Table of Contents

Test Plan Identifier

The test plan identifier of the project is **TP_DSS_01.** Where TP means Test Plan and DSS means Dhaka Subway System. And the numeric numbers will track a unique identity of every single test plan.

Introduction

This test plan describes the testing approach and the overall workflow to test the project **Dhaka Automated Ticket Issuing System for Dhaka Subway Systems (DSS).** The test plan contains the methods and approach we have to follow for a successful testing of the project. The following items we have described and planed for the project.

- Software risk issues
- Features to be tested
- Features not to be tested
- Test items
- Items Pass/Fail criteria
- Remaining test tasks
- Approach
- Test Deliverables
- Environmental needs
- Staffing and Training needs
- Scheduling
- Planning Risks and Contingencies etc.

The test plan directs our thought by writing complete planning like this. Writing a test plan helps us face the obstacles that await us and concentrate our thinking on important subjects. It will also help the testers, developers, and project team members make the project clearer. Software risk issues

Software risk issues

With a focus on the testing process, these are the risk issues:

- Lack of staff resources is to start while checking.
- Lack of available hardware, software, data, or resources needed.
- Late distribution of applications, hardware, or tools.
- Delays in technology or software training.

- Modifications to the original specifications or versions.
- Complexities found in application testing

A sufficient number of days would be shifted from the test schedule and production schedule, which is rarely happens to It happens, as most projects tend to have dates of the delivery set.

- The number of experiments that are carried out will be decreased.
- The number of defects permissible will be increased.
- The research team will be added with money.
- The research team is going to work overtime (this could affect team morale).
- The plan's scope can be modified.
- There could be some resource optimization.

Management, even though they have seen situations such as the one above, is generally hesitant to recognize.

The main thing to note is that the expected outcome is that research is cut off if you do nothing at all. Completely back or omitted, none of which should be an appropriate choice.

Features to be tested

The following features will be tested for the project Automated Ticket Issuing System for Dhaka Subway Systems (DSS).

- The user interface or GUI testing
- Navigation testing for users
- Proper train scheduling testing
- Testing of single and multiple ticket purchasing options.
- Limitation for the users on purchasing tickets.
- User confirmation testing before buying tickets.
- Ticket cancellation system for the administrator testing.
- Payment methods testing.
- Current availability of tickets testing.
- 24/7 uptime testing.
- Coin and taka recognition system testing.

Test Items

Test case 01:

Project name: Automated Ticket Issuing System for DSS.				
Test case ID: **TC_DSS_01**				
Test priority: High				
Module name: Ticket showing system				
Test title: Verification of proper ticket schedule showing				
Description: When users enter into the system, they can view the list of upcoming trains, their destinations, arrival times and departure times and the fare.				
Pre-Condition: N/A				
Steps	*Test Data*	*Expected Result*	*Actual Result*	*Status*
1.Enter into the system 2.Look for a train	Mohakhali to Uttara train details	User should see the train information with fare and times.	As expected,	Passed
Post condition: N/A				

Test case 02:

Project name: Automated Ticket Issuing System for DSS.				
Test case ID: **TC_DSS_02**				
Test priority: High				
Module name: Find train.				
Test title: Searching for train with user's entered data.				
Description: User can enter **form** and **to,** and then they can find the upcoming trains for that.				
Pre-Condition: N/A				
Steps	*Test Data*	*Expected Result*	*Actual Result*	*Status*
1.Enter into the system 2.Enter from. 3. Enter to. 4.Click on Find.	Find for a train Mohakhali to Uttara	User should see some trains based on the search.	As expected,	Passed
Post condition: N/A				

Test case 03:

Project name: Automated Ticket Issuing System for DSS.

Test case ID: **TC_DSS_03**				
Test priority: High				
Module name: Book Ticket				
Test title: Book ticket.				
Description: User can select a train, and book for seats for the train.				
Pre-Condition: N/A				
Steps	*Test Data*	*Expected Result*	*Actual Result*	*Status*
1.Select a train 2.Select number of tickets 3. Click on confirm 4.Pay ticket price though their card. 5. Done	Book a ticket from Mohakhali to Uttara.	User should see a booking confirmation message.	As expected,	Passed
Post condition: N/A				

Test case 04:

Project name: Automated Ticket Issuing System for DSS.				
Test case ID: **TC_DSS_04**				
Test priority: High				
Module name: Cancel Ticket				
Test title: Cancel a ticket by administrator.				
Description: An administrator can cancel user’s booked ticket.				
Pre-Condition: Have to be logged into the system as an administrator.				
Steps	*Test Data*	*Expected Result*	*Actual Result*	*Status*
1.Login as administrator 2. Search for the ticket to cancel. 3. Select the ticket and cancel.	Cancel one user’s ticker	Administrator should see the ticket cancelling confirmation.	As expected,	Passed
Post condition: N/A				

Test case 05:

Project name: Automated Ticket Issuing System for DSS.
Test case ID: **TC_DSS_05**

Test priority: High				
Module name: Payment				
Test title: Payment system testing.				
Description: User can pay for his/her booked ticket.				
Pre-Condition: Have to select a train and number of tickets first.				
Steps	*Test Data*	*Expected Result*	*Actual Result*	*Status*
1. Select train and number of tickets. 2. Confirm booking. 3. Pay the amount though card.	Pay for a randomly selected ticket.	User should see the payment successful message.	As expected,	Passed
Post condition: N/A				

Item Pass/Fail criteria

TCI	Module name	Expected Input	Expected Result	Pass Rate	Fail Rate
TC_DSS_01	Ticket showing system	Mohakhali to Uttara train details	User should see the train information with fare and times.	100%	0%
TC_DSS_02	Finding Train	Find for a train Mohakhali to Uttara	User should see some trains based on the search.	85%	15%
TC_DSS_03	Book Ticket	Book a ticket from Mohakhali to Uttara.	User should see a booking confirmation message.	95%	5%
TC_DSS_04	Cancel Ticket	Cancel one user’s ticker	Administrator should see the ticket cancelling confirmation.	100%	0%
TC_DSS_05	Payment	Pay for a randomly selected ticket.	User should see the payment successful message.	92%	8%

Pass Rate: 94.4 % Rate of Error: 5.6 %.

Remaining test tasks

As we have worked with sample test cases and sample data for the project, we have shown some sample test cases and results for the project so that the testers and the other team members can understand the procedure of the testing plan and the excitation of those in a systematic way like above. Therefore, many test tasks are remailing to be tested for the project.

- ✓ GUI Testing
  - o Touch screen testing.
  - o Keyboard interface testing.
- ✓ Single ticket purchase testing
- ✓ Multiple ticket purchasing testing.
- ✓ Admin access testing.
- ✓ Ticket booking limitation testing.
- ✓ Payment validation on the server end.
- ✓ Ticket navigation testing.
- ✓ Testing of information display on the web.
- ✓ Ticket availability testing.
- ✓ Coin or Taka recognition and acceptance testing.

Features not to be tested

Before delivering the final product, the following features will not be tested. The Team for Growth to check these thresholds does not have the resources (hardware, software, and personnel).

- Restrictions on Scalability
- Time of response (to a limited extent)
- Dataset Size
- Availability of hardware

Since these features are not included in the specification of software specifications, they are not checked.

- Integrates with Applications
- Security for Website/Application

Approach

This test plan has a specific approach to meet the goal of successful testing of the project Dhaka Automated Ticket Issuing System for Dhaka Subway Systems (DSS). This approach consists of two steps.

- The first step is for analyzing the key phases of the software testing life cycle, required items for the test planning, and test design processes.
- The second step is to identify the attributes of the software test planning and the test design processes.

Test Deliverables

Test Deliverables are the test objects provided during a software project's stakeholders Life cycle for Software Creation. The numerous software projects that follow SDLC are subject to Phases until distribution to the client. A list is given below of some test deliverables.

Test Case: A test case is a document in one of the test objects that allows testers to create test cases for a specific test scenario to check whether or not the application's features function as intended. Test cases are collecting positive and negative executable steps of a test scenario with a set of pre-conditions, test data, expected outcome, post-conditions, and actual outcomes. The critical areas for test cases are listed below.

- Project name.
- Reference document.
- Module name.
- Created By.
- Date of creation
- Pre-condition
- Test case id.
- Test steps.
- Expected results

Test Plan: A document of the test plan contains the list for all the research activities to have a quality product. The document for the Test Plan is extracted from the Product. Overview, SRS, or Use Case documentation for all the potential project activities. The Test Lead or Test Manager usually prepares it, and the focus of the document is to explain what to test, what not to try, how to test, and who will do what test

Bug Report: The object of using the defect report template or the Bug report template is to convey the detailed report information (such as descriptions of the environment, replication steps, etc.) about the bug to the designers. It enables developers to replicate the error quickly. Bug Report Template Components:

- Defect ID
- Title/Summary
- Reporter Name
- Defect Reported Date
- Who Detected?
- How Detected
- Project Name
- Release/Build Version
- Defect/Enhancement, Environment
- Description
- Steps to Reproduce
- URL
- Expected Result

- Actual Result

Test summary report: It contains the summary of test activities and final test results which contains all the test results.

Environmental needs

A test environment is a server that allows the test cases that you have identified to be run. More than just setting up a server to run tests on, the test environment contains more. It also requires the configuration of hardware and networks. In other words, any time need to test a product, a test environment allows a tester to build similar environments. To have confidence in the testing results, it is an essential instrument for a testing engineer.

As per as the introduction, it takes several elements to build the right test setting. Here is a list of criteria when designing test environments that a test engineer will have to complete:

- Create and insert test
- data for test environments
- Database
- configuration
- Configure the atmosphere
- Choose the correct hardware and operating system
- Set up the Network

Documenting all acts is the most significant step which is essential to recreate the environment for other users. Comprehensive documentation helps the testing engineer set up various test environments, such as staging and development environments.

Staffing and Training needs

The software test plans must have some requirements on the stuffing and training. First of all, there will be some testers to execute the test plan. And the tester should have knowledge on the following:

- Knowledge of Programming
- Basic knowledge of Database (SQL)
- Basic commands of Linux
- Hands-on experiences on a Test Management Tool
- Hands-on experiences on a Test Automation Management Tool
- Hands-on experiences on Defect Tracking Tool

And during the test planning, there should also be some Training needs. Like:

- **Test Coverage:** Test coverage decisions are to be made during test planning. This will help to answer how much of the requirement, design, and code to be tested. Before starting the testing, these decisions are important to take.
- **Quality of Test Plan**: The quality of the test plan should also be introduced on the test plan. This will help the testers to compare different modules of the software and the quality of each module.
- **Goal:** There should be a goal of the testing, which can help the testers to be able to meet the goal as per the planning of the testing. That’s why it is also an important task to have.

Responsibilities

The research plan outlines the steps taken to ensure the reliability of the findings. Actual titles can significantly differ, but these positions are found in most non-trivial IT ventures. A dedicated QA team may be involved in any or all aspects of planning and researching many organizations. But if QA workers or another project performs team members' positions, they need to be accounted for in the project plan.

The project Test Lead-It is the person in charge of leading the testing of the project. This individual is also responsible for the procedures used to ensure the quality of the deliverables. It may be the project manager, or someone from QA, or someone else at all, but quality assurance control is the responsibility of this person.

Testing Manager-This is the person responsible for performing and conducting quality assurance testing on the test schedule. Different test managers may be assigned to different types of tests or other sections of the project, but a role is required if there is a test form.

Test Designer-This is the person in charge of designing the test scripts, scenarios, test lifespans, etc. The pass/failure criteria, also referred to as the 'right answer,' are part of these things, of course. Different test designers of different styles and even multiple designers working on the same test form may be present. Recall that (usually) the unit test designer is the developer.

Test Approver-This is the person responsible for the test material produced by the Test Designer for evaluating, validating, and approving.

Tester-He is the person who performs and records the results of the test scripts. Many testers can, at different times
or simultaneously, work on a project. Some might even be automated.

Reviewer: This entity must review the tester results and determine the steps that are subsequently taken, which may be the designers, project managers, programmers, DBA, and all other members of the team, but the testing role assists them in providing feedback on product quality.

Note: The reviewer is not usually the one who can "fix" any defect that has been confirmed to be a defect.

Schedule

Test Step	Start Date	End Date
A. Information Collecting		
• Prepare for interview	20-12-2020	22-12-2020
• Conduct interview	22-12-2020	24-12-2020
• Summarize findings	24-12-2020	25-12-2020
B. Test Planning		
• Build Test Plan	25-12-2020	28-12-2020
• Define the Metric Objectives	29-12-2020	30-12-2020
• Review/Approve Plan	30-12-2020	01-01-2020
C. Test Case Design		
• Design function test	01-01-2020	02-01-2020
• Design GUI test	02-01-2020	04-01-2020
• Define acceptance tests	04-01-2020	06-01-2020
• Review test designs	06-01-2020	07-01-2020
D. Test Development		
• Develop test scripts	07-01-2020	09-01-2020
• Review test development	09-01-2020	10-01-2020
E. Testing execution		
• Setup and testing	10-01-2020	12-01-2020
• Evaluation	12-01-2020	14-01-2020
F. Prepare for next sprint		
• Prepare documents	14-01-2020	15-01-2020
• Prepare other test items	15-01-2020	16-01-2020
G. Review test plans		
• Review test plans	16-01-2020	17-01-2020
• Review acceptance tests	17-01-2020	18-01-2020
H. Summarize test results	18-01-2020	20-01-2020
I. Prepare final test report	20-01-2020	24-01-2020
J. Review/Approve the final Test Report	24-01-2020	26-01-2020

Planning Risks and Contingencies

What are the overall risks to the project with an emphasis on the testing process?

- Lack of personnel resources when testing is to begin.
- Lack of availability of required hardware, software, data or tools.
- Late delivery of the software, hardware or tools.
- Delays in training on the application and/or tools.
- Changes to the original requirements or designs.
- Complexities involved in testing the applications

Specify what will be done for various events, for example: Requirements definition will be complete by January 1, 2021, and, if the requirements change after that date, the following actions will be taken:

The test schedule and development schedule will move out an appropriate number of days. This rarely occurs, as most projects tend to have fixed delivery dates.

- The number of tests performed will be reduced.
- The number of acceptable defects will be increased.
- Resources will be added to the test team.
- The test team will work overtime (this could affect team morale).
- The scope of the plan may be changed.
- There may be some optimization of resources.
- This should be avoided, if possible, for obvious reasons

Management is usually reluctant to accept scenarios such as the one above even though they have seen it happen in the past.

The critical thing to note is that the expected outcome is that testing is cut down or skipped entirely if a tester does nothing, none of which should be an appropriate choice.

Approvals

The undersigned acknowledge that they have checked and agree with the methodology in this Test Plan paper's presentations. Any corrections to this concept of specifications shall be coordinated with the undersigned or appointed representatives and approved by them.

Signature: _____________________________________ Date: ________
Name: _______________
Title: ________________
Role: ________________

Signature: ____________________________________ Date: ________
Name: _______________
Title: ________________
Role: ________________

Signature: _____________________________________ Date: ________
Name: ________________
Title: _________________
Role: _________________

Reference

- Slides provided by course faculty.
- Book - *Software Quality Engineering - Testing, Quality Assurance and Quantifiable Improvement*.
- https://www.softwaretestinghelp.com/
- https://testfort.com/qa-documentation