Airline Crew Scheduler

Software Design Document

Team Name

Ryan Witkowski

Nicholas Lawrance

James Fox

# Introduction

The objective of the listed tests ran on the current Airport-Database-Scheduler were checking the login and logout system to be able to access and change the data. Adding a flight to the database, changing a current flight already made and cancelling the flight. We also ran a test on qualifications to check the crew for the flights, so that only the qualified were in the correct positions. We are additionally adding test cases to break down add flight and check some interfaces.

# System Overview

The system is an airline crew scheduler that will manage the flight crew of an airline. The system will be able to handle different calls and changes that are made to the flights and workers. The workers will be able to view flight information without being able to make changes if they’re not an admin.

# Features to be Tested

|  |  |  |
| --- | --- | --- |
| Testing  Requirement | System  Requirement(s) | Short Description |
| 1 | F1, N1, N2, N3, N4, N5 | System test of being able to Login to the system, when you have an authorized username and password. |
| 2 | F1, N1, N2, N3, N4, N5 | System test of not allowing unauthorized guests to be able to log in to the database. |
| 3 | F1, F5, N1, N2, N3, N4, N5 | System test of the flight cancelling interface menu and method. |
| 4 | F1, F5, N1, N2, N3, N4, N5 | System test of the whole add flight menu and options to make sure everything works. |
| 5 | F1, F5, N1, N2, N3, N4, N5 | System test of how an authorized user navigates the menu’s to cancel a flight. |
| 6 | F1, F5, N1, N2, N3, N4, N5 | System test of how the user interacts with a drop down menu. |
|  |  |  |

# Test Environment

The environment that was needed was first a computer that is able to download and run Intellij Community and has the classes and functions that the test cases are using.

GitLab repository.

# Test Cases

**Test Case- Login and Logout Testing**

**Test Case Number:** 1

**Component Under Test:** Airline-Crew-Scheduler

**Feature(s) to be tested:** We will be checking to see if a matched username and password for an individual will be successful or not, if inputted correctly we will be directed into the database upon successful login attempt.

**Initial Conditions:** The GUI Airport database scheduling manager user interface will have a interface menu with an input box for both username and password and a sign in and back button option. The database will already have a stored username and password that both must match the inputted username and password the user has used.

**Expected Behavior: (Action Response)**

1. The user will begin in the main menu and select the “Sign In” button option
2. The system will redirect the user to the sign in menu
3. (Side note)The user can then press back if they do not have a username and password for the page and the system will redirect the user back to the main menu interface.
4. The user enters their authorized username and password keywords into the designated matching boxes in the login screen.
5. The user will then press “Sign In”.
6. The system then checks the username and password of the individual to the stored username and password data for each authorized username and password match to look for a match.
7. If found the system will either login to the database and the user will be redirected to the logged in menu or prompt “Invalid Password”.
8. The system will redirect the user to the Logged in menu.
9. The user can then press the Logout button option
10. The system will then the sign the user out of the database
11. The system will redirect the user to the main menu interface

**Test Case- Login and Logout Testing for invalid password**

**Test Case Number:** 2

**Component Under Test:** Airline-Crew-Scheduler

**Feature(s) to be tested:** We will be checking to see if a matched username and password for an individual will be successful or not, if inputted correctly we will be directed into the database upon successful login attempt.We will be inputting an incorrect username and checking to make sure that only authorized users allowed to sign in.

**Initial Conditions:** The GUI Airport database scheduling manager user interface will have a interface menu with an input box for both username and password and a sign in and back button option. The database will already have a stored username and password that both must match the inputted username and password the user has used.

**Expect Behavior:**

**Input:**

1. The user enters a wrong username and password keywords into the designated matching boxes in the login screen.
2. The user will then press “Sign In”.
3. The system prompted the user with “Invalid Password” because of an incorrect username.

**Output:**

The system prompted the user with “Invalid Password” because of an incorrect username.

**Test Case- (Flight Changed)**

**Test Case Number:** 3

**Component Under Test:** Airline-Crew-Scheduler

**Feature(s) to be tested:** We will be checking to see how the database reacts to when the admin makes changes to a flight the system updates correctly for the change of a flight for its departure and arrival. We will be checking the user interface manager for the flight change form and the “FlightChange” class, which extends the “Flight” class.

**Initial Conditions:** The GUI Airport database scheduling manager user interface will be able to make adjustments to a flight by using the flight change form on the GUI. The database will need to have the working functions from the “FlightChange” class for various departures and arrival getters and setters and flight number. The flight number when selected will need to be linked to the plane that matches the ID. The user will have to be already logged in to be able to make changes to the flight and not be excluded to only be able to view flights.

**Expected Behavior: (Action Response)**

1. The user will be at the main menu and will select the sign in button.
2. The system will redirect the user to the sign in menu.
3. The user will then sign in to the login with their username and password.
4. The system will redirect the user to the logged in menu.
5. Once the user is logged in to the database manager. He will choose Change Flight button option
6. The system will redirect the user to the Change Flight menu.
7. The user will be prompted with a drop down box for choosing the Flight Number.
8. The system will provide a list of Flight Numbers to be modified in the first drop down menu box.
9. The user will choose the Flight Number they will be changing in the menu box.
10. The system will select the Flight Number chosen and have the fly selected in the Flight Number Box.
11. The user will then select the set the Flight Time in the next text box and select the apply button option when finished
12. The system will update that Flight Number change to the database

**Test Case- (Flight Scheduled)**

**Test Case Number:** 4

**Component Under Test:** Airline-Crew-Scheduler

**Feature(s) to be tested:** We will be checking the forms and how they interact with the classes. If the getters and setters work and if a flight can be implemented and made properly with the given functions already made in the “FlightScheduled” class that extends from the “Flight” class.

**Initial Conditions:** The GUI Airport database scheduling manager user interface will be able to add a new flight to the database. The database will have a add flight form, which will be able to add a new flight by specifying the required field of information. The user will have to be already logged in to be able to make changes to the flight and not be excluded to only be able to view flights.

**Expected Behavior: (Action Response)**

1. The user will be at the main menu and will select the sign in button.
2. The system will redirect the user to the sign in menu.
3. The user will then sign in to the login with their username and password.
4. The system will redirect the user to the logged in menu.
5. The user will then choose the “Add Flight” button option.
6. The system will redirect the user to the Add Flight menu.
7. The user will then type the input keywords in the required boxes (Plane Number, Flight Number, Scheduled Departure, Estimated Departure, Actual Departure). And then select the weather forecast for the time of the flight and the origin airport and destination airport.
8. The system will then take the user input and create a flight object and that object is passed to a method that will create the flight into the database.
9. (Side Note)The system will look through all of the flights and self-assign crew (crew is not apart of the main Add Flight schedule) based on the checks that we implemented.

**Test Case- (Flight Cancelled)**

**Test Case Number:** 5

**Component Under Test:** Airline-Crew-Scheduler

**Feature(s) to be tested:** We will be checking the database user interface menu forms and how they interact with the classes. If the getters and setters work and if a flight can be implemented and made properly with the given functions already made in the “FlightCancelled” class that extends from the “Flight” class.

**Initial Conditions:** The user will have to be already logged in to be able to make changes to the flight and not be excluded to only be able to view flights. The GUI Airport database scheduling manager user interface will have menus for cancelling a flight. The database will have getters and setters for flight cancelling and will output the modification. The user will have to be already logged in to be able to make changes to the flight and not be excluded to only be able to view flights.

**Expected Behavior: (Action Response)**

1. The user will be at the main menu and will select the sign in button.
2. The system will redirect the user to the sign in menu.
3. The user will then sign in to the login with their username and password.
4. The system will redirect the user to the logged in menu.
5. The user will select the Cancel Flight button option.
6. The system will redirect the user to the Cancelled Flight page.

(To still Implement)

1. The user will then select the “Flight Number” from the list of flights.
2. The system will then retrieve and select the flight information
3. The user will then select cancel flight by pressing the cancel flight button option
4. The system will then remove the flight from the database and remove the crew out of active status and place them to standbycrew.

**Test Case- (Flight Scheduled)**

**Test Case Number:** 6

**Component Under Test:** Airline-Crew-Scheduler

**Feature(s) to be tested:** We will be checking the drop down menu for destination airport and it being selected and showing in the box menu.

**Initial Conditions:** The GUI Airport database scheduling manager user interface will be able to add a new flight to the database. The database will have a add flight form, which will be able to add a new flight by specifying the required field of information. The database will already have the stored airports, that will be on the drop down menu. The user will have to be already logged in to be able to make changes to the flight and not be excluded to only be able to view flights.

**Expected Behavior:**

**Input:**

1. The user will move the cursor over the drop down menu
2. The user will select from a list of airports and click on the airport they want
3. After selection the menu box for the drop down menu should select the destination airport and have it appear in the box.

**output:**

The selected destination airport appeared in the box that was chosen.

**Test Case- (Qualifications) (Not tested yet)**

**Test Case Number:** 7

**Component Under Test:** Airline-Crew-Scheduler

**Feature(s) to be tested:** We will be checking to make sure the Boolean function will return true or false and allow for the crew member to be added if passed “true. If false then the crew member will not be able to be added. The

**Initial Conditions:** The GUI Airport database scheduling manager user interface will have methods that will check crew qualifications. The user will be able to add/remove a crew member to a flight. The user will have to be already logged in to be able to make changes to the flight and not be excluded to only be able to view flights.

**Expected Behavior: (Action Response)**

1. The user will be at the main menu and will select the sign in button.
2. The system will redirect the user to the sign in menu.
3. The user will then sign in to the login with their username and password.
4. The system will redirect the user to the logged in menu.
5. The user will be then select the Edit Crew button option.
6. The system will then redirect the user to the Edit Crew menu.

(To Still Implement)

1. The user will be able to assign a crew member to a flight by inputting their personId into the crew member box.
2. The system will apply that crew members id to that flight.
3. The user will then press the apply button option.
4. The system will then modify the flight changes made and append it to the database.