

American International University-Bangladesh (AIUB)  
Department of Computer Science  
Faculty of Science &Technology (FST)  
Spring 21 22

Section: C  
Software Quality Assurance and Testing

Automated Airline-Ticket Reservation System

A Report submitted

By

| SN | Student Name | Student ID |
| --- | --- | --- |
| 1 | Nahin Nasir | 18-36430-1 |
| 2 | Mohammed Shahriar Hossain | 18-37253-1 |
| 3 |  |  |
| 4 |  |  |

Under the supervision of

**Abhijit Bhowmik**

**Associate Professor**

Software Test Plan

for

Automated Airline-Ticket Reservation System

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

[Revision History 3](#_heading=h.gjdgxs)

[1.](#_heading=h.30j0zll) TEST PLAN IDENTIFIER: ATRS V-1.0 3

[2.](#_heading=h.1fob9te) REFERENCES 3

[3.](#_heading=h.3znysh7) INTRODUCTION 3

[Background to the Problem 3](#_heading=h.2et92p0)

[Solution to the Problem 4](#_heading=h.tyjcwt)

[4.](#_heading=h.3dy6vkm) REQUIREMENT SPECIFICATION 4

[4.1](#_heading=h.1t3h5sf) System Features 4

[4.2](#_heading=h.4d34og8) System Quality Attributes 4

[4.3](#_heading=h.2s8eyo1) System Interface 5

[4.4](#_heading=h.17dp8vu) Project Requirements 5

[5.](#_heading=h.3rdcrjn) FEATURES NOT TO BE TESTED 5

[6.](#_heading=h.26in1rg) TESTING APPROACH 5

[6.1](#_heading=h.lnxbz9) Testing Levels 5

[6.2](#_heading=h.35nkun2) Test Tools 6

[6.3](#_heading=h.1ksv4uv) Meetings 6

[7.](#_heading=h.44sinio) TEST CASES/TEST ITEMS 6

[8.](#_heading=h.2jxsxqh) ITEM PASS/FAIL CRITERIA 7

[9.](#_heading=h.z337ya) TEST DELIVERABLES 7

[10.](#_heading=h.3j2qqm3) STAFFING AND TRAINING NEEDS 8

[11.](#_heading=h.1y810tw) RESPONSIBILITIES 8

[12.](#_heading=h.4i7ojhp) TESTING SCHEDULE 8

[13.](#_heading=h.2xcytpi) PLANNING RISKS AND CONTINGENCIES 9

[14.](#_heading=h.1ci93xb) APROVALS 9

# Revision History

| Revision | Date | Updated by | Update Comments |
| --- | --- | --- | --- |
| 0.1 | 18.04.2022 | Nahin Nasir | First Draft |
| 0.2 | 19.04.2022 | Mohammed Shahriar Hossain | First Update |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# TEST PLAN IDENTIFIER: ATRS V-1.0

# REFERENCES

* Any reference documents with the test plan. For example:Software Requirement Specification (SRS) Document

# INTRODUCTION

## Background to the Problem

* Manual Air ticket reservation has become a big problem, The users usually buy their ticket manually and maybe wait in a queue to check the required flight.
* The importance of an airline reservation system is that it has a high potential to prevent mistakes that might arise when utilizing a manual system. Reservations are made easier, and the boarding procedure is speed up.
* But the existing Airline Reservation System have problems which are: passengers' inability to choose their chosen seat(s) from the reservation system, user interface is not optimal, there is no opportunity for travelers to print their boarding passes using the current systems. Passengers are not notified of flight cancellations or delays, and they do not have access to aircraft maintenance reports.

## Solution to the Problem

* What are the solutions you are going to propose to deal with the problem? why is this   
   solution is particularly appropriate to solve the problem? Is the solution feasible to the   
   meet the business objective?
* Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals
* Existing studies presented in the problem area. What are the existing software solutions   
   are available to solve the aforementioned problem?

# REQUIREMENT SPECIFICATION

## System Features

Common Features:

1. System Registration  
Functional Requirements

* 1. the system shall allow user to create a user account with their given information’s.
  2. if phone no and passport number are not valid, system won’t allow user to create account.
  3. password and confirm password should match.

Priority Level: High  
Precondition: Valid Phone number and Passport number.

2. System Login

functional requirements

2.1 The software shall allow users to login with their given username and password

2.2 If the username and/or password has been inserted wrong for more than three times, there will be a cooldown timer generated by the system asking user to retry login later.

Priority Level: high.

Precondition: Valid User name and Password.

3.Profile View/Edit.

functional requirements

2.1 All the users can view their own profile after successful login through their dashboards.

2.2 All the users can edit their own profile after successful login through their dashboards.

Priority Level: Medium

Precondition: Successful Login.

System Admin Features:

1. Add/remove Flights.
2. Add/remove Users.
3. Add/remove Discount Offers.
4. Manage Ticket reservation.
5. View routes.
6. View airport.
7. View Users.
8. Manage seats.
9. View user transactions.
10. Maintain reports.

System User Features:

1. Search Flights.
2. View/Select seats.
3. Pre-book tickets.
4. Purchase tickets.
5. Cancel tickets.
6. Print Boarding Pass.
7. View Transaction history.
8. Select payment method.
9. Submit Reports.
10. Get Maintenance Notifications.

## System Quality Attributes

Quality may be defined in a number of different ways. Quality is defined differently by each individual. Finally, certain ground rules should be set. If a product is user-friendly and contains all of the required functionalities, it is said to be of high quality. Quality Control Activities are aimed at discovering flaws in products and services, whereas Quality Assurance Activities or Attributes are concerned with avoiding faults from being introduced. As stated in the statement, the following can be employed to assure the greatest quality for all of the company's goods.

**Reliability:** Check the product's dependability to verify if it can resist any condition.

Furthermore, the findings must be accurate on a consistent basis. The performance of a

project in diver working contexts and situations determines its dependability.

**Maintainability:** The product should be straightforward to maintain. It should be easy to add code to an existing system for development and to update for new features and technologies as they become available. Maintenance should be both economical and straightforward. The system is straightforward to maintain, and making modifications to the program or correcting problems is simple.

**Usability:** This is something that can be measured in terms of usefulness. It should be straightforward to utilize the program. It should also be easy to understand. The navigation should be simple to understand. In terms of input preparation, operation, and output interpretation, the system must be simple to use. Ascertain that our other widely used systems follow the same user interface standards or practices. For new or infrequent users, the system is straightforward to understand.

**Portability:** It is possible to quantify cost concerns associated to porting, technical obstacles related to porting, and behavioral issues related to porting.

**Correctness:** The functionality, internal computations, and navigation of the program should all be correct. This implies that the app must fulfill all of the functional specifications.

**Efficiency:** A significant system quality attribute that is measured in terms of how long it takes the system to complete a job. The system should, for example, make the best use of CPU power, disk space, and memory. The user's performance will suffer if the system utilizes all available resources, and the system will be termed inefficient. If a system is inefficient, it cannot be used in real-time applications.

**Security:** Integrity and safety are intricately bound. The integrity or security of the system should be adequate to prevent unauthorized access to system operations, data loss, and software virus infection, as well as to preserve the privacy of data submitted into the system**.**

**Testability:** Testing and detecting weaknesses in the system should be straightforward. If required, it should be straightforward to split it down into distinct modules for testing.

## System Interface

* Draw the system interface where the users will interact with the system’s functionality.

## Project Requirements

* Development time 60 days
* Total budget 60,000 BDT
* Sufficient storage in system
* Visual Studio code [an open-source IDE software]
* MySQL database
* Appropriate environment and genuine group collaboration as it is a

group project

* Optimum knowledge on Software Quality and Testing.

# FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts.

* 24/7 availability.
* Maintenance, opening or closing of account, create, update, or delete customer and credit/debit card records.
* Each user's USER ID and PASSWORD should not be disclosed to the tester
* Limiting the number of tickets purchased at the same time or not.