

Final Project Design

Airline Search

Jeremy Johnson

INEW 2332

Summary

This application is a flight reservation system that allows the user to search for and reserve flights. The flights can be one way or round trip. It provides a registration process and requires a log in to reserve flights. The user can search for flights based on airport codes, departure date and return date. It returns a list of flights based on the search criteria. The user can then reserve the desired flight or flights. Finally, it provides the option to show the users reserved flights.

Requirements

This application will be composed of 6 GUI screens.

Buttons will redirect user to the appropriate screen.

1. Home Screen
 - a. This screen functions as the home screen for the app
 - b. Buttons
 - i. Register
 - ii. Log In
 - iii. One Way Search
 - iv. Round Trip Search
 - v. Show Reservations
 - vi. Update User
2. Register Screen
 - a. This screen allows the user to register and have their data written to the users table of the database.
 - b. Buttons
 - i. Home
 - ii. Register
 - iii. Login
 - iv. Update User
 - c. Input Fields
 - i. Username
 - ii. Password
 - iii. Verify Password
 - iv. First Name
 - v. Last Name
3. Log In / Log Out Screen
 - a. This allows the user to log into or out of their account
 - b. Buttons
 - i. Log In
 - ii. Log Out
 - iii. Home
 - iv. Register
 - c. Input Fields
 - i. Username

- ii. Password
- 4. Search One Way
 - a. This screen allows the user to search for available one way flights
 - b. Currently only searches for direct flights
 - c. Requires the use of airport codes
 - d. Flights will be searched based on
 - i. Departure Code
 - ii. Destination Code
 - iii. Departure Date
 - iv. Seats Requested
 - e. Buttons
 - i. Home
 - ii. Log In
 - iii. Search Flights
 - iv. Clear
 - f. Input Fields
 - i. Departure Airport Code
 - ii. Destination Airport Code
 - iii. Date
 - iv. Seats Requested
 - g. The Search Flights Button will open the Available Flights screen
- 5. Available Flights
 - a. This screen shows the available flights based on the users search criteria
 - b. The user can highlight the desired flight and reserve it
 - i. Reserving a flight will decrease the available seats in the flights table by the number of requested seats and write the data to the reservations table in the database.
 - c. Buttons
 - i. Home
 - ii. One Way Search
 - iii. Round Trip Search
 - iv. Reserve Flight
 - d. Output Fields
 - i. Airline
 - ii. Flight Number
 - iii. Departure Airport Code
 - iv. Destination Airport Code
 - v. Departure Date
 - vi. Departure Time
 - vii. Cost
 - viii. Number of Seats
 - ix. Number of Available Seats
- 6. Show Reservations
 - a. This screen shows the users reserved flights
 - b. Buttons

- i. Home
 - ii. Log In
 - iii. Cancel Reservation
 - iv. One Way Search
 - v. Round Trip Search
 - c. Output Fields
 - i. First Name
 - ii. Last Name
 - iii. Airline
 - iv. Flight Number
 - v. Departure Airport Code
 - vi. Destination Airport Code
 - vii. Departure Date
 - viii. Departure Time
 - ix. Number of Reserved Seats
 - d. Cancel Button
 - i. Cancel button deletes the highlighted reservation from the reservations table and increases the available seats in the flights table by the number of seats currently reserved
7. Update User
- a. This frame allows the user to update user information or delete their account
 - b. Buttons
 - i. Home
 - ii. Login
 - iii. Update User
 - iv. Delete User
 - c. Input Fields
 - i. Password
 - ii. Verify Password
 - iii. First Name
 - iv. Last Name
 - d. Update User button updates the user information with the new entries
 - e. Delete User button deletes the logged in user
8. Search Round Trip
- a. This screen allows the user to search for available round trip flights
 - b. Currently only searches for direct flights
 - c. Requires the use of airport codes
 - d. Flights will be searched based on
 - i. Departure Code
 - ii. Destination Code
 - iii. Departure Date
 - iv. Seats Requested
 - e. Buttons
 - i. Home
 - ii. Log In
 - iii. Search Flights

- iv. Clear
 - f. Input Fields
 - i. Departure Airport Code
 - ii. Destination Airport Code
 - iii. Departure Date
 - iv. Return Date
 - v. Seats Requested
 - g. The Search Flights Button will open the Available Flights screen
9. Available Flights Return Flights
- a. This screen shows the available return flights for round trip flight search
 - b. The user can highlight the desired flight and reserve it
 - i. Reserving a flight will decrease the available seats in the flights table by the number of requested seats and write the data to the reservations table in the database.
 - c. Buttons
 - i. Home
 - ii. One Way Search
 - iii. Round Trip Search
 - iv. Reserve Flight
 - d. Output Fields
 - i. Airline
 - ii. Flight Number
 - iii. Departure Airport Code
 - iv. Destination Airport Code
 - v. Departure Date
 - vi. Departure Time
 - vii. Cost
 - viii. Number of Seats
 - ix. Number of Available Seats

Application Decisions

Programming Language

- Python

Testing

- Pytest

GUI Library

- Tkinter

Database

- PostgreSQL
 - Production_db
 - Users Table
 - Flights Table
 - Reservations Table
 - test_db

- Users Table
- Flights Table
- Reservations Table

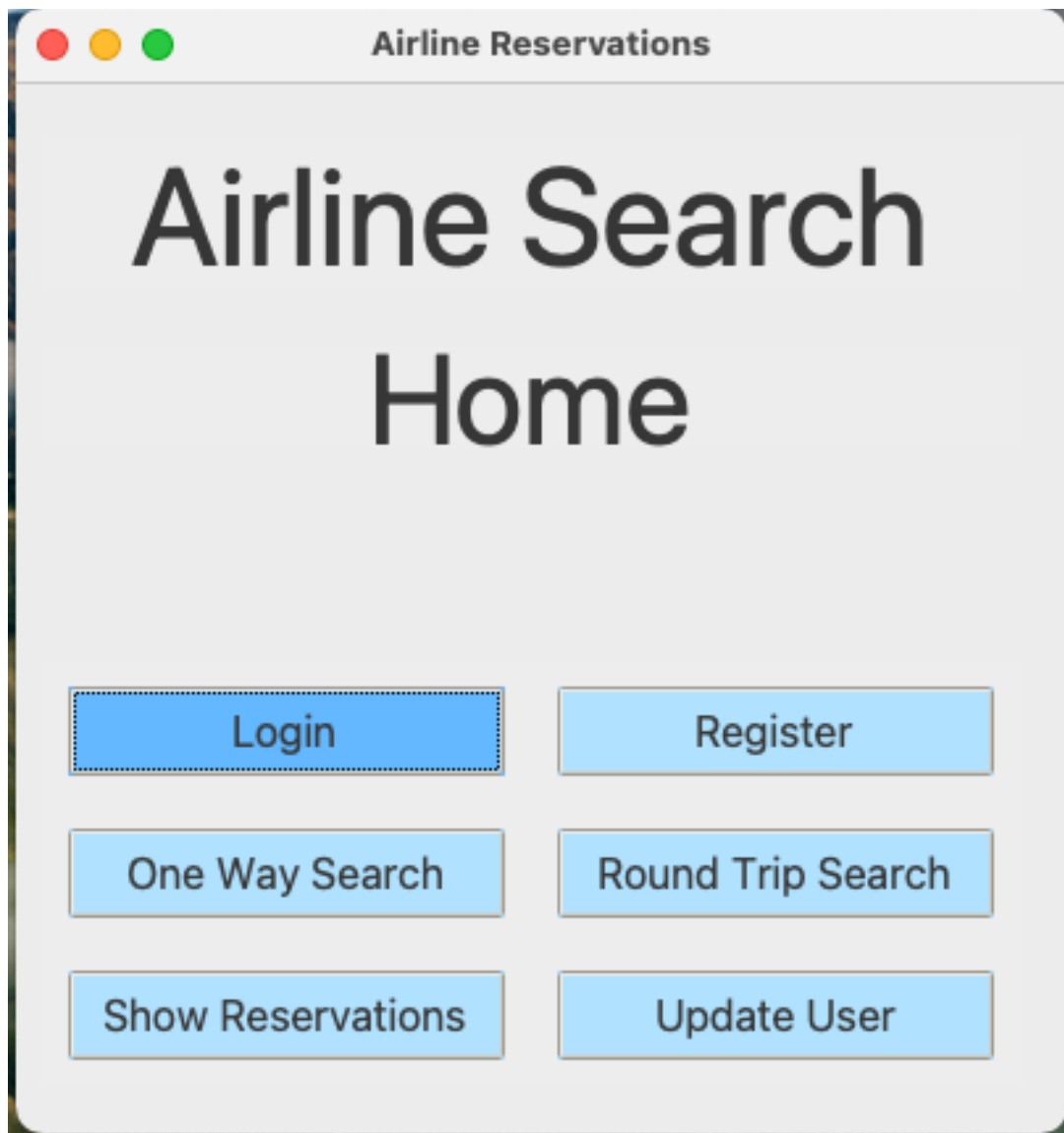
Future Features

Features to be added to project time permitting:

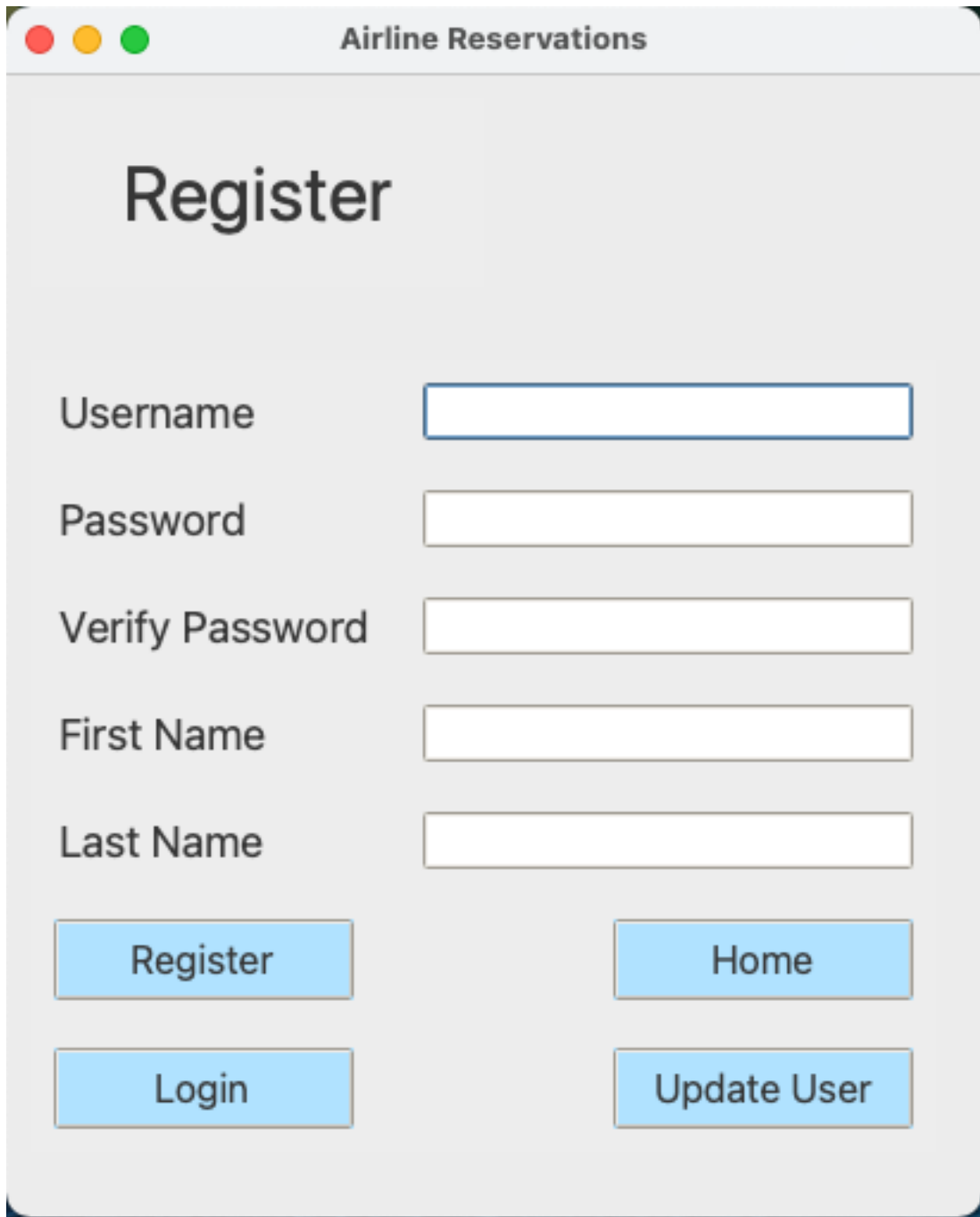
1. Implement feature to search for flights with multiple stops
2. Implement feature to show the city and state based on airport codes
 - a. Use it to search for flights based on city and state
 - b. Use it to display city and state in the reservation summary screen
3. Add additional User information and transfer all user data to reservation, i.e., address
4. Search reservations function that allows user to search current reservations based on date, departure code or destination code.
5. Impose password restrictions. i.e., length, special characters
6. Multi select on reservations to cancel reservations

Flow Charts, Diagrams, Database Fields, Tests

Home Frame



Register Frame



A screenshot of a web application window titled "Airline Reservations". The window has a light gray background and a title bar with three colored buttons (red, yellow, green) on the left. The main content area is titled "Register" in a large, bold, black font. Below the title, there are five input fields arranged vertically, each with a label to its left: "Username", "Password", "Verify Password", "First Name", and "Last Name". Each input field is a white rectangle with a thin blue border. Below the input fields, there are four buttons arranged in a 2x2 grid. The buttons are light blue with a thin blue border. The top-left button is labeled "Register", the top-right button is labeled "Home", the bottom-left button is labeled "Login", and the bottom-right button is labeled "Update User".

Airline Reservations

Register

Username

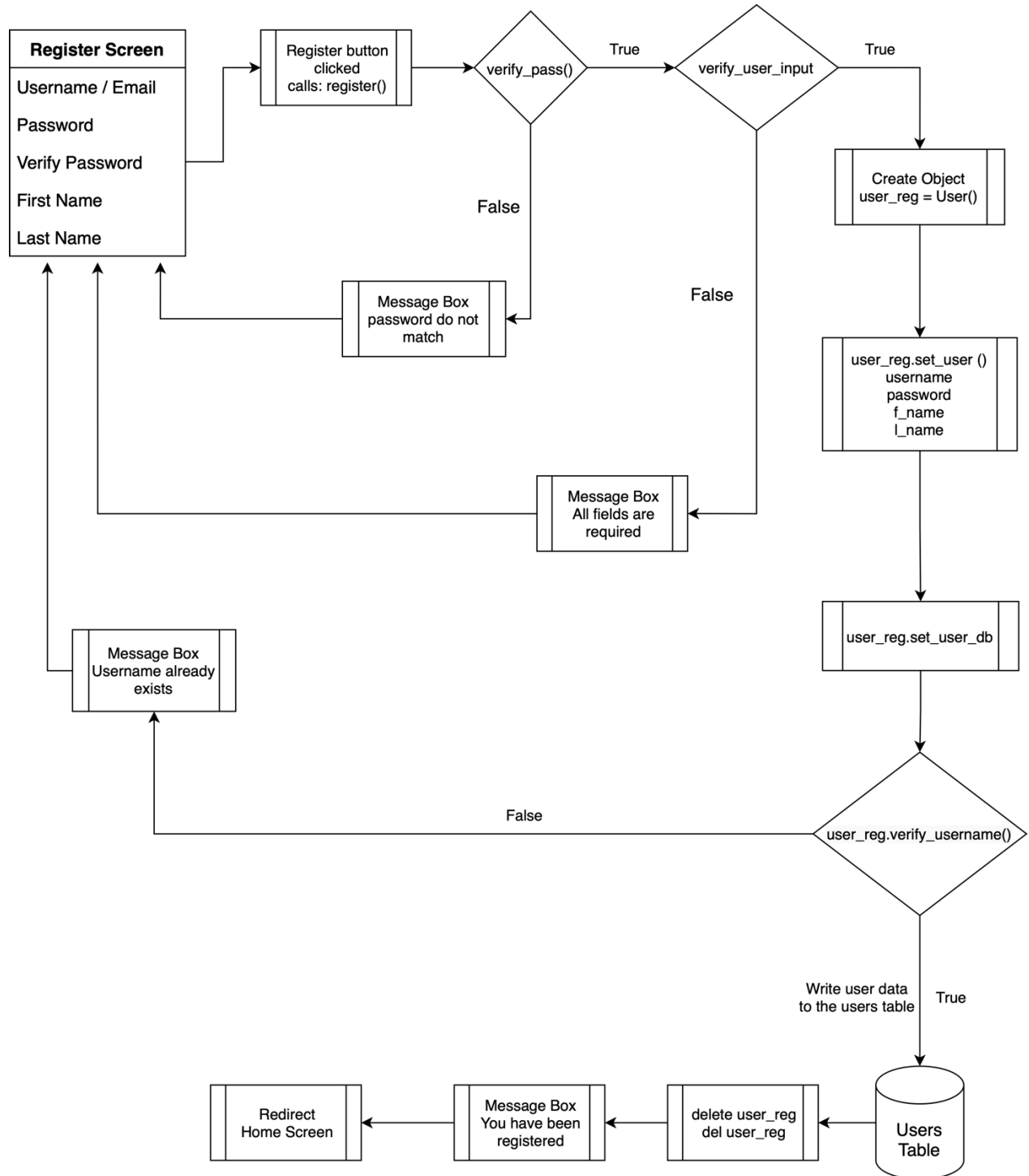
Password

Verify Password

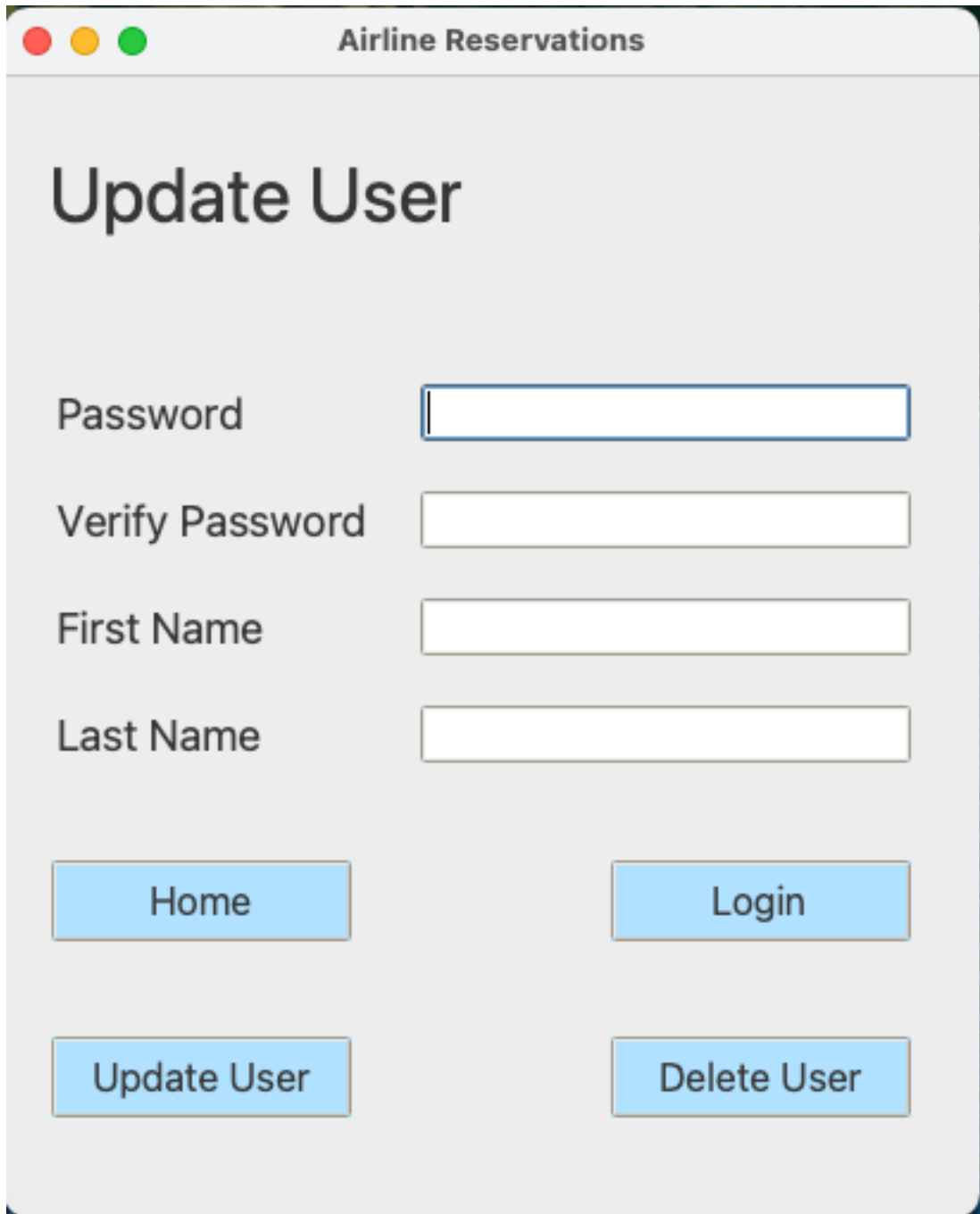
First Name

Last Name

Register Frame Flow Chart



Update User Frame



A screenshot of a web application window titled "Airline Reservations". The window has a light gray background and a title bar with three colored buttons (red, yellow, green) on the left. The main content area is titled "Update User" in a large, bold, dark gray font. Below the title, there are four input fields arranged vertically, each with a label to its left: "Password", "Verify Password", "First Name", and "Last Name". The "Password" field has a blue border, while the others have a brown border. At the bottom of the form, there are four light blue buttons with black text, arranged in a 2x2 grid: "Home", "Login", "Update User", and "Delete User".

Airline Reservations

Update User

Password

Verify Password

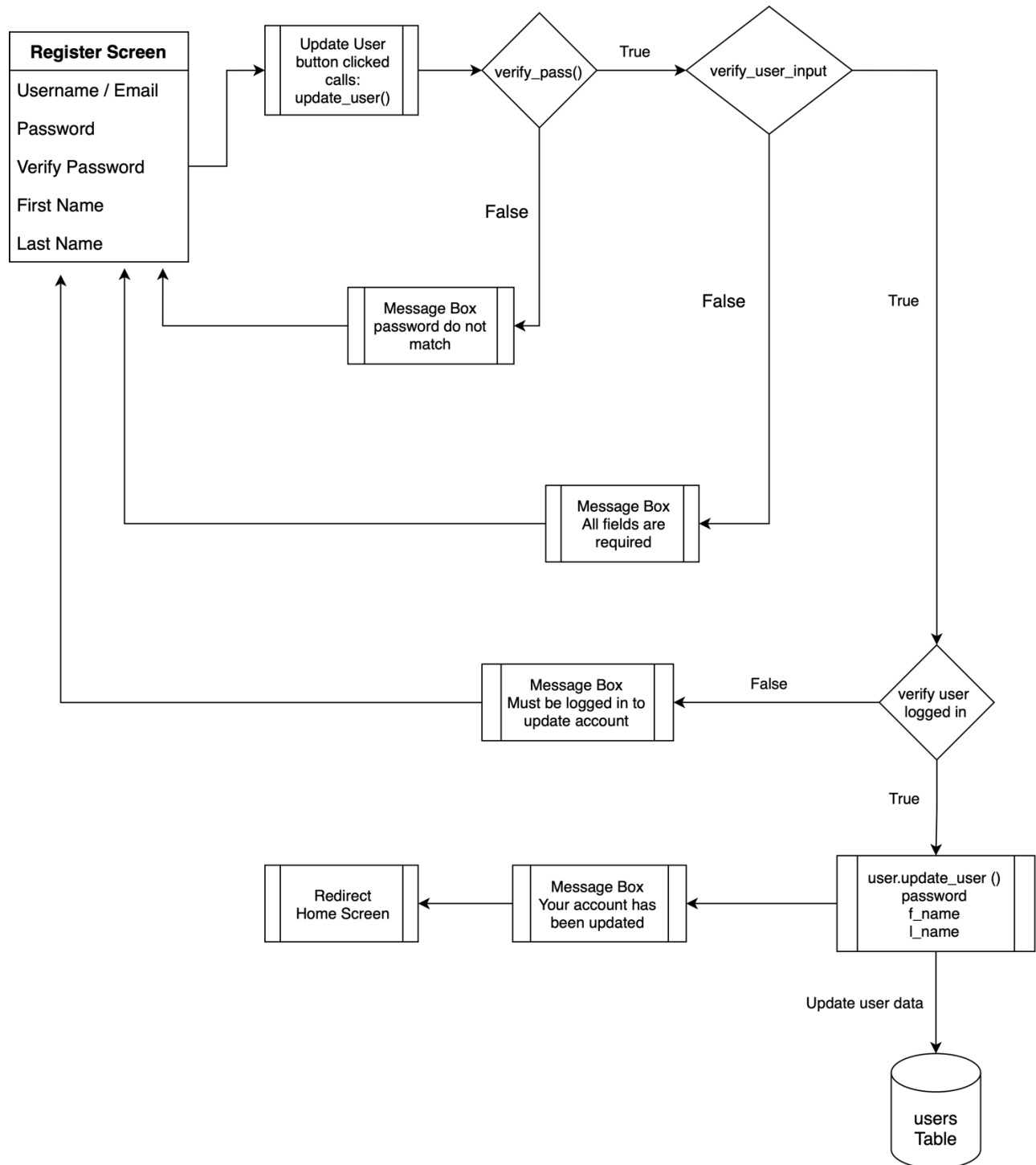
First Name

Last Name

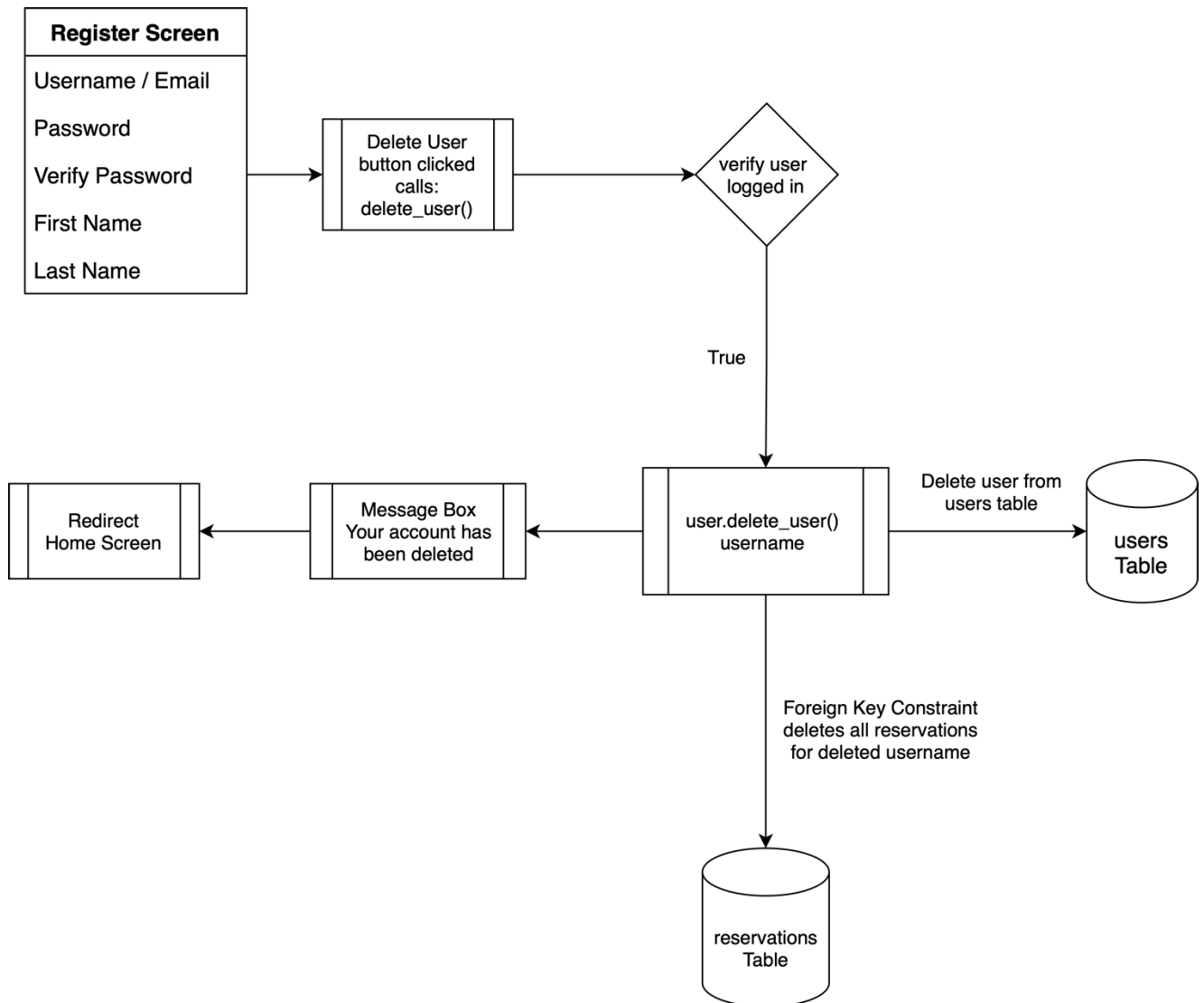
Home Login

Update User Delete User

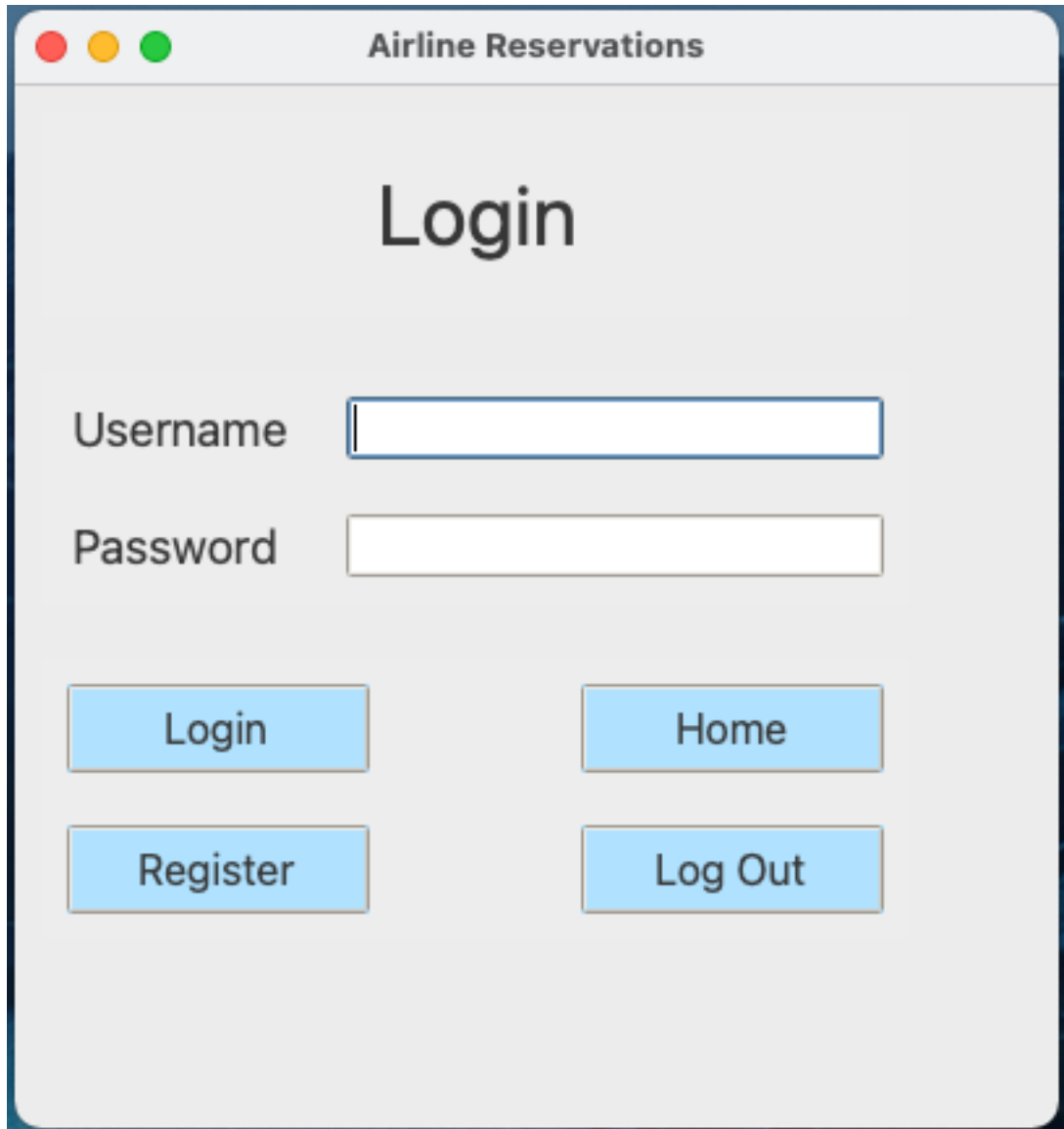
Update User Frame Flowchart



Delete User Flowchart



Login Frame



A screenshot of a web application window titled "Airline Reservations". The window has a light gray background and a dark blue border. At the top, there are three colored circles (red, yellow, green) and the title "Airline Reservations". The main content area is titled "Login" in a large, bold, black font. Below the title, there are two input fields: "Username" and "Password". The "Username" field is a white rectangle with a blue border, and the "Password" field is a white rectangle with a gray border. Below the input fields, there are four buttons arranged in a 2x2 grid. The buttons are light blue with black text and a thin gray border. The buttons are labeled "Login", "Home", "Register", and "Log Out".

Airline Reservations

Login

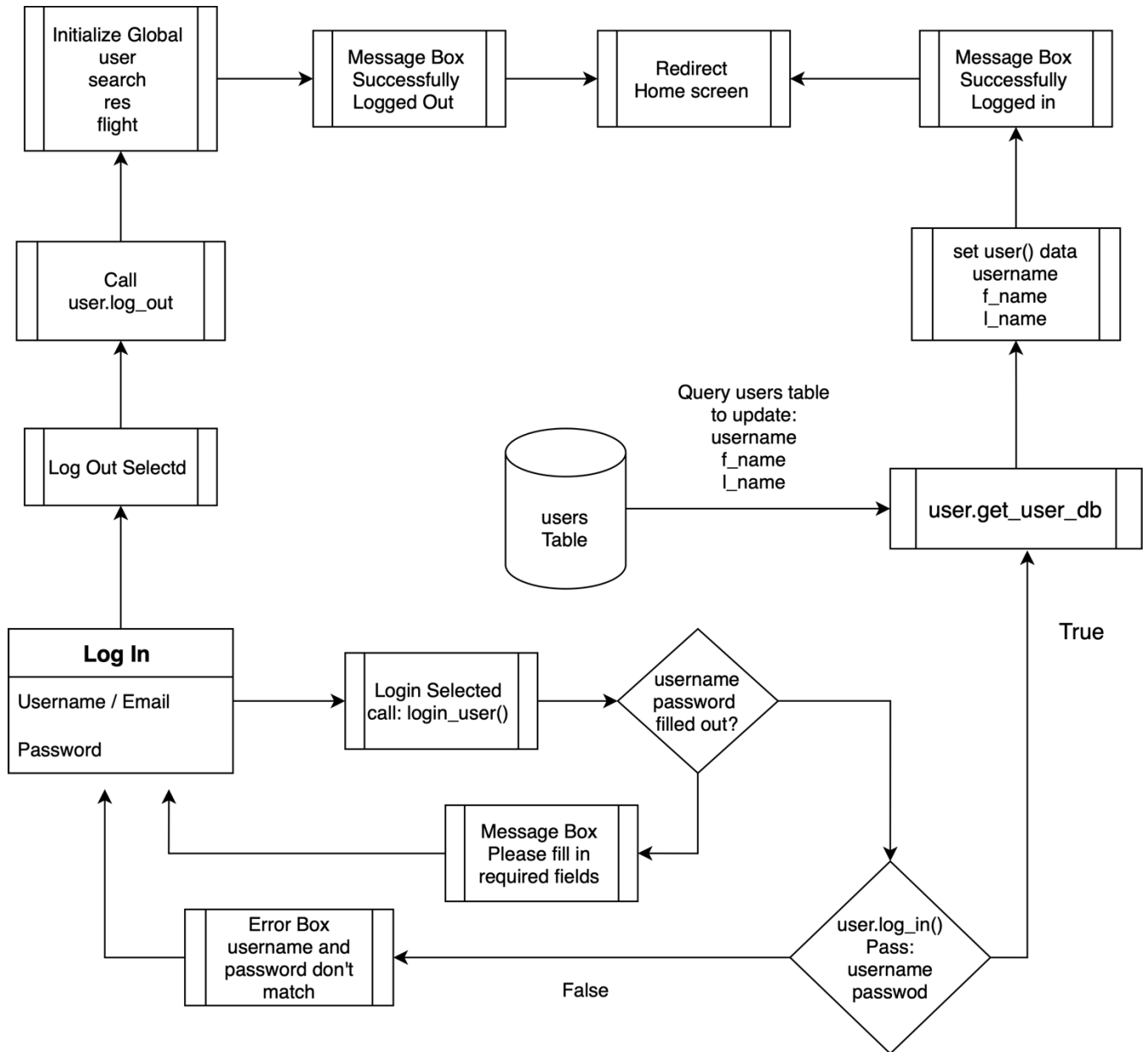
Username

Password

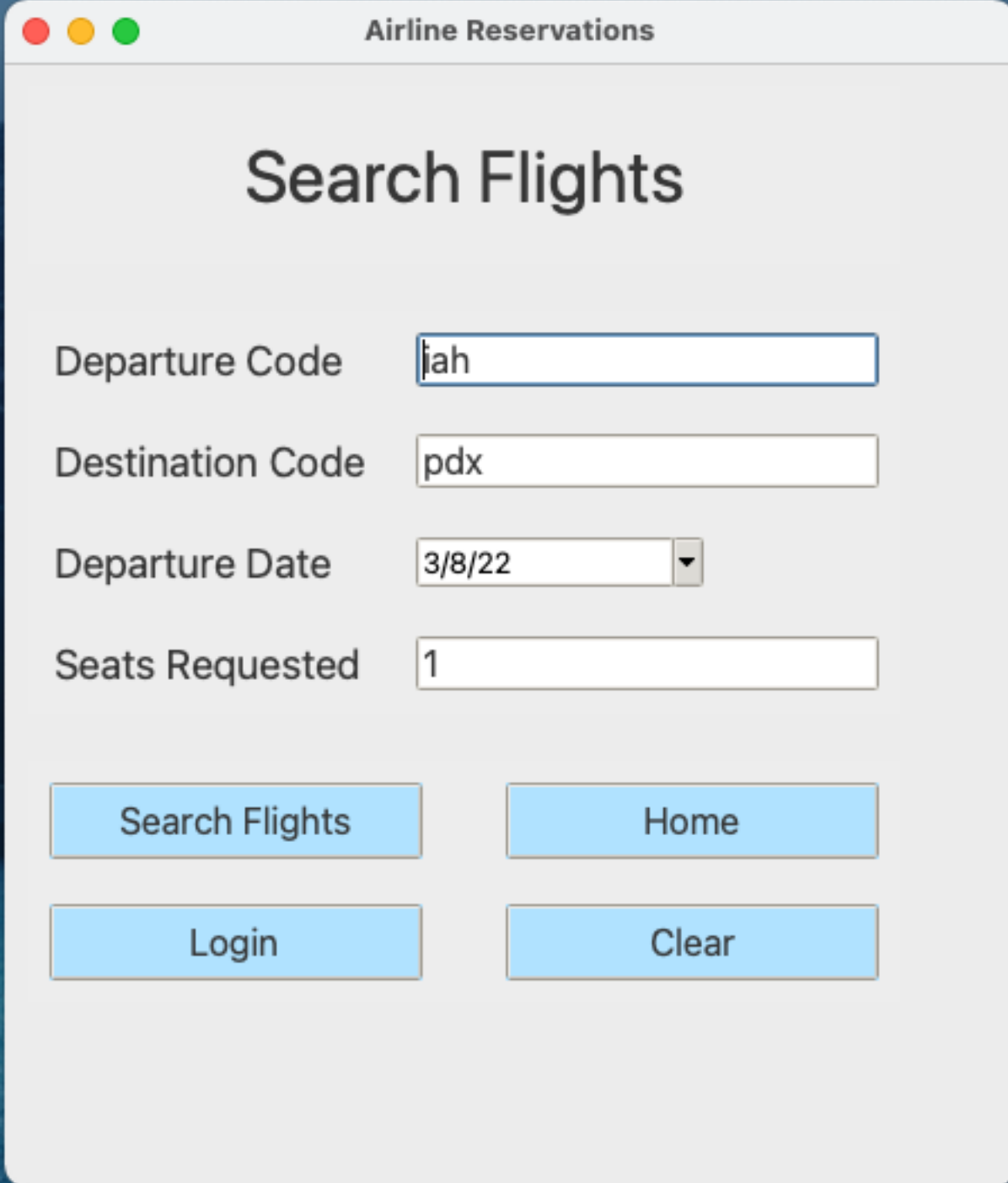
Login Home

Register Log Out

Login Frame Flowchart



Search One Way Flights Frame



A screenshot of a web application window titled "Airline Reservations". The window has a light gray background and a title bar with three colored buttons (red, yellow, green) on the left. The main heading "Search Flights" is centered at the top in a large, bold, dark gray font. Below the heading, there are four input fields arranged vertically. The first field is labeled "Departure Code" and contains the text "lah". The second field is labeled "Destination Code" and contains the text "pdx". The third field is labeled "Departure Date" and contains the text "3/8/22" with a small downward arrow icon to its right. The fourth field is labeled "Seats Requested" and contains the text "1". At the bottom of the form, there are four light blue buttons with black text, arranged in a 2x2 grid. The buttons are labeled "Search Flights", "Home", "Login", and "Clear".

Airline Reservations

Search Flights

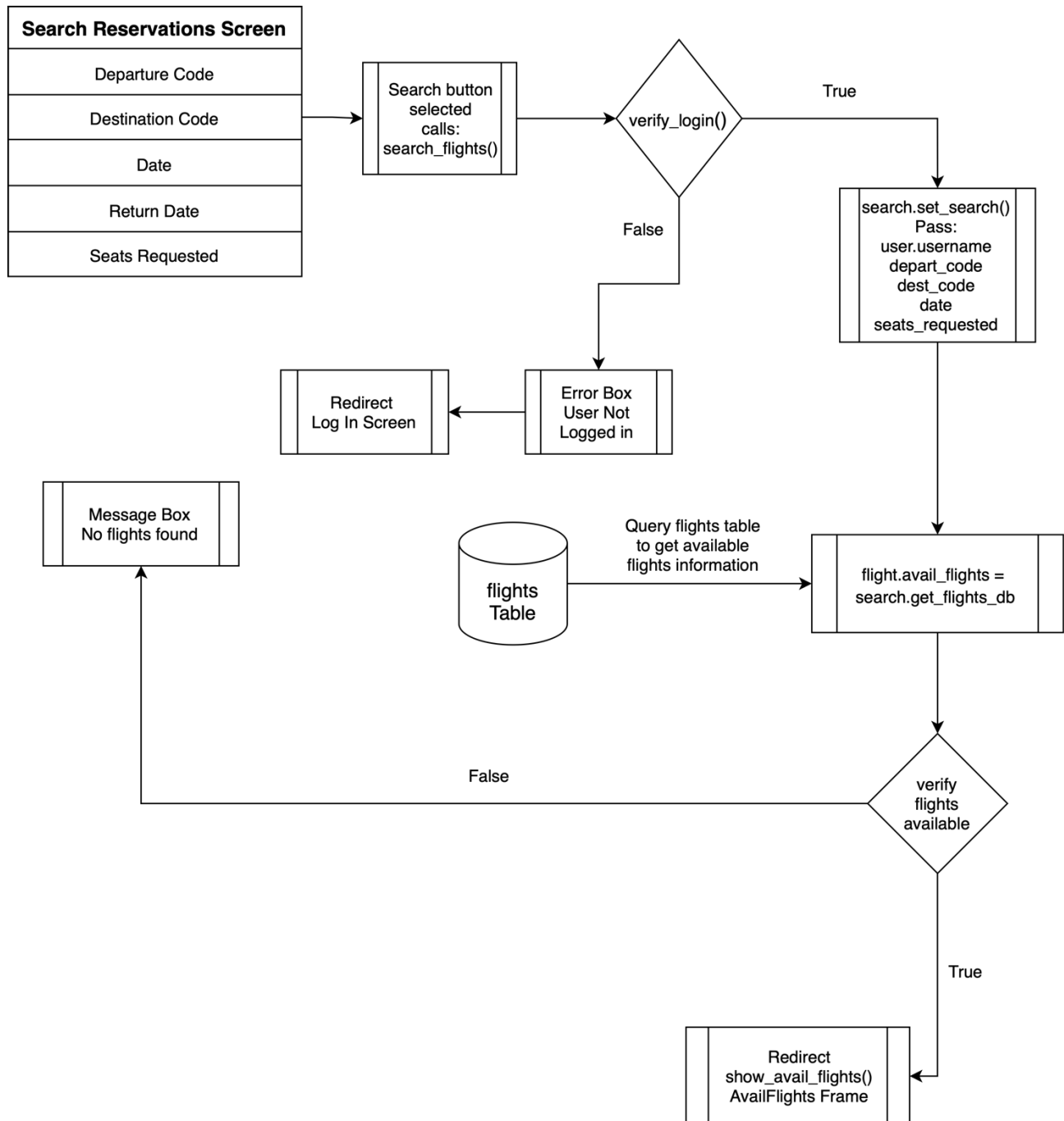
Departure Code

Destination Code

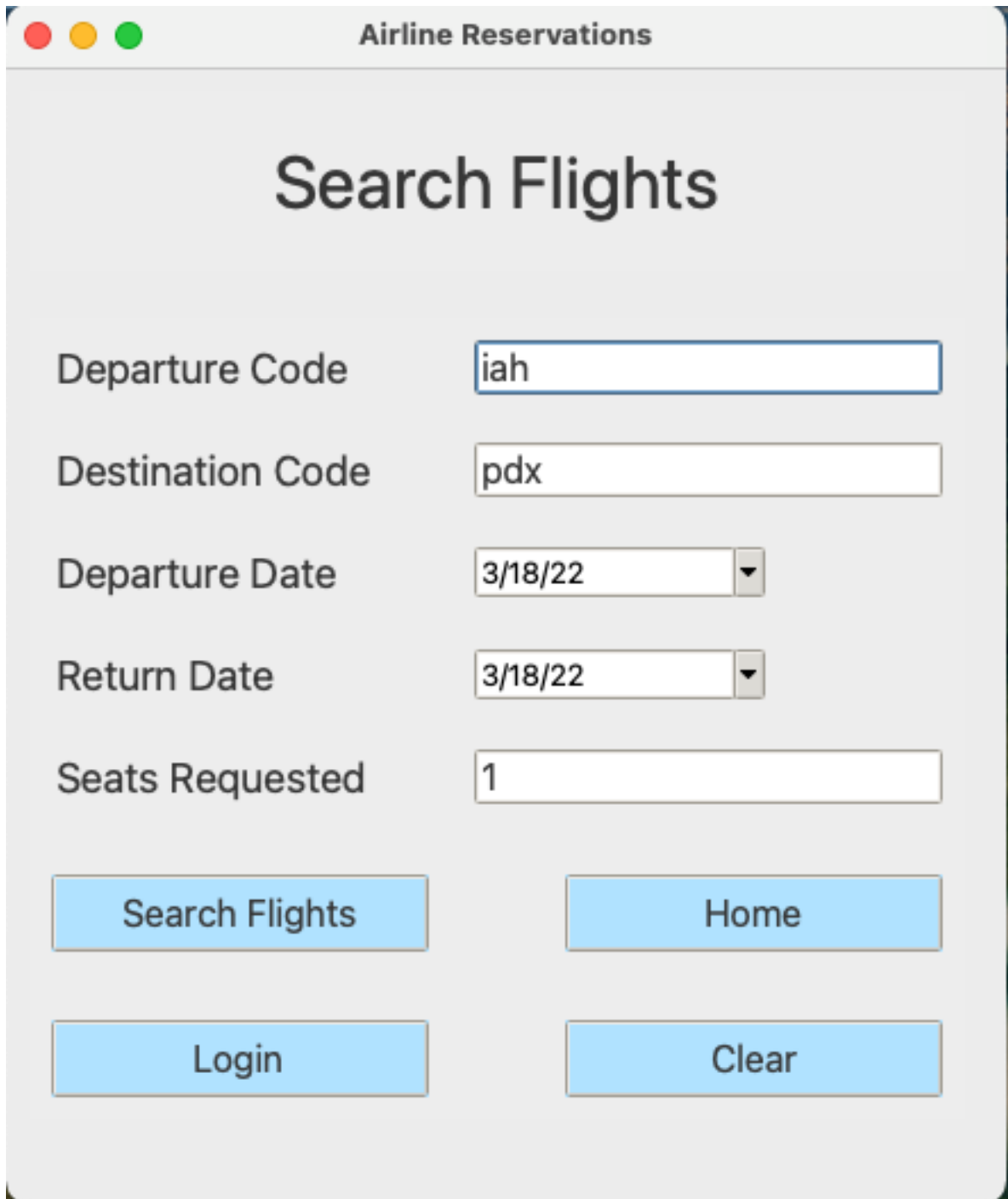
Departure Date ▼

Seats Requested

Search One Way Flights Frame Flowchart



Search Round Trip Flights Frame



A screenshot of a web application window titled "Airline Reservations". The window has a light gray background and rounded corners. At the top, there are three colored window control buttons (red, yellow, green) on the left. The main heading "Search Flights" is centered in a large, bold, dark gray font. Below the heading, there are five input fields arranged vertically. Each field has a label to its left and a text input box to its right. The first field is "Departure Code" with the value "iah". The second is "Destination Code" with the value "pdx". The third is "Departure Date" with the value "3/18/22" and a small downward arrow icon. The fourth is "Return Date" with the value "3/18/22" and a small downward arrow icon. The fifth is "Seats Requested" with the value "1". At the bottom of the form, there are four light blue buttons with black text, arranged in a 2x2 grid. The top row contains "Search Flights" and "Home". The bottom row contains "Login" and "Clear".

Airline Reservations

Search Flights

Departure Code

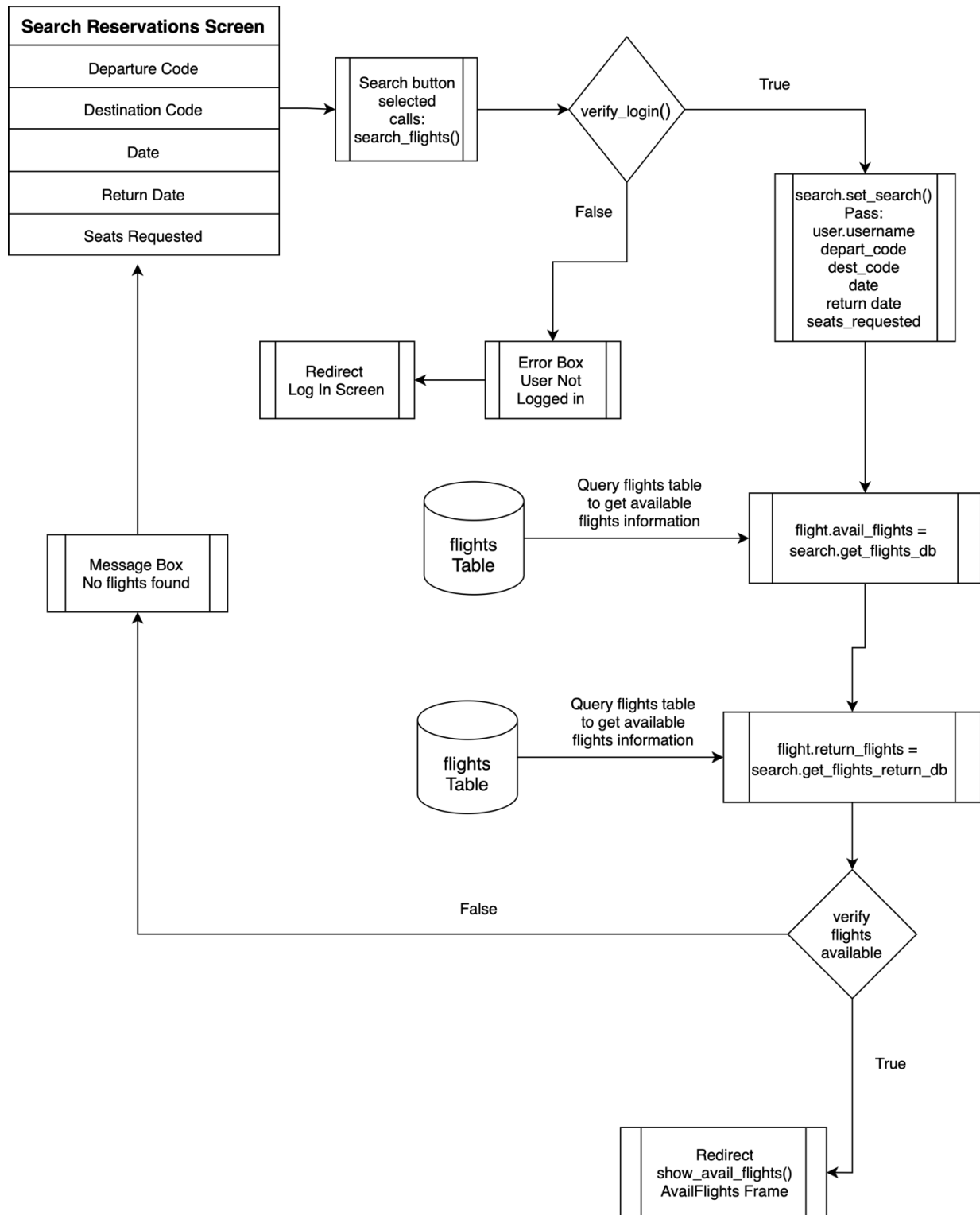
Destination Code

Departure Date

Return Date

Seats Requested

Search Round Trip Flights Frame Flowchart



Available Flights Frame

Airline Reservations

Available Flights

Airline	Flight Number	Departure Code	Destination Code	Departure Date	Departure Time	Cost	Available Seats
American Airlines	AA5678	IAH	SFO	02/16/2022	22:00	\$300.00	62
United Airlines	UA5678	IAH	SFO	02/16/2022	12:00	\$300.00	12

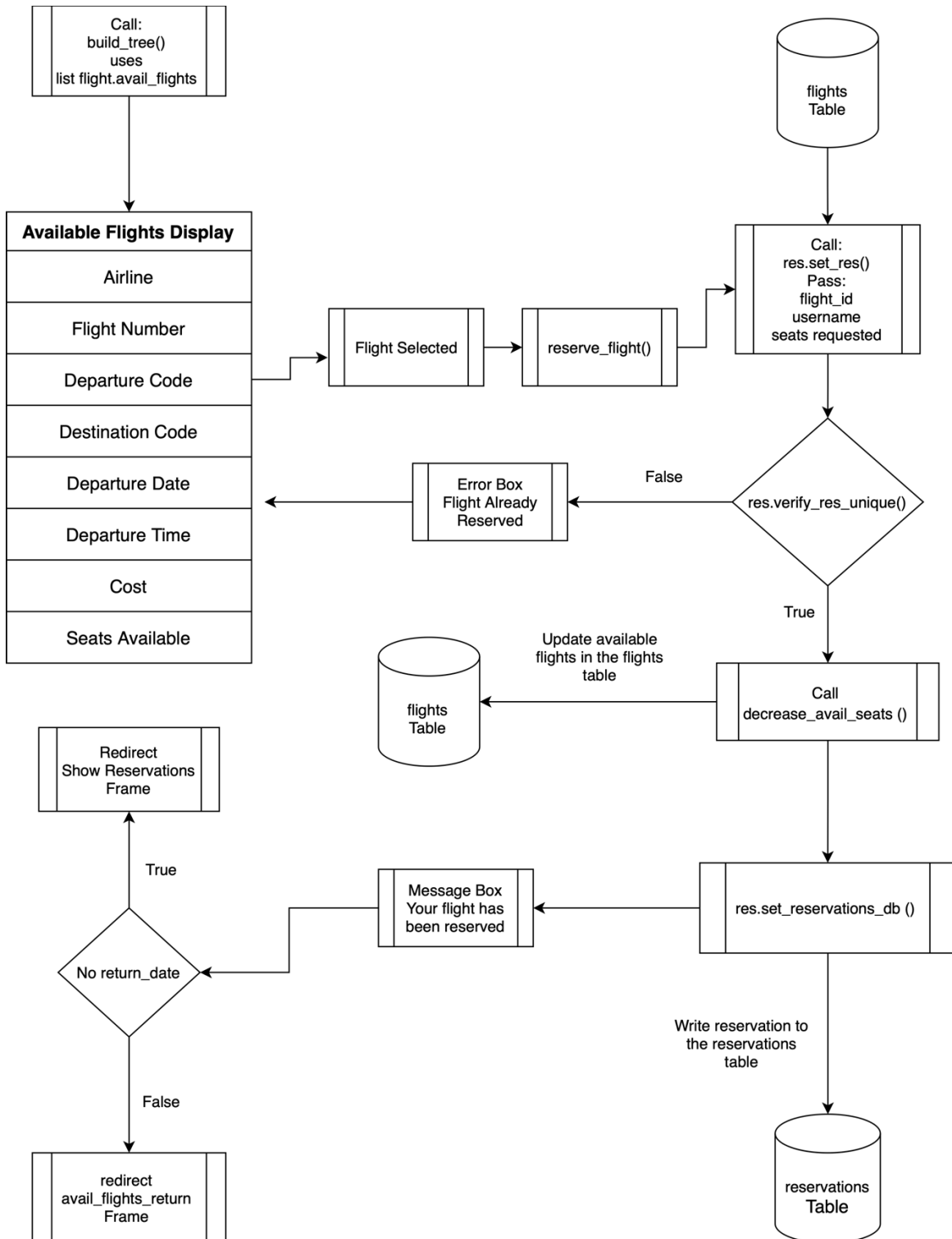
Home

Reserve Flight

One Way Search

Round Trip Search

Available Flights Frame Flowchart



Available Flights Return Flight Frame

Airline Reservations

Available Flights - Return Flight

Airline	Flight Number	Departure Code	Destination Code	Departure Date	Departure Time	Cost	Available Seats
American Airlines	AA2345	SFO	IAH	02/22/2022	14:00	\$300.00	96
United Airlines	UA2345	SFO	IAH	02/22/2022	14:00	\$300.00	86

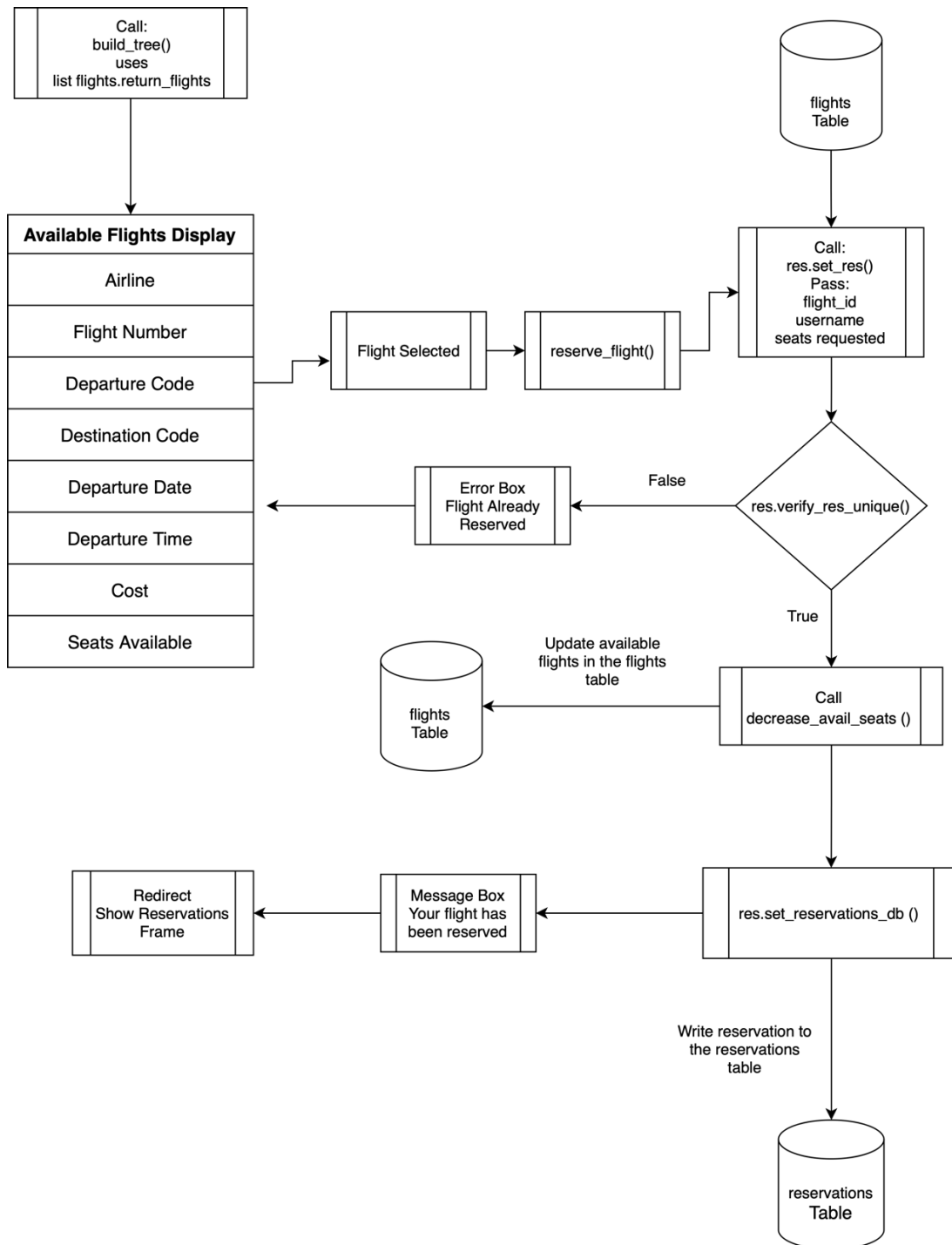
Home

Reserve Flight

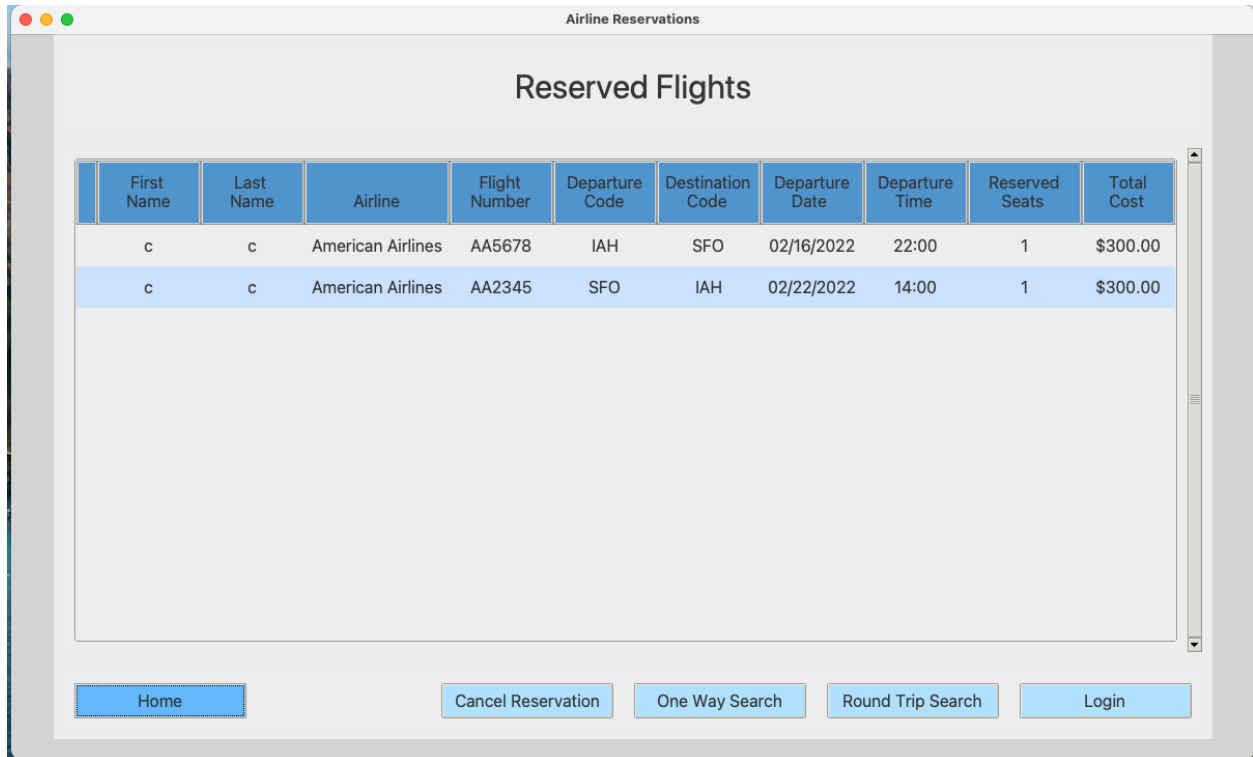
One Way Search

Round Trip Search

Available Flights Return Flight Frame Flowchart



Show Reservations Frame

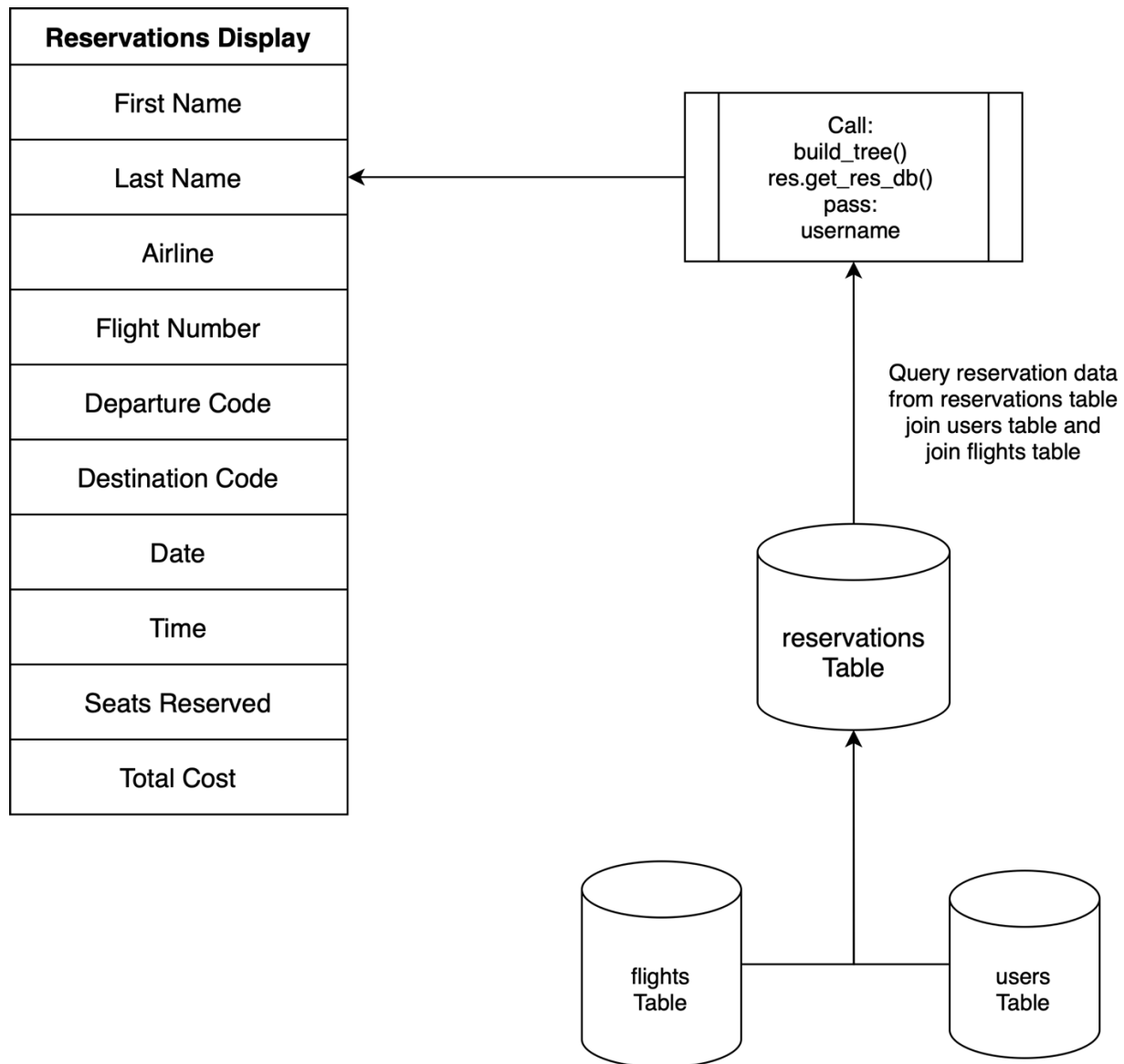


The screenshot shows a web application window titled "Airline Reservations". Inside the window, there is a section titled "Reserved Flights" which contains a table. The table has 10 columns: First Name, Last Name, Airline, Flight Number, Departure Code, Destination Code, Departure Date, Departure Time, Reserved Seats, and Total Cost. There are two rows of data in the table. Below the table, there is a navigation bar with five buttons: Home, Cancel Reservation, One Way Search, Round Trip Search, and Login.

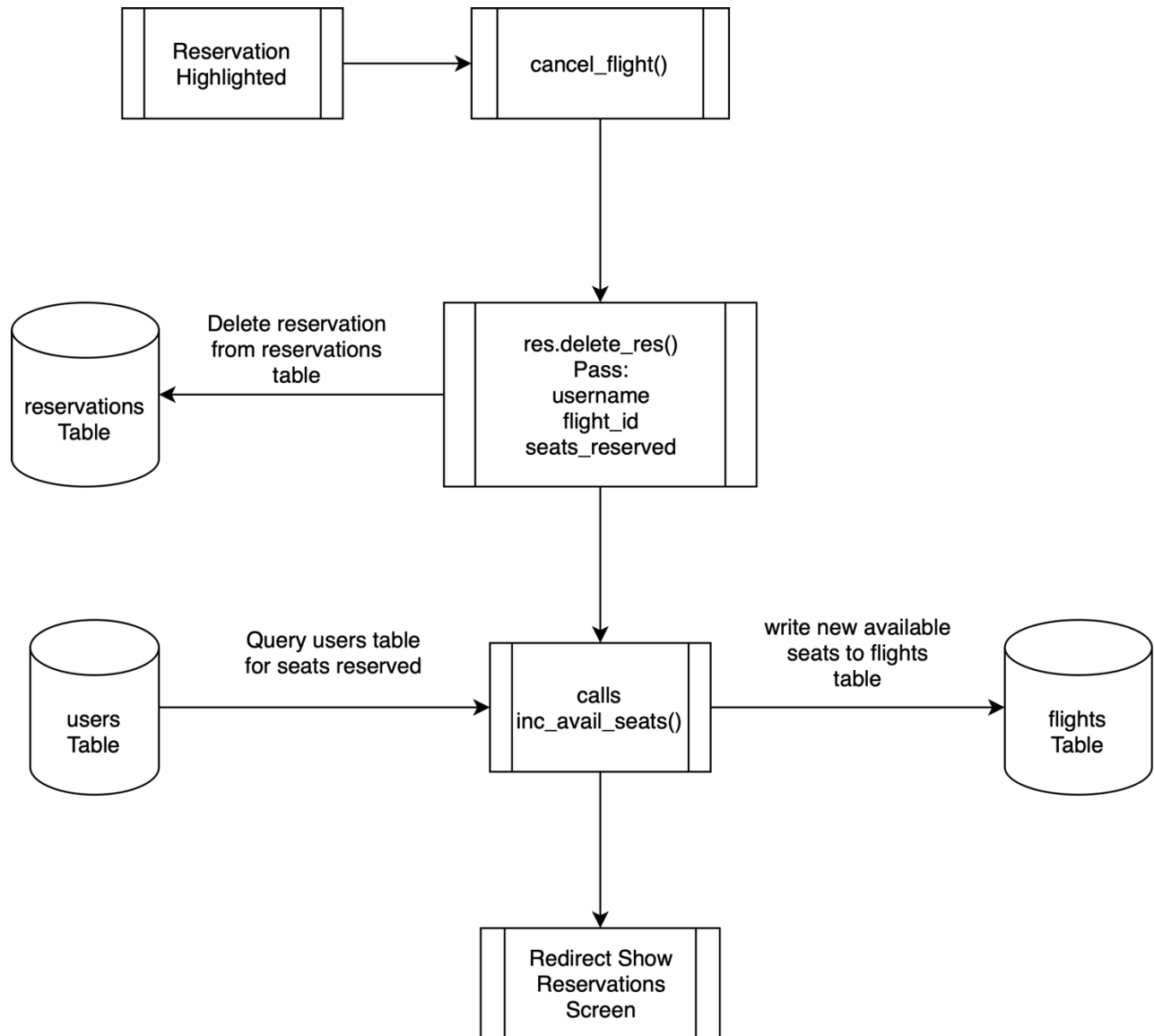
First Name	Last Name	Airline	Flight Number	Departure Code	Destination Code	Departure Date	Departure Time	Reserved Seats	Total Cost
c	c	American Airlines	AA5678	IAH	SFO	02/16/2022	22:00	1	\$300.00
c	c	American Airlines	AA2345	SFO	IAH	02/22/2022	14:00	1	\$300.00

Home Cancel Reservation One Way Search Round Trip Search Login

Show Reservations Frame Flowchart



Cancel Reservation Flowchart



Database Fields

Flights Table

Table Fields	Datatype
flight_id	Serial Primary Key
Airline	Varchar – Not Null
Flight_number	Varchar – Not Null
Depart_code	Varchar – Not Null
Dest_code	Varchar – Not Null
Depart_Date	Date – Not Null
Depart_Time	Time – Not Null
Cost	int – Not Null
Avail_seats	Int – Not Null

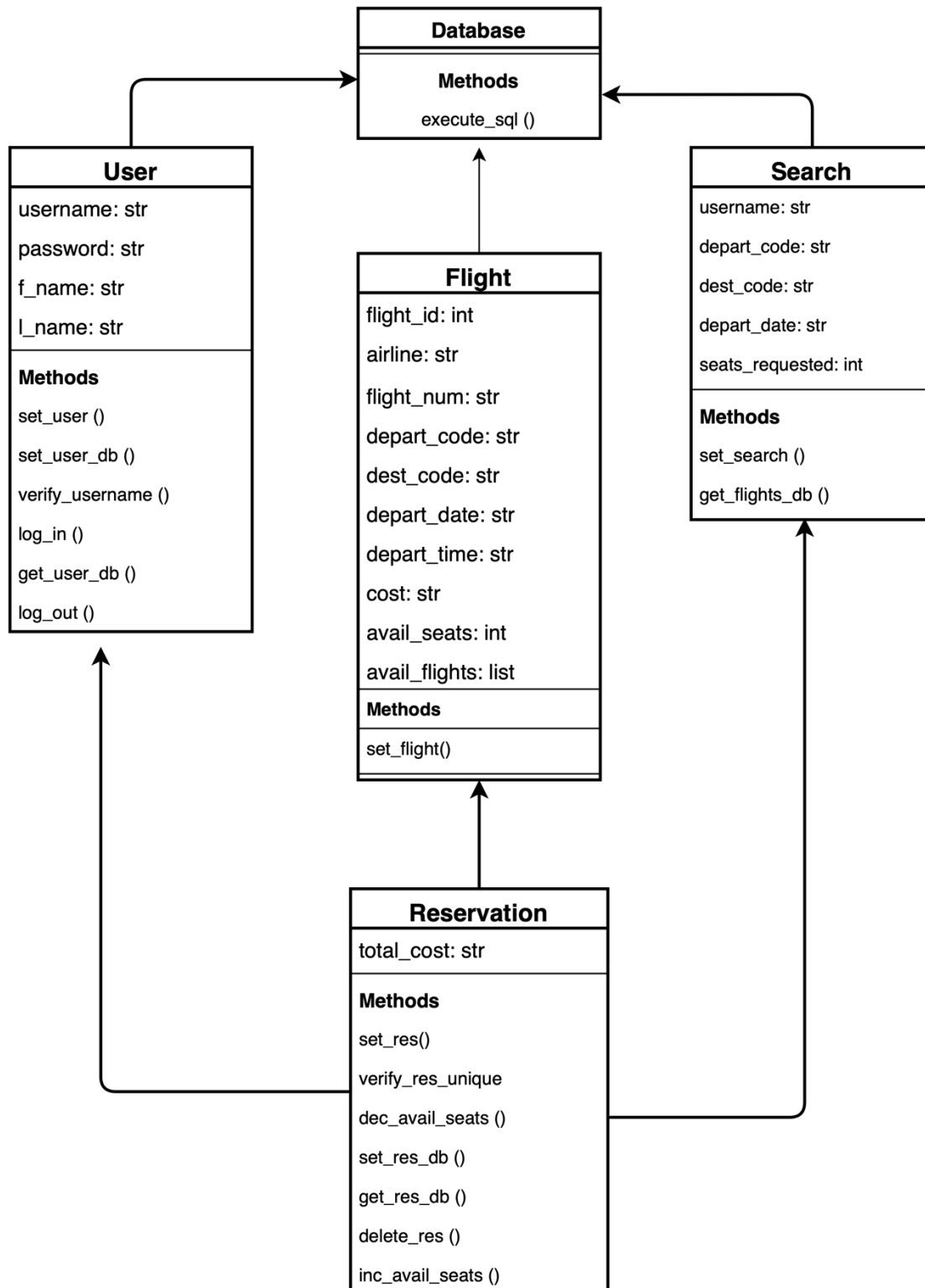
Users Table

Table Fields	Datatype
username	Varchar - Primary Key
Pass	Varchar – Not Null
F_Name	Varchar – Not Null
L_Name	Varchar – Not Null

Reservations Table

Table Fields	Datatype
Flight_id	Int – Primary Key
Username	Varchar – Foreign Key
Total_Cost	Int
Seats_reserved	Int – Not Null

UML Diagrams



Global Variable

1. Global user
 - a. User = User ()
2. Global search
 - a. Search = Search ()
3. Global res
 - a. res = Reservation ()
4. Global flight
 - a. Flight = Flight ()

Class Summary

1. Database()
2. User():
 - a. Inherits Database()
3. Flight():
 - a. Inherits Database()
4. Search():
 - a. Inherits Database()
5. Reservation():
 - a. Inherits User()
 - b. Inherits Flight()
 - c. Inherits Search()

Method Summaries

Database ()

1. Connect_db
 - a. Input: sql, update=False
 - b. Creates a connection to test_db
 - c. If update is True – INSERT or UPDATE
 - i. Runs SQL
 - ii. Returns None
 - d. If update is False -- SELECT
 - i. Runs SQL
 - ii. Returns results of SQL query
 - e. Closes connection with test_db

User ()

1. User_reg.Set_users (username, password, f_name, l_name)
 - a. Sets data for a user object
 - b. Sets object values
 - c. Returns None
2. User_reg.set_users_db ()
 - a. Updates user table with user registration information
 - b. Calls verify username
 - c. If username unique – returns True
 - d. If username not unique – returns False
3. User_reg.Verify_username ()
 - a. Verifies requested username is unique
 - b. Queries the users table to determine if username has been used
 - c. If username unique – returns True
 - d. If username not unique – returns False
4. User.Log_in (username, password)
 - a. Verifies username and password match
 - b. Uses global variable user
 - c. Queries users table
 - d. If username and password match – returns True
 - e. If username and password don't match or username doesn't exist – returns False
5. User.Get_user_db ()
 - a. Uses global variable user
 - b. Query users table for user information
 - c. Updates object value
 - i. User.username

- ii. User.f_name
 - iii. User.l_name
- 6. User.Log_out ()
 - a. Initializes all global variables at log out
 - b. Uses global variable user
- 7. Update_user (username, f_name, l_name)
 - a. Allows the user to update password, f_name and l_name
 - b. Must be logged in to update user
- 8. Delete_user (username)
 - a. Allows the user to delete their account
 - b. Must be logged in to delete user
- 9. Get_password(username)
 - a. Returns the password for the selected username

Flight ()

- 1. Flight.Set_flight(flight_id, airline, flight_num, depart_code, dest_code, depart_date, depart_time, cost, avail_seats, avail_flights)
 - a. Set flight data

Search ()

- 1. Search.set_search (depart_code, dest_code, depart_date, seats_requestes)
 - a. sets data for the search object
- 2. search.get_flights_db ()
 - a. queries flights table to determine available flights
 - b. applies SQL logic to ensure there are enough available seats
 - c. returns False if there are no seats
 - d. returns list flight.avail_flights if there are flights available
- 3. search.get_return_flights_db()
 - a. queries flight table for return flights from round trip search
 - b. applies SQL logic to ensure there are enough available seats
 - c. returns False if there are no seats
 - d. returns list flight.return_flights if there are flights available

Reservations ()

- 1. res.set_res (flight_id, username, seats_requested)
 - a. sets values of reservation
 - b. calls verify_res_unique()
 - c. if reservation is not unique – returns False
 - d. if reservation is unique updates:
 - i. res username
 - ii. res flight_id

- iii. res seats_reserved
 - iv. res total_cost
 - v. call dec_avail.seats()
- 2. self.Dec_avail_seats
 - a. Decreases the available seats field in the flights table by the number of seats to be reserved
 - b. Called by res.create_res ()
 - c. UPDATES flights table to the correct number of avail_seats based on seats_requested
- 3. Res.Set_res_db ()
 - a. Updates reservations table with the reserved flight information
 - b. UPDATE operation on reservations table
 - c. If reservation was successfully created – returns True
 - d. If reservation was not successfully created – returns False
- 4. Res.Get_res_db (username)
 - a. Gets the reservation from the user, flights and users table to display on the Show Reservations screen
 - b. Queries 3 tables to get all reservation information
 - c. Returns a list of reserved flights
- 5. Res.Delete_res (username, flight_id, seats_reserved)
 - a. Deletes a reservation from the reservations table
 - b. Calls inc_avail_seats (flight_id, seats_reserved)
- 6. Self.Inc_avail_seats (flight_id, seats_reserved)
 - a. Increases avail_seats in the flights table by the number of seats_reserved in the reservations table
 - b. Queries flights table to get avail_seats
 - c. Calculates new avail_seats
 - d. UPDATE flights table avail_seats by new value, queries by flight_id

GUI Frames

Home Frame

1. Load from for the application

Login Frame

1. Allows the user to login or logout of the system
 - a. Login required to search flights or show reservations
2. Methods
 - a. Login_user()
 - i. Calls verify_entry()
 - ii. Calls user.login()
 - iii. Calls clear_text()
 - b. Verify_entry() – verifies all fields are fill out
 - c. Clear_text()
 - d. Logout_user()
 - i. Calls user.logout()

Register Frame

1. Allows the user to register information
2. Methods
 - a. Register()
 - i. Calls verify_pass_match()
 - ii. Calls verify_entry()
 - iii. Calls user_reg.set_user()
 - iv. Calls user_reg.set_user_db()
 - v. Deletes user_reg
 - b. Verify_pass_match()
 - c. Verify_fields()
 - d. Clear_text()

Search One Way Flights Frame

1. User searches for available one way flights
2. Methods
 - a. Search_flights()
 - i. Calls verify_login()
 - ii. Calls search.set_search()
 - iii. Calls search.get_flights_db
 - iv. Flight.avail_flights gets search.get_flights_db
 - b. Verify_login()
 - c. Clear_text()

Search Round Trip Flights Frame

1. User searches for available round trip flights
2. Methods
 - a. Search_flights()
 - i. Calls verify_login()
 - ii. Calls search.set_search()
 - iii. Calls search.get_flights_db()
 - iv. Flight.avail_flights = search.get_flights_db()
 - v. Calls search.get_return_flights_db()
 - vi. Flight.return_flights = search.get_return_flights_db()
 - b. Verify_login()
 - c. Clear_text()

Show_Res Frame

1. Shows reservations for logged in user
2. Methods
 - a. Build_tree() - builds the tree for the reservation information
 - i. Calls res.get_res_db()
 - ii. Res = res.get_res_db()
 - b. Cancel_flight()
 - i. Calls res.delete(res)

Avail_Flights Frame

1. Shows the results of the available flights search
2. Methods
 - a. Build_tree() – builds the tree to show available flights
 - i. Accesses flight.avail_flights
 - b. Reserve_flight() – reserves the selected flight
 - i. Calls res.set_res()
 - ii. Calls res.set_res_db
 - c. Calls avail_flights_return frame if round trip search
 - d. Else calls show_res frame

Avail_Flights Return Flights Frame

1. Shows the results of the return flights for a round trip search
2. Methods
 - a. Build_tree() – builds the tree to show available flights
 - i. Accesses flight.return_flights
 - b. Reserve_flight() – reserves the selected flight
 - i. Calls res.set_res()
 - ii. Calls res.set_res_db
 - c. Calls show_res frame

Flights Table Data Set

Flight ID	Airline	Flight Number	Depat Code	Dest Code	Depart Date	Depart Time	Cost	Available Seats
1	United Airlines	UA1234	IAH	PDX	2/16/22	9:00:00	\$ 200.00	10
2	United Airlines	UA5678	IAH	SFO	2/16/22	12:00:00	\$ 300.00	20
3	United Airlines	UA9876	IAH	PDX	2/16/22	18:00:00	\$ 200.00	30
4	United Airlines	UA9123	IAH	LAX	2/16/22	14:00:00	\$ 400.00	40
5	American Airlines	AA1234	IAH	PDX	2/16/22	11:00:00	\$ 200.00	50
6	American Airlines	AA9876	IAH	PDX	2/16/22	20:00:00	\$ 200.00	60
7	American Airlines	AA5678	IAH	SFO	2/16/22	22:00:00	\$ 300.00	70
8	American Airlines	AA9123	IAH	LAX	2/16/22	8:00:00	\$ 400.00	80
9	United Airlines	UA2345	SFO	IAH	2/22/22	14:00:00	\$ 300.00	90
10	United Airlines	UA3456	LAX	IAH	2/22/22	14:00:00	\$ 400.00	100
11	United Airlines	UA4567	PDX	IAH	2/22/22	14:00:00	\$ 200.00	0
12	American Airlines	AA2345	SFO	IAH	2/22/22	14:00:00	\$ 300.00	100
13	American Airlines	AA3456	LAX	IAH	2/22/22	14:00:00	\$ 400.00	1
14	American Airlines	AA4567	PDX	IAH	2/2/22	14:00:00	\$ 200.00	0
15	ABC	ABC1234	IAH	SEA	2/22/22	14:00:00	\$ 400.00	100
16	DEF	DEF1234	IAH	WAS	2/22/22	14:00:00	\$ 400.00	100

Unit Testing

1) User Class Tests

1. Test_initialize_user()
2. Test_set_user()
3. Test_set_user_db(global_user1)
 - a. Tests that the system can write to the database with no errors
 - b. Calls verify_username()
4. Test_set_user_db_user_already_registered(global_user1)
 - a. Tests to see if a repeat user can be written to the database
5. Test_verify_username_user_already_registered(global_user2)
 - a. Tests to see if a username can be used more than once
6. Test_verify_username_user_does_not_exist()
 - a. Test to see if a unique username can be registered
7. Test_login_username_password_match(global_user1)
 - a. Tests to see if a user can login with a matching username and password
8. Test_login_username_and_password_do_not_match(global_user1)
 - a. Tests to see if a registered user can login with the wrong password
9. Test_login_username_does_not_exist()
 - a. Tests to see if an unregistered user can login
10. Test_logout(global_user1, global_search)
 - a. Tests to see if logout() initializes all global objects
 - i. User
 - ii. Search
 - iii. Res
 - iv. Flight
11. Test_delete_existing_user()
12. Test_update_user_existing_user()

2) Search Class Tests

1. Test_initialize_search()
2. Test_set_search()
3. Test_get_flights_db_no_seats_available(global_user1)
 - a. Tests if a user can find any flight on a search that has 0 available seats
4. Test_get_flights_db_not_enough_seats_available(global_user1)
 - a. Tests that one flight is returned when a user searches for 2 seats. One flight has enough available seats the other flight has 1 available seat.
5. Test_get_flights_db_incorrect-flight_data(global_user1)

- a. Tests if any flights are returned with a search that does not exist in the flights table
- 6. Test_get_flights_db_1_flight_available(global_user1)
 - a. Tests one flight is returned for a search that only has one available flight
- 7. Test_get_flights_db_2_flights_available(global_user1)
 - a. Tests 2 flights are returned for a search that has 2 available flights
- 8. Test_get_flights_return_db_no_return_flight(global_user1)
 - a. Tests get_flights_return_db method returns false when no round trip flights are found
- 9. Test_get_flights_return_db_return_flight_available(global_user1)
 - a. Tests the return flight for the get_flights_return_db method

3) Reservation Class Tests

- 1. Test_initialize_reservatio()
- 2. Test_set_res()
 - a. Also tests decrease_avail_seats()
 - b. Resest_avail_seats resets seat number for subsequent testing
- 3. Test_set_res_db_and_get_res_db()
 - a. Tests that a reservation is written to the database and can be queried
 - b. Resest_avail_seats resets seat number for subsequent testing
- 4. Test_verify_res_unique_reservation_already_exists(global_user1, global_search)
 - a. Tests to see if a reservation can be made for the same flight number more than once
- 5. Test_verify_res_unique_no_reservation_exists(global_user1)
 - a. Tests that a reservation is unique when no reservation exists
- 6. Test_increase_avail_seats()
 - a. Tests that the seat number is increased by the number of seats reserved
 - b. Resest_avail_seats resets seat number for subsequent testing
- 7. Test_only_logged_in_user_name_returns_in_query(global_user1, global_user2, global_search)
 - a. Set up reservations under 2 usernames and verify the only flights returned by get_user_db are for the username queried. Not both of the usernames
 - b. Delete the reservations
 - c. Resest_avail_seats resets seat number for subsequent testing
- 8. Test_multiple_reserved_flights_are_queried(global_user1, global_user2, global_search)
 - a. Tests that multiple reservations can be returned by get_user_db
 - b. Delete reservations
 - c. Resest_avail_seats resets seat number for subsequent testing