Student Name: **Weight: 20%**

Student ID: **Marks: /100**

Assignment: Abstract Classes, Event-Driven Applications and Exception Handling

Scenario

A local travel agency, Traveless, has hired your company to implement a new flight reservation management system to improve its productivity and services. After meeting with the agency, you realize that to tackle this complex request, you must break down the solution to three major parts:

- Front-end GUI
- Backend
- Functional front end connected to the backend

Equipment and Materials

For this assignment, you will need:

- Visual Studio IDE
- Supplied data files:
  - flights.csv
  - airports.csv

Instructions

1. Review the scenario, and then carefully read the Traveless program Details and Program Guidelines sections of this document.
2. Working outside of class time, complete the submission sections of this assignment.
3. Review the grading criteria for the group submissions.
4. See the course schedule and/or Brightspace for due dates.

Submission

1. Create the code for a program that meets the requirements described bellow.
2. Test your code against the expected output provided.
3. Check your program against the detailed marking criteria at the end of this document.

4. Submit the following to Brightspace as a group (*Only one copy is required per group, and any of the group members may submit*):
    - Github URL for your program code (invite your instructor to be a member of the project repository)
    - A copy of the test output (.txt file)

Peer Assessment (5%)

Each student must also complete a peer assessment of their group members. Your instructor will provide further submission details.

Program Details

Create a functional, event-driven program that manages the travel agency's data (contained in the provided data files) and allows a user to do the following:

- Find flights:
  - A travel agent can find a flight by providing the origin airport, the destination airport and the day of the week the flight departs.
- Make a reservation:
  - A travel agent can make a flight reservation for a traveller.
  - A reservation code is generated and assigned to the traveller's name and citizenship.
- Find reservations:
  - A travel agent can find existing flight reservations using the reservation code, and/or the airline and/or the traveller's name.
- Modify an existing reservation:
  - A travel agent can update the traveller's name and citizenship.
  - An existing flight reservation can be soft-deleted, marking it as inactive and freeing up a seat on the flight.

Detailed Requirements

When the graphical user interface is launched, the user can choose to either search flights and make a reservation, or search for and modify a reservation.

Find Flights

The findFlights method receives as its input arguments: the originating airport, the destination airport, and the day of week. The method returns a List of any matching *Flight* objects. If no matches are found, the list control is empty.

Make Reservation

When a travel agent selects a flight from the list, the text fields are populated with the selected flight code, airline, day, time and cost. The travel agent enters the traveller's full name and citizenship. The flight code, airline, day, time and cost cannot be edited. An error message is displayed if:

- A reservation is to be made but no flight is selected
- The name field is empty
- The citizenship field is empty

The makeReservation method receives as its input arguments: a Flight object, the travelers name and citizenship. An exception is thrown if the flight is completely booked, or the flight is null, or the name is empty/null, or the citizenship is empty/null. If there are no exceptions thrown a Reservation object is created, saved to the binary file and returned by the method.

Find Reservations

A travel agent can search for an existing reservation that contains the specified reservation code, and/or airline and/or traveller's full name. The list is populated with any reservations that are found. Each row in the list displays the code of the corresponding reservation record.

The findReservation method receives as its input arguments: reservation code, airline and/or traveler's full name. The method returns a list of matching *Reservation* objects. If no matches are found, the list control is empty. If the user doesn't enter any input, then all the reservations are displayed in the list.

Update Reservation

When a reservation in the list generated by the findReservation method is selected, the corresponding fields are populated, displaying the following information:

- Reservation code
- Flight code
- Airline name
- Cost

- Name
- Citizenship
- Status (active or inactive)

The only fields that can be edited are the name, citizenship and status. None of the other fields can be modified in any way by the user. After the travel agent has made changes to the reservation, they click the **Update** button. The mutator methods in the Reservation object are called and an error is displayed if an exception occurs.

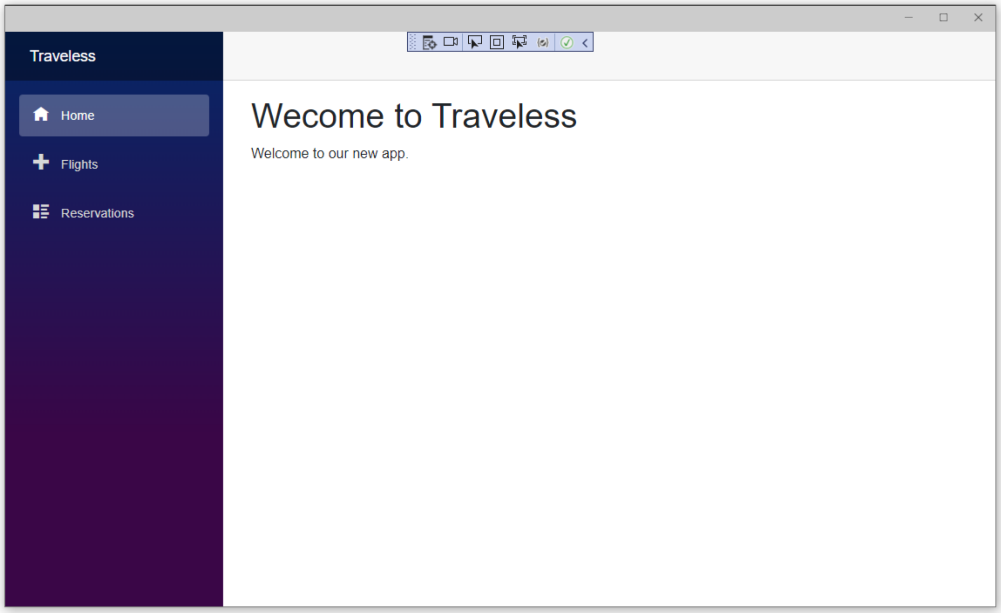
The persist method in the *ReservationManager* class saves all Reservation objects to a binary file on the hard drive.

Notes

- The *ReservationManager* class generates the reservation code.
- Each reservation is for one seat only.
- The name and citizenship do not need to follow any specific format; however, they cannot be empty.
- Each problem domain class overrides the *toString()* method.
- Flight codes use the following format: (L meaning letter, D meaning digit)
  - *LL-DDDD* (e.g.: *GA-1234*)
- Reservation codes use the following format: (L meaning Letter, D meaning Digit)
  - LDDDD (e.g., *I1234*)
- Times are in 24-hour format: HH:MM
- A reservation that is set to **inactive** is persisted and retained when the program opens again.

Expected Output

First page:


Traveless

Home

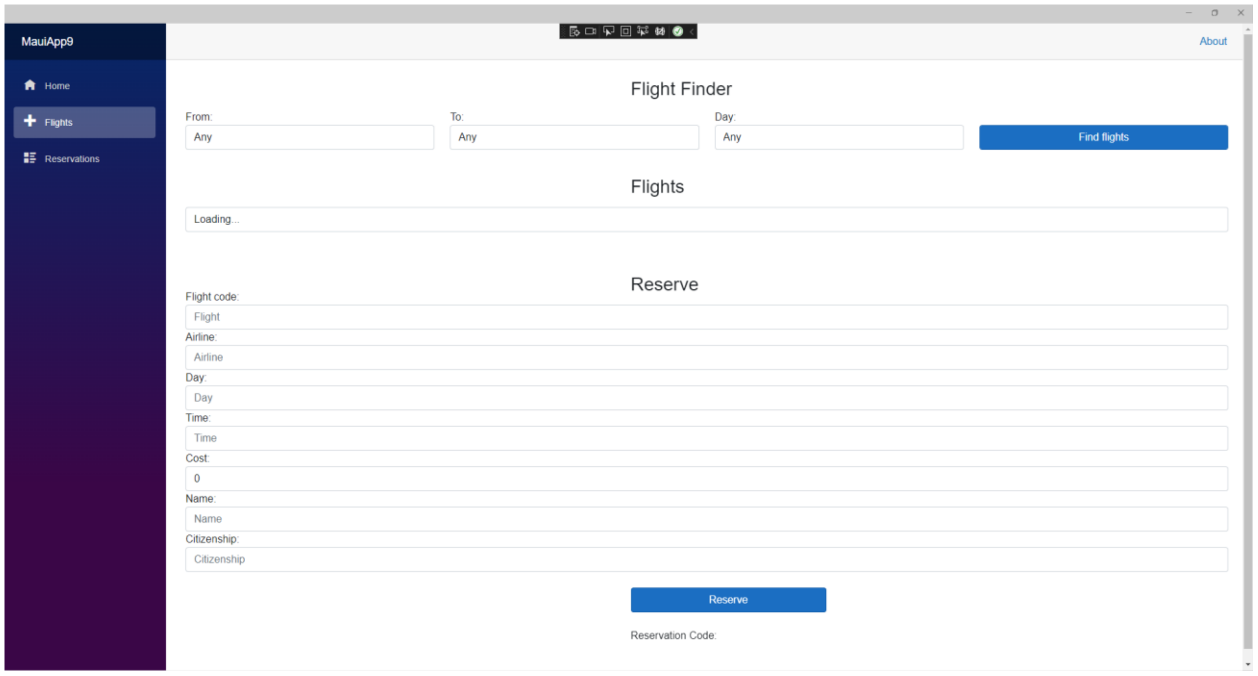
Flights

Reservations

Wecome to Traveless

Welcome to our new app.

Second page:


MauiApp9

Home

Flights

Reservations

About

Flight Finder

From: Any

To: Any

Day: Any

Find flights

Flights

Loading...

Reserve

Flight code: Flight

Airline: Airline

Day: Day

Time: Time

Cost: 0

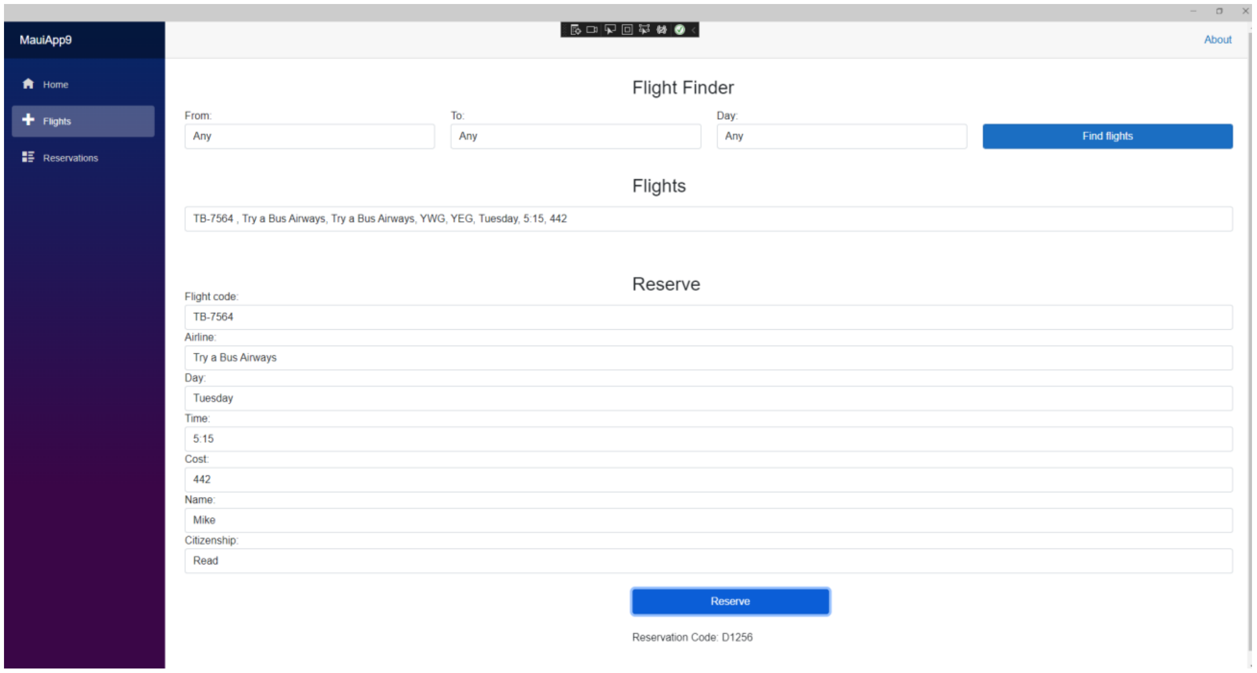
Name: Name

Citizenship: Citizenship

Reserve

Reservation Code:

After searching for the flights and selecting a flight:


MauiApp9

About

Home

Flights

Reservations

Flight Finder

From: Any

To: Any

Day: Any

Find flights

Flights

TB-7564 , Try a Bus Airways, Try a Bus Airways, YWG, YEG, Tuesday, 5:15, 442

Reserve

Flight code: TB-7564

Airline: Try a Bus Airways

Day: Tuesday

Time: 5:15

Cost: 442

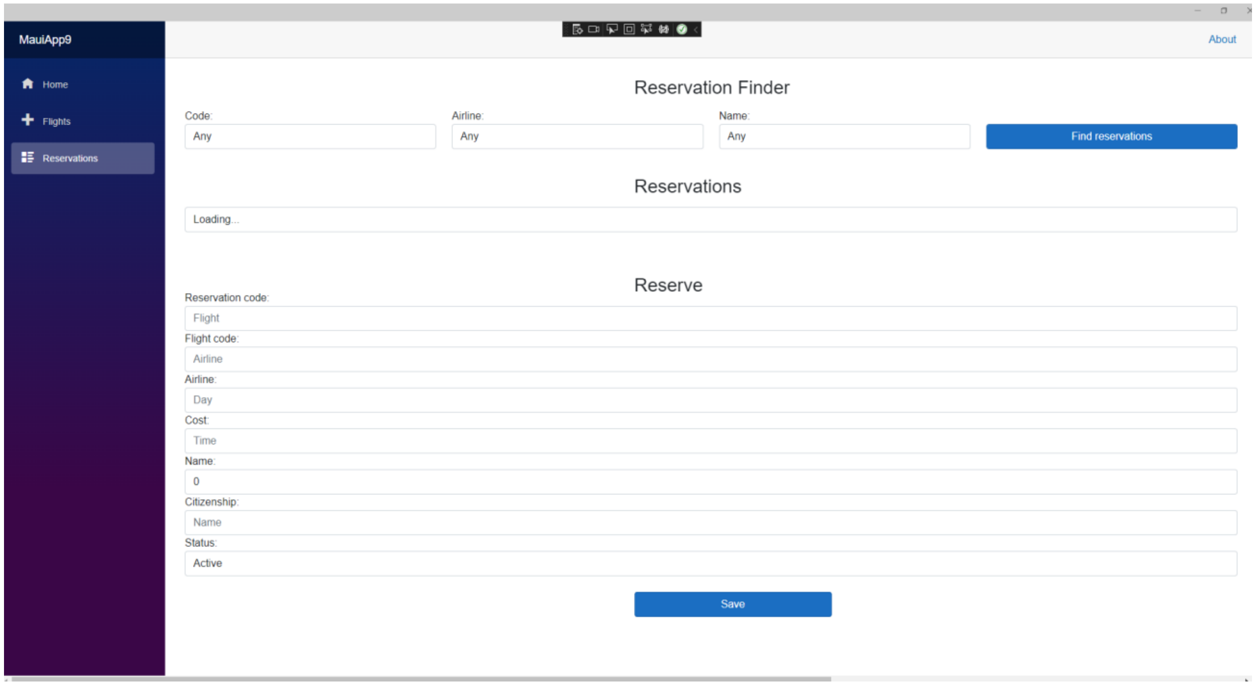
Name: Mike

Citizenship: Read

Reserve

Reservation Code: D1256

Third page:


MauiApp9

About

Home

Flights

Reservations

Reservation Finder

Code: Any

Airline: Any

Name: Any

Find reservations

Reservations

Loading...

Reserve

Reservation code: Flight

Flight code: Airline

Airline: Day

Cost: Time

Name: 0

Citizenship: Name

Status: Active

Save

Marking Criteria

Peer assessment/Attribution List Submission

	Not submitted (0%)	Submitted (100%)	Marks
Peer assessment	• Not submitted	• Completed for the group member	**/5**

Group Submission

	Needs Improvement (0–50%)	Good (51–75%)	Excellent (76–100%)	Marks
Working Code	• The project doesn't run in all scenarios • Syntax errors	• The project runs in all scenarios	• The project runs in all scenarios	**/10**
	• Input requests don't work	• Input requests work but don't match the scenario	• Input requests match the scenario exactly	**/10**
	• Output works but doesn't match the scenario	• Output works but doesn't match the scenario in all cases	• Output matches the scenario perfectly	**/10**
	• User interface design and functionality requires improvement	• User interface design and functionality works, but there is room for minor improvement	• User interface design and functionality works perfectly and is thoughtfully designed	**/15**

	Needs Improvement (0–50%)	Good (51–75%)	Excellent (76–100%)	Marks
	Many of the following items don't work: • Populate airports and flights • Find flights • Generate a reservation code • Make a reservation	Some of the following items don't work: • Populate airports and flights • Find flights • Generate a reservation code • Make a reservation	All of the following items work correctly: • Populate airports and flights • Find flights • Generate a reservation code • Make a reservation	**/10**
	Many of the following items don't work: • Find reservations • Store reservations in a random-access file • Modify and cancel a reservation • Populate lists and allows selection	Some of the following items don't work: • Find reservations • Store reservations in a random-access file • Modify and cancel a reservation • Populate lists and allows selection	All of the following items work correctly: • Find reservations • Store reservations in a random-access file • Modify and cancel a reservation • Populate lists and allows selection	**/10**
	Many of the following items don't work: • Handle Invalid flight code exception • Handle invalid citizenship exception • Handle no more seats exception • Applicable fields not editable	Some of the following items don't work: • Handle Invalid flight code exception • Handle invalid citizenship exception • Handle no more seats exception • Applicable fields not editable	All of the following items work correctly: • Handle Invalid flight code exception • Handle invalid citizenship exception • Handle no more seats exception • Applicable fields not editable	**/10**
Style	• Indentation – not consistent • Readability – poor variable names • Documentation	• Indentation – mostly consistent • Readability – some variable names are not ideal • Documentation	• Indentation – consistent • Readability – good variable names • Documentation	**/10**

	Needs Improvement (0–50%)	Good (51–75%)	Excellent (76–100%)	Marks
	○ No comments are included at the top. ○ No comments indicating major code sections or what they do	○ Comments at the top are missing or incomplete. ○ Comments indicating major code sections and what they do are incomplete	○ Comments at the top are complete and include name, date, program description including details on inputs, processing and outputs (4–5 sentences minimum). ○ Comments indicate major code sections and what they do	
Testing	• Sample output doesn't match the provided expected output • Output is not formatted according to the specification (expected output)	• Parts of the sample output don't exactly match the expected output • Output formatted according to the specification (expected output)	• Sample output exactly matches the provided expected output • Output formatted according to the specification (expected output)	**/10**
			Total	**/100**