Assignment 1 Use Cases

# Add a Passenger

|  |  |  |
| --- | --- | --- |
| Step | User Action | System Response |
| 1 | User requests to add individual passenger |  |
| 2 |  | Prompt user for passenger data.  Name:  Service Class:  Seat Preference: |
| 3 | User inputs data |  |
| 4 |  | Look for available seat matching given preference. |
| 5 |  | Print seat number assigned to passenger. |
| 6 |  | Return to main menu and prompt for user input. |

Variation #1

1. Step 3, user inputs invalid seat preference or service class.
2. System responds, “Request failed”, and specifies if the class or seat was incorrect.
3. Return to step 2.

Variation #2

1. Step 4, there is no seat matching the user’s preference.
2. System responds, “Request failed”, and the specified seat preference isn’t available.
3. Return to step 2.

# Add Group

|  |  |  |
| --- | --- | --- |
| Step | User Action | System Response |
| 1 | User requests to add a group |  |
| 2 |  | Prompt user for group’s data.  Group Name:  Names:  Service Class: |
| 3 | User inputs data |  |
| 4 |  | Look for available adjacent seats for the group. |
| 5 |  | Print seat numbers given to the group. |
| 6 |  | Return to main menu and prompt for user input. |

Variation #1

1. Step 3, user inputs invalid service class.
2. System responds, “Request failed”, and specifies that the class is incorrect.
3. Return to step 2.

Variation #2

1. Step 4, insufficient seats available.
2. System responds, “Request failed”, and that there aren’t enough seats.
3. Return to step 2.

# Cancel Reservations

|  |  |  |
| --- | --- | --- |
| Step | User Action | System Response |
| 1 | User asks to cancel a reservation |  |
| 2 |  | System asks if the cancellation is for an individual or group.  Cancel [I]ndividual or [G]roup? |
| 3 | User specifies request |  |
| 4 |  | Ask for name of individual or group to cancel request for. |
| 5 | User inputs name |  |
| 6 |  | System looks for all seats reserved by the individual or group. |
| 7 |  | Clears the seat(s) of reservations and notify user of successful cancelation. |
| 8 |  | Return to main menu and prompt for user input. |

Variation #1

1. Step 3, user specifies invalid request.
2. System responds that the request failed.
3. Return to step 2.

Variation #2

1. Step 5, user specifies invalid name.
2. System responds that the name isn’t valid.
3. Return to step 4.

Variation #3

1. Step 6, no seats occupied by specified entity.
2. System notifies that no seats removed.
3. Continue to step 8

# Print Seating Availability Chart

|  |  |  |
| --- | --- | --- |
| Step | User Action | System Response |
| 1 | User asks for seating availability chart |  |
| 2 |  | System asks for service class to display.  Service Class: |
| 3 | User specifies the class |  |
| 4 |  | System prints seat availability chart for the requested class. |
| 5 |  | Return to main menu and prompt for user input. |

Variation #1

1. Step 3, user specifies invalid class.
2. System responds that the class is incorrect.
3. Return to step 2.

# Print Manifest

|  |  |  |
| --- | --- | --- |
| Step | User Action | System Response |
| 1 | User asks for manifest |  |
| 2 |  | System asks for service class to display.  Service Class: |
| 3 | User specifies the class |  |
| 4 |  | System prints manifest for the requested class. |
| 5 |  | Return to main menu and prompt for user input. |

Variation #1

1. Step 3, user specifies invalid class.
2. System responds that the class is incorrect.
3. Return to step 2.

# Quit

|  |  |  |
| --- | --- | --- |
| Step | User Action | System Response |
| 1 | User asks to quit program |  |
| 2 |  | System saves all changes to text file |
| 3 |  | Program ends |