**City Tech Innovations (Group 21)**

**Testing plan**

**Client:**

• Mr Lancaster (AirVia LTD)

**Team members:**

* Rati Sturua (Team Manager and Developer)
* Abbib Akram (Deputy Manager and Systems Analyst)
* Aymen Said (Developer and Systems Designer)
* Ethan Brewer (Developer and Systems Designer)
* Ishmail Hoque (Systems Analyst and Tester)
* Uzair Mahmood (Systems Analyst and Tester)
* Jocubas Stankaitis (Developer and Tester)

**Author:**

* Abbib Akram
* **Version History:**

**Table of Contents**

Scope of testing:

In this project we intend to test the ten core functionalities of the AirVia ticketing system (ATS). The ten core functionalities are: voidBlank, produceBlankReport, createSalesReport, payLater, createIndividualReport, reassignBlank, cancel Ticket, recordTransaction, sellTickets, alteringTravelAdvisor.

We will be testing the core functionalities using unit testing which runs tests on specific areas and functionalities of a software product by inputting some test data to see how it will behave in response.

Test objectives:

The objectives of testing this software is to test the features which are required according to the specification, testing features which are not stated within the specification but may enhance system performance and ensuring stability throughout the product lifecycle, In addition to checking for bugs.

Test strategy:

The strategy for our testing will be the following:

Whenever a specific unit of our testing is complete a tester will complete unit tests on the unit.

When the whole system is complete, we will carry out the use test cases, which will allow us to find irregularities between the specification and the system that has been developed.

The findings will be reported directly to the developer responsible for the unit, which was tested, and the necessary changes will be made to the code if the test fails.

This will be a recursive process, until all the tests have passed.

Tools used for testing:

UML Use test cases

IntelliJ idea unit testing tool

UML test cases:

We intend to carry out use case testing on ten key use cases: …

|  |
| --- |
| Use case ID:TM01 |
| Use case name:VoidBlank |
| Test number:1 |
| Objective: To test the main flow |
| Set up: The tester will be logged in and will be commencing the creation of a blank. |
| Expected results:  1.The system will output: “Are you sure?”  2.The system will terminate the flow and return to the previous state.  3.The system no longer recorded as issued. |
| Test:  1.The tester will start the creation of a use case.  2.The tester will then make a mistake.  3.The actor will select ‘yes’ when the system asks them if they want to continue.  4.The tester will check if the flow has been terminated and that the blank has no longer been recorded. |
| Test record: |
| Date: |
| Tester: |
| Result: |
| Date: |
| Tester: |
| Result: |

|  |
| --- |
| Use case ID:TM02 |
| Use case name: ProduceBlankReport |
| Test number:2 |
| Objective: To test the main flow |
| Set up:  The tester will login and select ‘produceBlankReport”. |
| Expected results:  1.The system displays the contents of the report on the screen  2.The tester would be able to choose from the options: “used blanks”, “turnout” and “monthly turnover”. |
| Test:  1.The tester will click on “Create a blank report”.  2.The tester will check whether a window opens, prompting the tester to choose the options: “Used blanks”, “turnover “and “monthly turnover”.  3.The tester will select one of these options.  4.The tester will check whether the system displays the data on the screen. |
| Test record: |
| Date: |
| Tester: |
| Result: |
| Date: |
| Tester: |
| Result: |

|  |
| --- |
| Use case ID:TM02.1 |
| Use case name:noBlankUsed |
| Test number:2.1 |
| Objective: Test the alternative flow where there have been no used blanks. |
| Set up: The tester will log on and click on the “create a blank report”. |
| Expected results:  1.Message stating that there are “No blanks used”.  2.The system returns to the main dashboard |
| Test:  1.The tester clicks on create reports  2.The tester will check to see if the system states that a report on the number of blanks cannot be produced as there are no blanks being used. |
| Test record: |
| Date: |
| Tester: |
| Result: |
| Date: |
| Tester: |
| Result: |

|  |
| --- |
| Use case ID:TM03 |
| Use case name: createSalesReport |
| Test number:3 |
| Objective: Test the main flow |
| Set up: The tester logs in to the system and clicks on createSalesReport |
| Expected results:  1.The system displays the options: “create individual report” and” Create summary”.  2.System displays:  Create reports per advisor.  Create reports per USD rate.  Display sub total.  Display grand total. |
| Test:  1.The tester clicks: “Create sales report”  2.The tester clicks “Create summary”.  3.The tester clicks “Create individual report: |
| Test record: |
| Date: |
| Tester: |
| Result: |
| Date: |
| Tester: |
| Result: |

|  |
| --- |
| Use case ID:TM04 |
| Use case name:payLater |
| Test number:4 |
| Objective: Test the main flow. |
| Set up: The tester is logged in |
| Expected results:  1.Tester selects “pay later” or “decline”.  2.If the payment is not received within thirty days, the system should output a notification to the tester, stating this. |
| Test:  1.The tester will choose if the client wants to pay thirty days later.  2.The tester will accept the “paylater” request.  3.If the payment is not logged within the database within 30 days, then the notification will be sent. |
| Test record: |
| Date: |
| Tester: |
| Result: |
| Date: |
| Tester: |
| Result: |