

Ticket (entity)

- Store Ticket's flight number
- Store Ticket's departure and arrival city
- Store Ticket's departure and arrival time
- Store Ticket's passenger's name
- Store Ticket's passenger's username
- Store Ticket's passenger's luggage number
- Store Ticket's passenger's seat class
- Store Ticket's seat number
- Store Ticket's price
- Store Ticket's id
- Store Ticket's boarding gate
- Getters/setters for everything above
- Generate a ticket by using all information above with toString method.

Flight
Customer
TicketManager

Flight (entity)

- Store flight number
- Store origin city of the flight
- Store destination city of the flight
- Store departure and arrival time of flight using `LocalDateTime` class
- Store seat capacity, available seats number
- Store boarding gate
- Store price
- Store distance traveled, i.e., from departure to destination
- Store Array of seats number available(haven't been reserved yet), with initially 10 seat number stored, where seat number is string in the format of "1A". As each seat is assigned, set the corresponding element of the array to "X" to indicate that the seat is no longer available.
- (All these private attributes can't be modified with the methods defined in flight class, except number of available seats, and array of seats number)

FlightManager
Ticket
Customer

Customer (entity)

- Stores customer personal information(name, password and username).
- Stores account balance, millage earned.
- Store membership information(a method for upgrading, i.e, if reached xxx miles, then upgrade)
- Check if the given password in string form is correct or not for this customer.
- Replace the given password, name and username in string form and replace the original personal information.
- Methods: Getter methods to get this customer's personal information such as name, username, current balance amount, current millage amount.
- Upgrade the current balance amount for this customer (Increase or decrease).
- Upgrade the membership status for this customer.
- Show customer's personal information with toString method.

Ticket
CustomerManager

TicketManager (Use-case)

- Store all sold tickets
- Generate tickets for customers
- Getter method
- Book tickets
- Cancel sold tickets
- Modify sold tickets (date, time)
- Store total revenue
- Class upgrade.
- Apply discounts to ticket price if applicable.
- Apply fines for late cancellation or late modification if applicable.

Ticket

FlightManager (Use-case)

- Contains a `HashMap<String, Flight>` (default to be empty) maps flight number to flight.
- Add flight to this hashmap
- Access flight by flight number
- Verify flight number
- Order flights by travel distance
- Provide a list that contains all flight numbers from the departure city and destination.
- Get flight's price.
- Return *ArrayList of availableSeat by flight number*

Flight

CustomerManager (Use-case)

- Add new customers to customer manager system.
- Access each customer by the customer's username, and show the corresponding customer's personal information including: username, name, email address, account balance, mile earned, membership status)
- Check if the given username exist in the customer manager system or not.
- Check if the given password is correct for the given username in the customer manager system.
- Updates customer personal information(name).
- Updates customer personal information(password).
- Updates customer personal information(username).
- Updates account balance increase or decrease.
- Updates mile earned.
- Updates membership status.

Customer

TicketReservationSystem (Controller)

- Generate all required managers.
- Transfer user inputs into requests.
- Send Ticket, User, Flight requests to the corresponding Manager:
- CustomerManager: Add new customer to customer manager system, check if the given customer username exist or not. Show the given customer information and update the customer's balance amount.
- FlightManager: match routes with flight and return list of flight nums, print flight info by flight number, select flight by flight num, shows ArrayList of seats in this flight by flight number, reserve seat and return seat_num.
- TicketManager: Create ticket by given username, seat number and flight number, then store the ticket into the system, print out the air ticket.
- Generate and sell tickets using those Manager classes.

FlightManager
TicketManager
CustomerManager

UI

(basic command line interface)

- Let user input info like name, username, password and so on by using a Scanner class instance.
- Process user's requests like loading balance, selecting flight, selecting seat number.

TicketReservationSystem
TicketManager
FlightManager
CustomerManager