****

Project Name: Developing a Test Plan for Automated Ticket Issuing System for Dhaka Subway Systems (DSS)

Subject: Software Quality and Testing

Section: D

**GUIDED BY**

S.M. Abdur Bhuiyan Rouf

Department of Computer Science

American International University-Bangladesh

**PREPARED BY**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **NAME** | **ID** | |
| 1. | RAHMAN, SHOJIBUR | | 18-36205-1 | |
| 2. | ISMAIL, ISRITA | | 17-35675-3 | |
| 3. | RAFA, HUMAYARA CHOWDHURY | | 17-35413-3 | |
| 4. | RANA, MD.SOHEL | | 16-32603-2 | |
| 5. | AHMED, MAIMUN | | 16-31950-2 | |

**Contents**

[1. Test Plan Identifier 2](#_Toc26221844)

[2. References 2](#_Toc26221845)

[3. Introduction: 2](#_Toc26221846)

[4. Test Items 3](#_Toc26221847)

[5. Software risk issues 3](#_Toc26221848)

[6. Features to be tested 4](#_Toc26221849)

[7. Features not to be tested 4](#_Toc26221850)

[8. Approach 5](#_Toc26221851)

[9. Item Pass/Fail criteria 5](#_Toc26221852)

[9.1 Component 5](#_Toc26221853)

[9.2 Integration 6](#_Toc26221854)

[9.3 System 6](#_Toc26221855)

[10. Suspension Criteria and Resumption Requirements 6](#_Toc26221856)

[10.1 Automated Unit Test Suite 6](#_Toc26221857)

[10.2 Build Acceptance Test (BAT) 7](#_Toc26221858)

[10.3 Regression Testing 7](#_Toc26221859)

[10.4 System Design Changes: 7](#_Toc26221860)

[11. Test Deliverables 8](#_Toc26221861)

[12. Remaining test tasks 8](#_Toc26221862)

[13. Environmental needs 9](#_Toc26221863)

[14. Staffing and Training needs 9](#_Toc26221864)

[14.1 Staff Needs 9](#_Toc26221865)

[14.2 Training Needs 9](#_Toc26221866)

[15. Responsibilities 10](#_Toc26221867)

[16. Planning Risks and Contingencies 10](#_Toc26221868)

[17. Schedule 11](#_Toc26221869)

[18. Approvals 11](#_Toc26221870)

[19. Glossary 12](#_Toc26221871)

**1. Test Plan Identifier**

DS\_ Automated Ticket Issuing System\_1.1

**2. References**

* TEST PLAN OUTLINE (IEEE 829 FORMAT) http://ece.uprm.edu/~icom5047/documents/OtherDocuments/TestPla n.pdf The Functional
* Development and Test standards Process: IEEE 829-1998 Standard for Software Test Documentation.
* Pressman, Roger S. Software Engineering - A Practitioner's Approach. Fifth edition. The McGraw-Hill companies, Inc.
* Kaner, C., Falk, J., Nguyen, H.-Q. Testing Computer Software. Wiley Computer Publishing, 1999.

**3. Introduction:**

This document is an overview defining our testing strategy for the automated ticket selling application of Dhaka Subway Systems. This project's goal is to provide automated ticket selling for public uses. This document will address the different standards that will apply to the unit, integration and system testing of the specified application. We will utilize testing criteria under the white box, black box, and system-testing paradigm. This paradigm will include but is not limited to, the testing criteria, methods, and test cases of the overall design. Throughout the testing process, we will be applying the test documentation specifications described in the IEEE Standard 829 for Software Test Documentation.

The software has the following set of requirements:

* The software will support interface to touch screen monitors as well as keyboard interface.
* The software will support display of the list of incoming trains, their destinations and arrival and departure times, fare.
* The software will support multiple ticket purchase simultaneously.
* The software will support limiting the number of tickets purchased at the same time. This privilege control will be done by the administrator access only.
* The software will support ticket cancellation before the final confirmation of the purchase.
* The software will support purchased ticket cancellation support by the administrator.
* The software will support credit transaction and validation.
* The software will support the next and previous navigation during the ticket purchase process.
* The software will support ticket availability information.
* The software will support information display via the web.
* The software will use the Oracle database server.

**4. Test Items**

The major functionalities of the system are as follows:

* 24/7 service
* Ticket availability information display
* Train arrival and departure time display
* Touch screen menu selection
* Source and destination selection
* Multiple ticket issue in one transaction
* Limit the number of ticket issue at the same time
* Cancellation of transactions any time during the transaction
* Credit/Debit card transaction
* Coin/Taka recognition and acceptance

**5. Software risk issues**

Ticket availability information display function to be tested. Touch screen menu selection should be test because if it is not working properly then the whole process will be ruined. The Transaction Function should be tested it is more than important. Check off all Third-party Delivery product. Checking the Most Complex Function. A new version of the product should be highly risked to match with the older one. Poorly documented modules or change requests.

There are some inherent software risks such as complexity; these need to be identified.

* The whole Transaction Safety issue
* Real-time Access of Client is a big risk issue
* Government regulations and rules. Another key area of risk is a misunderstanding of the original requirements.

These types of risk are frequently meet with software process.

**6. Features to be tested**

* Touch Screen monitors and keyboard interface.
* Touch screen menu selection.
* Displayed trains arrival and departure time information, fare.
* Multiple ticket purchase support simultaneously.
* Limiting the number of tickets purchased at the same time by privilege control and the administrator access only.
* Ticket cancellation support before the final confirmation of the purchase.
* Purchased ticket cancellation support by the administrator
* Ticket availability information.
* Credit/Debit card transaction.
* Coin/Taka recognition and acceptance.
* Displayed information via Website.
* Oracle database server support.

**7. Features not to be tested**

The features are not to be tested given bellow:

* 24 Hours per week service.
* Ticket availability information display.
* Train arrival and departure time display.
* Touch screen menu selection.

Source and destination Selection.

* Credit/Debit card acceptance.
* Coin/Taka recognition and acceptance.

**8. Approach**

The following represents the overall flow of the testing process approach:

Identify the requirements to be tested. All test cases shall be derived using the current Program Specification. Identify which particular test(s) will be used to test each module Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit. Identify the expected results for each test. Document the test case configuration, test data, and expected results. Perform the tests. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR). Successful unit testing is required before the unit is eligible for component integration/system testing. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered its possible cause and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

**9. Item Pass/Fail criteria**

This section specifies generic pass/fail criteria for the tests covered in this plan. They have supplemented bypass/fail criteria in the test design specification.

* 1. **Component**

Tests executed on components only pass when they satisfy the signatures, constraints, and interfaces dictated by the Object Design Specification for that component. This includes positive tests, negative and stress tests, and boundary tests. If a test exhibits a product fails to meet the objectives of the object design specification, it will fail and a defect/issue will be reported in the defect tracking system for review by the triage team.

* 1. **Integration**

Tests executed on integrated components only pass when they satisfy the signatures, constraints, and interfaces dictated by both the object design specification and the system architecture specification. This includes positive tests, negative and stress tests, boundary conditions, and tests that explicitly manipulate the interface environment (such as the physical connection to the database server). If a test exhibits a product fails to meet the objectives of both the object design specification and the system architecture specification, it will fail and a defect/issue will be reported in the defect tracking system for review by the triage team

* 1. **System**

Tests executed against the system use the functional requirements, non-functional requirements, and use cases as the oracle to determine pass or fail. If a test exhibits a product fails to meet the objectives of any of the functional requirements, non-functional requirements, or the use cases, it will fail and a defect/issue will be reported in the defect tracking system for review by the triage team.

**10. Suspension Criteria and Resumption Requirements**

This section specifies the criteria for suspending the testing on the test items associated with the plan. It also specifies the test activities that must be repeated when testing is resumed.

* 1. **Automated Unit Test Suite**

As components are being developed, unit tests will be developed to test the interfaces of the components and low-level unit tests will be developed to test the methods of the underlying classes in the components. As a prerequisite to the BAT, the automated unit test suite will be run by the build server on a per-build basis. When the unit-test suite reports failures, testing will not occur on that build until the failures have been analyzed and resolved. Testing will resume on a build that passes the automated unit test suite.

* 1. **Build Acceptance Test (BAT)**

A build is deemed ready to test by development, a build acceptance test will be run on the build. The BAT will consist of a broad but shallow set of tests to determine the overall stability of the build and decide if it is worth testing. If the BAT fails on a particular build, testing will suspend until another build is created with any BAT failure issues fixed, verified by running the BAT again. Testing will resume on a build that passes the BAT. Different build acceptance tests will be developed and used for the different test phases. Component BATs will be small and localized for each of the components. Integration BATs will vary based on the level of integration testing is performed. The System Test BAT will contain a set of tests that will utilize each of the components of the system.

* 1. **Regression Testing**

On a build by build basis, major bug fixes or code changes will be reviewed to determine the effects they may have on the system. If the changes are deemed to cause a sufficient amount of risk, regression test sets of the appropriately judged size will be created and executed. A system-wide regression will also be run on the release candidate build to ensure incremental changes to the system have not altered the results of the tests that were run early in the test cycle.

* 1. **System Design Changes:**

If at any point in time issues are submitted that require a design change to the system, all testing will be suspended. After the changes to the requirements, system architecture, and object design are made, a review and updates will be performed of the test specifications to ensure they properly align with the revised system changes. After updates are made, testing will resume. Tests in the vicinity of the change must all be rerun. A 10% regression of other tests must also be performed to ensure the changes did not adversely affect other parts of the system.

**11. Test Deliverables**

Test Deliverables are the test artifacts which are given to the stakeholders of a software project during the SDLC (Software Development Life Cycle). Every software application goes through different phases of SDLC and STLC.

* Test Strategy
* Test Plan Document
* Test Cases
* Test Data
* Test Defect Report
* Test Summary Report
* Test Status Report
* Error Corrective Actions
* Test Evaluation Report
* Execution Log Files
* Summary

**12. Remaining test tasks**

* Create  an Acceptance Test Plan
* Create System/Integration Test Plan
* Define Unit Test rules and Procedures
* Define Turnover procedures for each level
* Verify prototypes of Screens
* Verify prototypes of Reports
* Website and GPS device of software interaction.
* GUI response and directly database testing.

**13. Environmental needs**

* + Hardware and software are needed to execute our system properly.
  + The database is needed to execute the system.
  + Access to the recovery process.
  + Need maximum capacity to use the system at the same time.
  + Need commonly used a browser to use the system.

**14. Staffing and Training needs**

* 1. **Staff Needs**

During the 1st quarter of the project, the project/test manager can play the role of a full-time test engineer with a part-time test engineer to assist with the review and initial planning of the project. After the initial build has been done, the validation and verification testing need 2 full-time testers and one of them needs to be the person connected with the project from the inception as assisting tester.

* 1. **Training Needs**

The Test Manager and Project Manager need to collaborate with administration personnel and train on the inner working of a course allocation flow and learn about available courses and prerequisite course needs. [Later more staff and training facilities will be added based on the earlier portion of this test plan document]

**15. Responsibilities**

**Test lead:**

* Defining the testing activities for subordinates – testers or test engineers.
* All responsibilities of test planning.
* To check if the team has all the necessary resources to execute the testing activities.
* To check if testing is going hand in hand with the software development in all phases.
* Prepare the status report of testing activities.
* Required Interactions with customers.
* Updating project manager regularly about the progress of testing activities.

**Test engineers/QA testers/QC testers:**

* To read all the documents and understand what needs to be tested.
* Based on the information procured in the above step decide how it is to be tested.
* Inform the test lead about what all resources will be required for software testing.
* Develop test cases and prioritize testing activities.
* Execute all the test case and report defects, define severity and priority for each defect.
* Carry out regression testing whenever changes are made to the code to fix defects.

**16. Planning Risks and Contingencies**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Probability | Risk Type | Owner | Contingencies Approach |
| Unable to acquire the necessary number of skilled personnel as the components become ready to test. | 35% | Personnel Schedule | Test Manager | Resources for components will be split between the existing resources. Schedule must be adjusted accordingly. |
| Unable to acquire some of the necessary hardware and software required for integration and system testing | 25% | Equipment | Program Test Development manager | Utilize existing acquired hardware. Split test execution into morning and evening shifts such that testing can occur for multiple teams in the same day using the limited hardware. |
| Components are not delivered on time | 20% | Schedule | Development Manager | Integration testing with those components must be delayed until the  Component is delivered.  Overall integration test approach may be modified to do an appropriate amount of bottom-up as well as top-down or sandwich integration. Schedule must be adjusted accordingly |
| Turnover | 20% | Personnel | Test Manager | Testers will work in pairs on components. If a single member of the team decides to leave, a secondary testing with the knowledge of the component will still be able to train a new tester or finish the work. Schedule must be adjusted accordingly. |

**17. Schedule**

|  |  |  |
| --- | --- | --- |
| Time | Test Preferences | Responsibilities |
| 25/09/2019 | Unit test of project planning | Test Team |
| 1/10/2019 | System design reviews | Test Manager |
| 11/10/2019 | Detail Design reviews | Test Manager |
| 18/10/2019 | Change control and regression testing | Test team and developers |
| 28/10/2019 | System testing documentation and execution | Test Team |
| 14/11/2019 | Change control and regression testing | Test Manager |
| 24/11/2019 | System testing documentation and execution | Test team and test manager |
| 8/11/2019 | Test case review | Test Manager |

**18. Approvals**

* Test manager then project manager prove Documentation and Execution.
* System Structure is approved by the Development Team Leader and Project Manager.
* To do the Unit testing, Test Manager Approval is needed.
* To do the Acceptance testing, Test Manager Approval is needed.
* Validation of data is approved by the Administration Personnel
* Project Manager then Test Manager then administration personnel approved the Final Project completion.

**19. Glossary**

|  |  |
| --- | --- |
| ATS | Automated Ticket System |
| MTP | Master Test Plan |
| PM | Project Manager |
| TEST SUITE | A collection of test cases that have the same test objective. |
| STR | System Test Report |
| BAT | BAT Build Acceptance Test |
| QA | QA Quality Assurance |
| SDLC | software development life cycle |