section.

The estimated timeline for this project is very aggressive (1 month), such that any delay in the development, installation and verification of the third party software will have a significant effect on the project .

**Test Items**

The following is a list by ID of the items to be tested :

1. Training ID: 1.
   1. Data Reader ID: 1.1.
   2. Data manipulation ID: 1.2.
   3. Parameter tuning and training ID: 1.3.
2. Release ID: 2.
   1. Data Reader ID: 2.1.
   2. Predict ID: 2.3.

**Items to to excluded**

Any item that is implemented using well-known libraries needs no unit testing.

**Critical Steps**

Before testing any item the required libraries must be installed and imported.

This can be identified from the software design document with the aid of the component ID.

Ensure that the testing environment provides at least the minimum hardware support required by the software in the software requirement specification document.

**Features To Be Tested**

|  |  |  |
| --- | --- | --- |
| **High Risk** | **Medium Risk** | **Low Risk** |

|  |  |
| --- | --- |
| 1. Import Dataset |  |
| 1. Import Model |  |
| 1. Display visualization for input data |  |
| 1. Display visualization for output |  |
| 1. Predict Results |  |

**Feature not to be tested:**

|  |  |  |
| --- | --- | --- |
| The input language | Already mentioned in SRS that Decide can classify english reviews  Non english reviews will be handled by python libraries |  |
| Input of output visualization component that is prior to predict component of  ID: 2.3. | Already tested in previous component. |  |

**Approach**

The testing for the **Decide** will consist of Unit, System/Integration (combined) and Acceptance test levels. However, with the timeline constraints established; most testing will be done by one member of the development team and will be reviewed by the team leader.

**UNIT Testing** will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person.

Test person can find unit information in SDD document.

**SYSTEM / INTEGRATION Testing** will be performed by a member of development team with the agreed test cases.

Testing member must consider required libraries and packages that must be installed based on the software components.

Programs will enter into System/Integration test after all critical

defects have been corrected.

**ACCEPTANCE** Testing will be performed by the actual end users with the assistance of team leader.

The acceptance test will be done in after completion of Integration testing

And for one week.

**Configuration Management/Change Control**

Movement of the program through development , testing and production phase will be controlled by github and project leader.

This will ensure that programs under development and those in full test will have the same version controls and tracking of changes.

Once the system has reached a reasonable level of stability where no critical defect, we will move the system from unstable to stable.

All changes, enhancements and other modification requests to the system will be handled after the agreement with team leader.

**Test Tools**

All editing, compiling will be done using the Jupyter Notebook.

Github and Git is used as the version control tool.

**Meetings**

The test will meet once or twice at the beginning of the project, then a periodic online meeting will be conducted to track the updates within the team itself and with the team leader.

Additional offline meetings can be called as required for emergency situations.

The online meeting will be conducted on JITSI MEET.

**ITEM PASS/FAIL CRITERIA**

The completion criteria is based on the completion of all lower level plans  
Unit test: include the success in test cases for each unit.

System Test: include the successful integration between software components and the satisfaction of functional requirement mentioned in SRS.

Acceptance testing: Include receiving the feedback report after one week of testing as agreed.

A defect will be left or maintained based on its severity to the software based on the acceptance test report.

**Suspension Criteria and Resumption Requirements**

1. The default model is not already developed

The prediction testing Is not useful if the default model is not already developed and tested.

**TEST DELIVERABLES**

Unit Test cases

System Test plan

Acceptance Test plan

**REMAINING TEST TASKS**

|  |  |
| --- | --- |
| **Task** | **Responsibility** |
| Define Unit test cases | Team member, Team Leader |
| Define Integration test | Team member, Team Leader |
| Acceptancetest | Client |

**ENVIRONMENTAL NEEDS**

The following elements are required to support the overall testing :

1. Windows OS (7 or higher)

**STAFFING AND TRAINING NEEDS**

1. It’s required to have a tester with a background in testing learning algorithms to test the default loaded model.
2. Developers must be trained on using pyUnit for testing each unit.

**RESPONSIBILITIES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **TL** | **DEV team** | **Client** |
| Acceptance test documentation | **X** |  | **X** |
| Integration test documentation | **X** | **X** |  |
| Unit testing |  | **X** |  |
| Change control | **X** |  |  |

**SCHEDULE**

Time has been allocated within the project plan for the following testing activities.

The specific dates and times for each activity are defined in the project plan timeline.

The persons required for each process are detailed in the project timeline and plan as well.

Coordination of the personnel required for each task, test team, development team, management and customer will be handled by the team leader in conjunction with the development team.

**Risks and Contingencies**

|  |  |  |
| --- | --- | --- |
| Risk | What to do |  |
| Developer doesn’t have windows operating system and can’t run a virtual machine | The developer must be assigned to tasks where it’s independent to the operating system |  |
| Developer doesn’t have the required  hardware for testing especially in the training component. | Consider online development for the component.  Start pair programming technique |  |
| Delay of training component. | Use pre-trained and similar low accuracy model |  |