**CST338**

**Homework 7**

**Otter Airways Flight Reservation System**

**Objective:**

* implement an object-oriented android application with multiple activities using room and a database
* read and understand application requirements presented in the form of use cases.
* there are 15 test cases for this application. You will create a document with a summary of the 15 test cases along with 15 screen shots showing that your application works for each test case on your laptop.
* the application should start with the usernames and passwords defined on page 2 and the special admin userid !admiM2
* and the application starts with the 5 flights defined on page 4
* Upload and submit to ilearn
  + the document containing the summary table of 15 tests and the 15 screenshots of test executions
  + a zip file (must be zip file) of your application

**Requirements:**

**Use Case Diagram**



**Use Case – Create Account**

**1. Basic Flow**

{Display Main Menu}

(1) The use case begins when the System prompts the Customer with options of “Create Account”, “Reserve Seat”, “Cancel Reservation”, and “Manage System”.

(2) The Customer selects the “Create Account” option.

{Create Account}

(3) The Customer enters a username and a password.

(4) The System verifies that the username and password are valid. In the project, the username and password should have at least one special symbol (!, @, #, or $), one number, one uppercase alphabet, and one lowercase alphabet. Note that **!admiM2** is reserved for the System Operator. Thus, the Customer can’t use it as his/her username. Additionally, we assume that the System already has three usernames/passwords as below:

|  |  |
| --- | --- |
| **Username** | **Password** |
| A@lice5 | @cSit100 |
| $BriAn7 | 123aBc## |
| !chriS12! | CHrIS12!! |

(5) The System informs the Customer that his/her account has been created successfully.

{Log the Transaction}

(6) The System records the transaction information that includes a transaction type (= new account), the Customer’s username, and current date/time (= hour and minute).

{Use Case Ends}

(7) The System displays the main menu.

(8) The use case ends.

**2. Alternative Flows**

2.1 Handle Incorrect Username/Password

At {Create Account}, if the Customer’s username and/or password do not meet the required criteria (= at least one special symbol (!, @, #, or $), one number, etc),

1. The System informs the Customer that the username and/or password are not in the correct format.

2. The Customer is prompted to enter new username and password to create an account.

3. If the Customer fails again,

a. The System displays an error message.

b. The Customer confirms the error message.

c. The System displays the main menu.

d. The use case ends.

2.2 Handle Duplicate Account

At {Create Account}, if the Customer’s username already exists in the System,

1. The System informs the Customer that the username already exists.

2. The Customer is prompted to enter new username and password to create an account.

3. If the Customer fails again,

a. The System displays an error message.

b. The Customer confirms the error message.

c. The System displays the main menu.

d. The use case ends.

**Use Case – Reserve Seat**

**1. Basic Flow**

{Display Main Menu}

(1) The use case begins when the System prompts the Customer with options of “Create Account”, “Reserve Seat”, “Cancel Reservation”, and “Manage System”.

(2) The Customer selects the “Reserve Seat” option.

{Enter Departure and Arrival Information}

(3)The System prompts the Customer to enter the departure and arrival information he/she wants to reserve. Since our Otter Airways is a small company at the moment, we serve only three cities such as Monterey, Los Angeles, and Seattle.

(4) The Customer enters the number of tickets he/she wants to reserve. In the project, we assume that a Customer can’t reserve more than 7 tickets.

{Select Flight}

(5) Based on the departure, arrival, and number of tickets selected by the Customer, the System displays all flights available for the reservation. In our case, we assume that there are five flights as below when the System starts:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Flight No.** | **Departure** | **Arrival** | **Departure Time** | **Flight Capacity** | **Price** |
| Otter101 | Monterey | Los Angeles | 10:30(AM) | 10 seats | $150.00 |
| Otter102 | Los Angeles | Monterey | 1:00(PM) | 10 seats | $150.00 |
| Otter201 | Monterey | Seattle | 11:00(AM) | 5 seats | $200.50 |
| Otter205 | Monterey | Seattle | 3:45()M) | 15 seats | $150.00 |
| Otter202 | Seattle | Monterey | 2:10(PM) | 5 seats | $200.50 |

(6) The Customer picks a flight that he/she would like to use.

{Login to System}

(7) The System prompts the Customer to enter his/her username and password.

(8) The Customer enters the username and password.

(9) The System confirms that the Customer information is valid.

{Confirmation}

(10) The System displays the Customer username, flight number, departure/arrival information, number of tickets, reservation number, and total amount owed.

(11) The System prompts the Customer if the information is correct.

(12) The Customer verifies that the information is correct.

{Log the Transaction}

(13) The System records the transaction information that includes a transaction type (= Reserve Seat), the customer username, flight number, departure/arrival information, number of tickets, a reservation number, a total amount, and current date/time.

{Use Case Ends}

(14) The System displays the main menu.

(15) The use case ends.

**2. Alternative Flows**

2.1 Handle Invalid Username or Password

At {Login to System}, if the Customer’s login information is not valid,

1. The System informs the Customer that the information entered is not valid.

2. The Customer is prompted to re-enter the username and password.

3. If the Customer fails again,

a. The System displays an error message.

b. The Customer confirms the error message.

c. The System displays the main menu.

d. The use case ends.

2.2 Handle No Seats Available

At {Enter Departure and Arrival Information}, if the Customer requests more than 7 tickets,

1. The System informs the Customer that the reservation can be made due to the system restriction.

2. The Customer confirms the error message.

3. The use case returns to {Enter Departure and Arrival Information} to prompts again the Customer to enter the information he/she wants to reserve.

2.3 Handle No Flight Available

At {Select Flight}, if there’s no flight available,

1. The System informs the Customer that there is no flight available for the information the Customer entered.

2. The System prompts the Customer to exit the program.

3. The Customer clicks the “Exit” and the System displays the main menu.

4. The use case ends.

**Use Case – Cancel Reservation**

**1. Basic Flow**

{Display Main Menu}

(1) The use case begins when the System prompts the Customer with options of “Create Account”, “Reserve Seat”, “Cancel Reservation”, and “Manage System”.

(2) The Customer selects the “Cancel Reservation” option.

{Login to System}

(3) The System prompts the Customer to enter his/her username and password.

(4) The Customer enters the username and password.

(5) The System confirms that the Customer information is valid.

{Display Reservations}

(6)The System displays all reservations the Customer made with the reservation number, flight number, departure/arrival information, departure time, and the number of tickets reserved.

{Cancel Reservation}

(7) The Customer selects a reservation to cancel.

(8) The System asks the Customer if he/she really wants to cancel the reservation.

(9) The Customer confirms the cancellation.

{Log the Transaction}

(10) The System records the transaction information that includes the Customer’s username, reservation number, flight number, departure/arrival information, departure time, number of tickets reserved, transaction type (=cancelation), and current date/time.

{Use Case Ends}

(11) The System displays the main menu.

(12) The use case ends.

**2. Alternative Flows**

2.1 Handle No Reservation

At {Display Reservations}, if the Customer doesn’t have any reservation at the moment,

1. The System informs the Customer that there is no reservation with the username.

2. The Customer verifies this to continue.

3. The System displays the main menu.

4. The use case ends.

2.2 Handle No Cancel Confirmation

At {Cancel Reservation}, if the Customer doesn’t confirm the cancellation and selects a disregard button,

1. The System informs the Customer that the cancellation has failed.

2. The Customer verifies this.

3. The System displays the main menu.

4. The use case ends.

**Use Case – Manage System**

**1. Basic Flow**

{Display Main Menu}

(1) The use case begins when the System prompts the Customer with options of “Create Account”, “Reserve Seat”, “Cancel Reservation”, and “Manage System”.

(2) The System Operator selects the “Manage System” option.

{Login to System}

(3) The System prompts the System Operator to enter his/her username and password.

(4) The System Operator enters the username and password. In the project, we assume that the System Operator uses **!admiM2** as a username and **!admiM2** as a password.

(5) The System confirms that the information System Operator entered is valid.

{Display Log Information}

(6) The System displays all transaction logs.

(7) The System Operator confirms it to continue.

{Add New Flight Information}

(8) The System asks the System Operator if he/she wants to add new flight information to the System.

(9) If the System Operator selects the NO option, the use case moves to {Use Case Ends}. However, if the System Operator selects the YES option, the System lets the System Operator enter the flight number, departure/arrival information, departure time, flight capacity, and price information.

(10) The System displays the information entered and asks if the information is correct.

(11) The System Operator confirms it.

{Use Case Ends}

(8) The System displays the main menu.

(9) The use case ends.

**2. Alternative Flows**

2.1 Handle No Log Information

At {Display Log Information}, if the System has no log information to display,

1. The System informs the System Operator that there is no log information at the moment.

2. The System Operator confirms.

3. The use case moves to {Add New Flight Information}.

2.2 Handle Invalid Flight Information

At {Add New Flight Information}, if the System Operator missed any information or the flight already exists in the System,

1. The System informs the System Operator that the information entered is not valid.

2. The System Operator confirms the error message.

3. The System displays the main menu.

4. The use case ends.

**Summary of Test Case Execution Result**

**Include this updated table (with Pass/Fail) for each test in your document.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No** | **Result** | **Test Case No** | **Result** |
| Test case 1 | **PASS** | Test case 9 | **PASS** |
| Test case 2 | **PASS** | Test case 10 | **PASS** |
| Test case 3 | **PASS** | Test case 11 | **PASS** |
| Test case 4 | **PASS** | Test case 12 | **PASS** |
| Test case 5 | **PASS** | Test case 13 | **PASS** |
| Test case 6 | **PASS** | Test case 14 | **PASS** |
| Test case 7 | **PASS** | Test case 15 | **PASS** |
| Test case 8 | **PASS** |  |  |

**Test Case descriptions**

# Note: You have to include screenshot(s) of each test case as evidence to show that your app works well on your laptop.

1. When your app starts, it should provide the main menu GUI with four options of “Create Account”, “Reserve Seat”, “Cancel Reservation”, and “Manage System”.
2. Create a correct account

- Select "Create Account"

- Username: !!Byun7

- Password: !!Byun7

🡺 The System informs the Customer that the new account has been created successfully.

After that, the System should display the main menu.

1. Manage the system - Verify the new account

- Select "Manage System"

- Username: !admiM2

- Password: !admiM2

🡺 The system should display

1. Transaction type: New account

2. Customer's username: !!Byun7

3. Transaction date: Depending on the test case 2 execution date.

4. Transaction time (= hour and minute): Depending on the test case 2 execution time.

- In this test case, the System Operator doesn’t add new flight information.

🡺 The System should display the main menu.

1. Create an incorrect account

- Select "Create Account"

- Username: byun7

- Password: !byun7

🡺 The System should display an error message.

- Username: byun7!

- Password: 7byun

🡺 The System should display an error message again.

The Customer confirms the error message.

Then, it displays the main menu.

1. Create an incorrect account - Duplicated username

- Select "Create Account"

- Username: $BriAn7

- Password: 123aBc##

🡺 System should display an error message.

- Username: !!Byun7

- Password: !!Byun7

🡺 The system should display an error message again.

The Customer confirms the error message.

Then it displays the main menu.

1. Reserve seat

- Select "Reserve Seat"

- Departure: Monterey

- Arrival: Seattle

- Number of tickets: 2

🡺 System should display two possible flights of Otter201 and Otter205

1. The Customer picks Otter201

2. The Customer enters Username: !!Byun7

Password: !!Byun7

3. The System confirms it and displays

- Username: !!Byun7

- Flight number: Otter201

- Departure: Monterey, 11:00(AM)

- Arrival: Seattle

- Number of tickets: 2

- Reservation number: Depending on the test.

- Total amount: $401.00

4. The Customer confirms it

5. The System displays the main menu.

1. Reserve seat - One more trial

- Select "Reserve Seat"

- Departure: Los Angeles

- Arrival: Monterey

- Number of tickets: 6

🡺 System should display one possible flight of Otter102

1. The Customer picks Otter102

2. The Customer enters Username: !!Byun7

Password: !!Byun7

3. The System confirms it and displays

- Username: !!Byun7

- Flight number: Otter102

- Departure: Los Angeles, 1:00(PM)

- Arrival: Monterey

- Number of tickets: 6

- Reservation number: Depending on the test.

- Total amount: $900.00

4. The Customer confirms it

5. The System displays the main menu.

1. Reserve seat - Handle No Seats Available

- Select "Reserve Seat"

- Departure: Monterey

- Arrival: Los Angeles

- Number of tickets: 10

🡺 System should inform the Customer that the reservation can’t be made due to the system restriction.

The Customer confirms the error message.

- Departure: Monterey

- Arrival: Los Angeles

- Number of tickets: 5

🡺 System should display one possible flight of Otter101

1. The Customer picks Otter101

2. The Customer enters Username: !!Byun7

Password: !!Byun7

3. The System confirms it and displays

- Username: !!Byun7

- Flight number: Otter101

- Departure: Monterey, 10:30(AM)

- Arrival: Los Angeles

- Number of tickets: 5

- Reservation number: Depending on the test.

- Total amount: $750.00

4. The Customer confirms it

5. The System displays the main menu.

1. Reserve seat - Handle No Flight Available

- Select "Reserve Seat"

- Departure: Seattle

- Arrival: Los Angeles

- Number of tickets: 1

🡺 The System informs the Customer that there is no flight available for the departure/arrival.

The System prompts the Customer to exit the program.

The Customer clicks the “Exit” and the System displays the main menu.

1. Cancel reservation - Handle No Reservation

- Select "Cancel Reservation"

- Username: !chriS12!

- Password: CHrIS12!!

🡺 System should display

1. The System informs the Customer that there is no reservation with the username.

2. The Customer verifies this to continue.

3. The System displays the main menu.

1. Cancel reservation

- Select "Cancel Reservation"

- Username: !!Byun7

- Password: !!Byun7

🡺 System should display

1. All three reservations of “!!Byun7” with the detailed information.

- Select "Otter102" to cancel it.

1. The System requests confirmation and the Customer confirms it.

2. The System should display the main menu.

1. Manage the system - Verify the cancellation

- Select "Manage System"

- Username: !admiM2

- Password: !admiM2

🡺 The System should display all transactions so far. But our focus in this test case is the last transaction which should include

1. Transaction type: Cancellation

2. Customer's username: !!Byun7

3. Flight number: Otter102

4. Departure: Los Angeles, 1:00(PM)

5. Arrival: Monterey

6. Number of tickets: 6

7. Reservation number: Depending on the test.

8. Transaction date: Depending on the test case 11 execution date.

9. Transaction time (= hour and minute): Depending on the test case 11 execution time.

- In this test case, the System Operator doesn’t add new flight.

🡺 The System should display the main menu.

1. Manage the system - Add a new flight

- Select "Manage System"

- Username: !admiM2

- Password: !admiM2

🡺 The System should display logs. But it's not important at this test case.

- The System Operator selects the YES option to add a new flight.

1. Flight number: Otter301

2. Departure: Los Angeles

3. Arrival: Seattle

4. Departure time: 12:00(PM)

5. Flight capacity: 10

6. Price: $350.50

🡺 The System requests confirmation and the System Operator confirms it.

After that, the System should display the main menu.

1. Manage the system - Handle Invalid Flight Information

- Select "Manage System"

- Username: !admiM2

- Password: !admiM2

🡺 The system will display logs. But it's not important at this test case.

- The System Operator selects the YES option to add a new flight.

1. Flight number: Otter205

2. Departure: Seattle

3. Arrival: Los Angeles

4. Departure time: 2:00(PM)

5. Flight capacity: 5

6. Price: $300.00

🡺 The System informs the System Operator that the information entered is not valid. In the test case, note that the flight number, Otter205, already exists.

The System Operator confirms the error message and the System should display the main menu.

1. Reserve seat - One more trial

- Select "Reserve Seat"

- Departure: Los Angeles

- Arrival: Seattle

- Number of tickets: 1

🡺 System should display one possible flight of Otter301

1. The Customer picks Otter301

2. The Customer enters Username: !!Byun7

Password: !!Byun7

3. The System confirms it and displays

- Username: !!Byun7

- Flight number: Otter301

- Departure: Los Angeles, 12:00(PM)

- Arrival: Seattle

- Number of tickets: 1

- Reservation number: Depend on the test.

- Total amount: $350.50

4. The Customer confirms it

5. The System displays the main menu.