Design document

Use cases

1. System create a plane
2. System read the file of prior reservations and store it in current plane
3. System display reservation menu

Use Cases #1: [Add individual passenger]

1. User input “P” to select add individual passenger
2. System ask the passenger name
3. User input the passenger name
4. System ask a class of service
5. User input a class of service
6. System ask seat preference
7. User input seat preference
8. System find machine seats
9. System make reservation, print succeed message
10. System go back to the beginning, display reservation menu again.

Variation #1

In step 5, if user input wrong service class name or class is full, print error message and ask user to enter again

Variation #2

In step 7, if user input wrong preference or preferred seats are full, print error message and ask user to enter again

Use Cases #2: [Add group passenger]

1. User input “G” to select add group
2. System ask group name
3. User input group name
4. System ask the name of passengers separated by comma
5. User input the name of passengers
6. System ask service class
7. User input service classes
8. System find adjacent seats for the group
9. System make reservation
10. System go back to the beginning, display reservation menu again

Variation #3

In step 3, if user input empty group name, print error message and ask user to enter again

Variation#4

In step7, if user input wrong class name, print error message and ask user to enter again

Use Cases #3: [Individual passenger reservation cancellation]

1. User input “C” to select cancel reservation
2. System ask user to input choice to cancel individual passenger or group reservation
3. User input “P” to cancel individual passenger
4. System ask user to input the passenger name
5. User input the passenger name
6. System race the plane seats, find the passenger
7. System make cancellation, print succeed message
8. System go back to the beginning, display reservation menu again

Variation #5

In step 5, when user input a passenger name in a passenger group, print error message and go back to the beginning

Use Cases #4: [Group reservation cancellation]

1. User input “C” to select cancel reservation
2. System ask user to input choice to cancel individual passenger or group reservation
3. User input “G” to cancel group reservation
4. System ask user to input the group name
5. User input the group name
6. System find the group
7. System make cancellation, print succeed message
8. System go back to the beginning, display reservation menu again

Variation #6

In step 5, if user input empty group name, print error message and ask user to enter again

Use Cases #4: [Print available seats]

1. User input “A” to print
2. System check empty seats
3. System print available seats

Use Cases #5: [Print manifest]

1. User input “M” to print
2. System check seated seats
3. System print manifest

Use Case #6: [Quit]

1. System save the reservation in the plane into a file
2. Print quit message
3. Program stops

CRC cards

|  |  |
| --- | --- |
| Reservation System | |
| Print system menu | Plane |
| Call reservation method | File Handler |

|  |  |
| --- | --- |
| Plane | |
| Add passenger | Seat |
| Add group | Group |
| Cancellation |  |
| Print available seats |  |
| Print manifest |  |

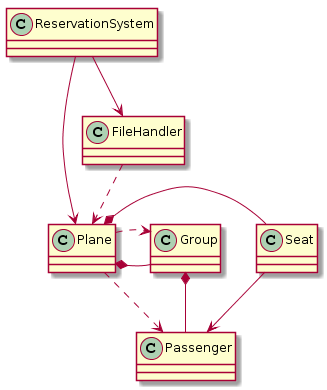
|  |  |
| --- | --- |
| Seat | |
| Seat row number | Passenger |
| Seat column number |  |
| Seat Preference |  |
| Service class |  |

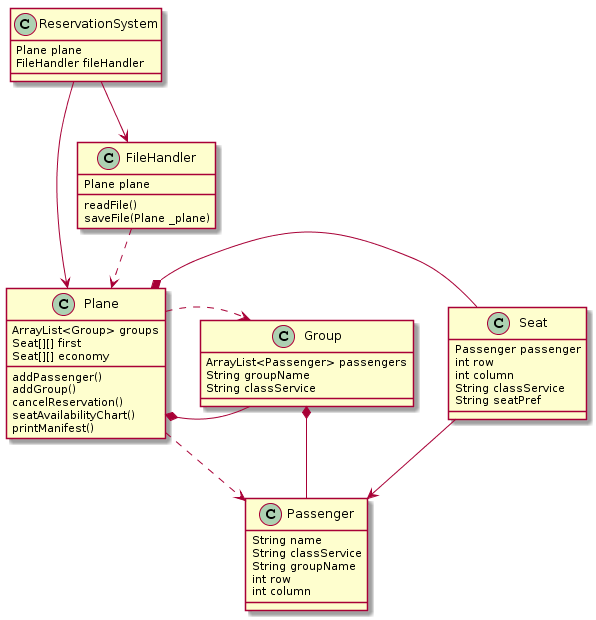
|  |  |
| --- | --- |
| Passenger | |
| Passenger name |  |
| Passenger group name |  |
| Passenger service class |  |
| Passenger row number |  |
| Passenger column number |  |

|  |  |
| --- | --- |
| Group | |
| Service class | Passenger |
| Group name |  |
| Passenger list |  |

|  |  |
| --- | --- |
| File Handler | |
| Read file | Plane |
| Save file | File |

UML Diagram





UML Sequence diagram

