CS009C Final Project Proposal

Name:_____Yiyang Chen_____ NetID: _____ychen1420___

1. Topic:
   Flight Online Comparing Platform

2. List what information each account will likely have (should have **at least** 10 pieces of information, and if you can, write their data types):

UserAccount:
1. string username
2. string password
3. string passportID
4. string orderID

FlightOrder:
5. string providerType ("Airline" or "ThirdParty")
6. string providerName
7. string flightNumber
8. string sourceAirport
9. string destinationAirport
10. string departureTime
11. string arrivalTime
12. double basePrice
13. double baggageFee
14. FareType* chosenFare (a pointer to the fare type they picked)

3. List the different types of accounts there can be (**at least 3**):

1. UserAccount : Represents passengers who log in and book flights

2.Provider Accounts (two types): -
Airline Provider - Direct airline bookings with included baggage and insurance
ThirdParty Provider: Third-party platform bookings with separate fees

3.FareType (Base Class) : Uses inheritance and polymorphism with two children classes:
a. EconomyClass : Standard pricing
b.FirstClass: 3*regular price

Note: My project has 5 classes in total. UserAccount represents passengers, and FlightOrder represents bookings. I also created a FareType base class, along with two child classes, they are : EconomyClass and FirstClass. They inherit from it and use polymorphism to calculate different prices.

4. List **at least 4** ways accounts can interact or are related with each other:

1. UserAccount works with FlightOrder:
When someone logs in, they can create flight bookings. Their passport ID gets saved with the booking so we know who booked what.

2. UserAccount → Provider Accounts(Airlines and Third Party)
When a user logs in, they can search, and compare flights from both Airline and third party providers. After that , they made decisions to choose one. The system shows price differences and fee comparisons to help users choose.

3. UserAccount → Shopping Cart
After the user choose a certain flight and price they want, users can add more different flight options to their cart.

4.By inheritance, FareType and its children classes:
EconomyClass and FirstClass both inherit from FareType and override the price calculation function.When a user picks First Class, the program automatically uses the 3x multiplier through polymorphism.

5. Write **at least 1** complex thing each account can have that is NOT a single data type:

First: The login function takes in a vector, of all users and returns a pointer to who successfully logged in.

Second: for FlightOrder, it uses a polymorphic FareType system. There's a FareType* pointer inside each FlightOrder that can point to either EconomyClass or FirstClass.
When you call calculateFinalPrice(), it runs different code depending on what type it's pointing to.

Third, for FlightOrder, the constructor takes 13 parameters at once, which contains a mix of 8 strings and 5 doubles.