**Project Summary:**

1. Project Name: A Flight Away
2. Members: Mia Nguyen
3. High-level Overview: A Flight Away is a Java application that will allow users to easily book flights. In this application, users will be able to log into their accounts, view upcoming flights to different places, compare flight prices, book multiple flights, and cancel booked flights.

**Project Requirements:**

1. Requirements and Responsibilities:
   1. Log in page
      1. Users should be able to create an account or log in with an existing username and password stored in the database
      2. There might be restraints on how many usernames/passwords we can store in our database
   2. View flights page
      1. Users should be able to view multiple flights for multiple locations, dates, and prices
      2. The number of flights displayed might be limited (i.e.) they might only be listed up to a certain date
   3. Book flights
      1. Users should be able to book any flights they see
      2. Users may only be able to book a certain number of flights at a time
   4. Take payment (not real payment in this application)
      1. Users will have to input payment and contact information to complete the booking process
   5. View booked flights
      1. Users can view the flights they have booked
   6. Cancel flights
      1. Users can cancel flights they have booked

**User and Tasks: Use Cases**

1. My application will have one user: people who want to book flights
2. Different Use Cases to Consider:
   1. Logging In
      1. Users will be prompted to enter in a username/password
      2. If successful, the application will then lead the user to a page where they can view flights
      3. If not successful, users will be prompted to enter in another username/password combination until there is a match found in the database
         1. Users can also choose to create a new account
   2. Viewing Flights
      1. Users will be able to view flights
      2. Users can filter flights by date, location, and price
         1. Users cannot view past flights or flights too far into the future
         2. Users cannot view flights to an invalid destination
         3. Users cannot view flights for invalid prices (negative prices)
   3. Booking Flights
      1. Users will be able to book flights
      2. Users will enter in payment and contact information to confirm their booking
      3. Users will be prompted to re-enter their payment information if it is invalid (for example, a credit card with 20 digits would be invalid)
   4. Cancelling Flights
      1. Users will be able to cancel flights they have booked
3. UML Activity Diagram

Diagram

Description automatically generated

1. Architecture Diagram

Diagram

Description automatically generated

1. Data Storage
   1. I am planning to use CSV files to store username and password information
   2. I am planning to use MySQL to store flight data (dates, times, destinations, prices, etc.)
      1. I might end up using CSV files to store flight data, depending on how I end up building out my project
2. UI Mockups/Sketches

Diagram, text

Description automatically generated

Text

Description automatically generated with medium confidence

1. Diagram

   Description automatically generated with medium confidenceUML Class Diagram and Pattern Use
2. User Interactions/UML Sequence Diagrams

Diagram, table

Description automatically generated