Test plan document for Online voting system

# INTRODUCTION

## Purpose

This test plan describes the testing approach and overall framework that will drive the testing the Online Voting system The document introduces:

* Test Strategy: Describes the rules the test will be based on.
* Execution Strategy: Describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
* Test Management: Process to handle the logistics of the test and all the events that come up during execution .

## Project Overview

The Online Voting System is a very efficient tool which helps in automating the election process .

It also helps in providing results faster and more accurately without the need of manual counting of ballot paper.

It also provides all the stakeholders of the system to keep track of the voting process in real time.

## Audience

* Project team members perform tasks specified in this document, and provide input and recommendations on this document.
* The admins will provide their views on functional changes.
* The other stakeholders(ie.candidates and voters) will take part in the UAT testing.

# TEST STRATEGY

## Test Objectives

The objective of the test is to verify that the functionality Online Voting System MODULE works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.

The final product of the test is twofold:

* A production-ready software;
* A set of stable test scripts that can be reused for Functional and UAT test execution.

## Test Assumptions

**Key Assumptions**

* Production like data required and be available in the system prior to start of Functional Testing

.

**General Assumptions**

* Performance testing is not considered for this estimation.
* All the defects would come along with a snapshot JPEG format
* The Test Team assumes all necessary inputs required during Test design and execution will be available.
* Test environment and preparation activities will be owned by Developer team.
* The project will provide test planning, test design and test execution support
* Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.
* There is no environment downtime during test due to outages or defect fixes.
* The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.

**Functional Testing**

* During Functional testing, testing team will use preloaded data which is available on the system at the time of execution
* The Test Team will be perform Functional testing only on Online Voting System MODULE.

**UAT**

* UAT test execution will be performed by end users and QA Group will provide their support on creating UAT script.

## Test Principles

* Testing will be focused on meeting the sytem objectives, cost efficiency, and quality.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
* Testing environment and data will emulate a production environment as much as possible.
* Testing will be a repeatable, quantifiable, and measurable activity.
* Testing will be divided into distinct phases, each with clearly defined objectives and goals.
* There will be entrance and exit criteria.

## Data Approach

* In functional testing, Online Voting System will contain pre-loaded test data and which is used for testing activities.

## Scope and Levels of Testing

### Exploratory

**PURPOSE**: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

**SCOPE**: First level navigation, candidate and admin modules

**TESTERS**: Testing team.

**METHOD**: this exploratory testing is carried out in the application without any test scripts and documentation

**TIMING**: at the beginning of each cycle.

### Functional Test

**PURPOSE:**  Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

**TESTERS**: Testing Team.

**METHOD**: The test will be performed according to Functional scripts, which are stored in CPP TEST.

**TIMING**: after Exploratory test is completed.

#### TEST ACCEPTANCE CRITERIA

1. Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
2. Test cases approved and signed-off prior to start of Test execution
3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
4. Test environment with application installed, configured and ready to use state

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | Test Plan | Test Lead | Teaching Assistant |
| 2. | Functional Test Cases | Test Team | Teaching Assistant Sign off |
| 3. | Logging Defects in CPP TEST | Test Team | Teaching Assistant |
| (4. | Daily/weekly status report | Test Team/ Test Lead | Teaching Assistant |
| 5. | Test Closure report | Test Lead | Lab Incharge |

### User Acceptance Test (UAT)

**PURPOSE**: this test focuses on validating the softare logic. It allows the end users to complete one final review of the system prior to deployment.

**TESTERS**: the UAT is performed by the end users .

**METHOD**: Test team write the UAT test cases based on the inputs from End user (candidates and voters) and admins.

**TIMING**: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

#### TEST DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | UAT Test Cases | Test Team | Teaching Assistants Sign off |

# EXECUTION STRATEGY

## Entry and Exit Criteria

* The entry criteria refer to the desirable conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.
* The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
* Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final “go-no go” decision.
* Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
* Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| 100% Test Scripts executed |  |  |  |
| 95% pass rate of Test Scripts |  |  |  |
| No open Critical and High severity defects |  |  |  |
| 95% of Medium severity defects have been closed |  |  |  |
| All remaining defects are either cancelled or documented as Change Requests for a future release |  |  |  |
| All expected and actual results are captured and documented with the test script |  |  |  |
| Test Closure Memo completed and signed off |  |  |  |
| Test environment cleanup completed and a new back up of the environment |  |  |  |

## Test Cycles

* + There will be two cycles for functional testing. Each cycle will execute all the scripts .
  + The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.
  + The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.
* UAT test will consist of one cycle.

## Validation and Defect Management

* It is expected that the testers execute all the scripts in each of the cycles described above. However it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.
* The defects will be tracked through CPPtest only. The technical team will gather information on a regular basis from CPPtest,.The technical team will work on fixes.
* It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review CPPtest on a regular basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tool “Mercury CPP TEST” and the categories are:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1 (Critical) | * This bug is critical enough to crash the system, cause file corruption, or cause potential data loss * It causes an abnormal return to the operating system (crash or a system failure message appears). * It causes the application to hang and requires re-booting the system. |
| 2 (High) | * It causes a lack of vital program functionality with workaround. |
| 3 (Medium) | * This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen. * This bug prevents other areas of the product from being tested. However other areas can be independently tested. |
| 4 (Low) | * There is an insufficient or unclear error message, which has minimum impact on product use. |
| 5(Cosmetic) | * There is an insufficient or unclear error message that has no impact on product use. |

## Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test preparation & Execution Status | To report on % complete, %WIP, % Pass, % Fail  Defects severity wise Status – Open, closed, any other Status | Weekly (optional) |
| Daily execution  status | To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects | Daily |
| Project Weekly Status report | Project driven reporting (As requested by PM) | Weekly – If project team needs weekly update apart from daily and there is template available with project team to use. |

## 

## Defect tracking & Reporting

Following flowchart depicts Defect Tracking Process:

# 5.TEST MANAGEMENT PROCESS

## Test Management Tool

CPP testis the tool used for Test Management. All testing artifacts such as Test cases, test results are updated in the CPPtest tool.

Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in CPP TEST.

* During the Test Design phase, all test cases are written directly into CPP test. Any change to the test case will be directly updated in the CPP TEST.
* Each Tester will directly access their respective assigned test cases and update the status of each executed step in CPP TEST directly.
* Any defect encountered will be raised in CPP TEST linking to the particular Test case/test step.
* During Defect fix testing, defects are re-assigned back to the tester to verify the defect fix. The tester verifies the defect fix and updates the status directly in CPP TEST.
* Various reports can be generated from CPP test to provide status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects etc.

## 5.2 Test Design Process

* The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
* Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
* Each of the Test cases will undergo review by the TA’s and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
* During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
* Testers will maintain a clarification Tracker sheet and same will be shared periodically with the Requirements team and accordingly the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
* Any subsequent changes to the test case if any will be directly updated in CPP TEST.

## 5.3 Test Execution Process

* Once all Test cases are approved and the test environment is ready for testing, tester will start a exploratory test of the application to ensure the application is stable for testing.
* Each Tester is assigned Test cases directly in CPP TEST.
* Testers to ensure necessary access to the testing environment, CPP TEST for updating test status and raise defects. If any issues, will be escalated to the Test Lead and in turn to the Project Manager as escalation.
* If any showstopper during exploratory testing will be escalated to the respective development SPOCs for fixes.
* Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the step directly in CPP TEST.
* Tester will prepare a Run chart with day-wise execution details
* If any failures, defect will be raised as per severity guidelines in CPP TEST tool detailing steps to simulate along with screenshots if appropriate.
* Daily Test execution status as well as Defect status will be reported to all stakeholders.
* Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
* If there are any defects that are not part of steps but could be outside the test steps, such defects need to be captured in CPP TEST and map it against the test case level or at the specific step that issue was encountered after confirming with Test Lead.
* This process is repeated until all test cases are executed fully with Pass/Fail status.
* During the subsequent cycle, any defects fixed applied will be tested and results will be updated in CPP TEST during the cycle.

## Test Risks and Mitigation Factors

| Risk | Prob. | Impact | Mitigation Plan |
| --- | --- | --- | --- |
| **SCHEDULE**  Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date. | High | High | * The testing team can control the preparation tasks (in advance) and the early communication with involved parties. * Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| **RESOURCES**  Not enough resources, resources on boarding too late (process takes around 15 days. | Medium | High | Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing. |
| **DEFECTS**  Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve. | Medium | High | Defect management plan is in place to ensure prompt communication and fixing of issues. |
| **SCOPE**  Scope completely defined | Medium | Medium | Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing. |
| Natural disasters | Low | Medium | Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities. |
| Non-availability of Independent Test environment and accessibility | Medium | High | Due to non availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution. |
| Delayed Testing Due To new Issues | Medium | High | During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve.  There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.  If these issues become showstoppers, it will greatly impact on the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |

# TEST ENVIRONMENT

A windows environment with Internet Explorer 8, 9 and 10, and with Firefox 27.0, as well as Google Chrome 32.0 and later should be available to each tester

# APPROVALS

The Names and Titles of all persons who must approve this plan.

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** |  |
| **Role:** |  |
| **Date:** |  |

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** |  |
| **Role:** |  |
| **Date:** |  |