**Make Trip Plan**

1. Participating Actors
   * Costumer
2. Basic Flow
   * System shows a form for trip plan
   * Customer fills the form with trip information (Routes, Dates) which he/she want to search.
   * By clicking on “Submit”, the system searches available flights and shows it customer.
   * Customer selects flights and system automatically calculates total fee.
   * After this point optionally, customer can make reservation.
3. Alternative Flows
   1. Customer fills incorrect data
      * System show error and return back.
   2. Customer clicks “Back“ button
      * System returns back the “Trip Plan Page” but all inputs have already filled.
   3. Customer clicks “Cancel” button
      * System returns back the “Trip Plan Page” but all inputs are blank.
4. Time Dependencies
   1. The frequency of execution: ~1000-1500 times a day
   2. The anticipated stagnation: N/A
   3. The typical execution time: ~ 1 min
   4. The maximum execution time: unlimited
5. Values obtained by the actors after use case finishing

* A message indicating the success or failure of the operation
* Total fee

**Make Reservation**

1. Participating Actors
   * Costumer
2. Basic Flow

* The system displays a form allowing for making reservation.
* Customer chooses route and date. Then, system shows available flights. Customer chooses flight also.
* Customer clicks “Submit” button. Authentication system works and control whether logged in.
* At this point optionally, Customer can make payment.
* The system verifies the completeness and accuracy of the data.
* The system saves the data in the reservations register.
* The system informs the operations carried out by displaying an adequate message.
* The system redirects to “Ticket Management Page”

1. Alternative Flows
   1. The system finds incomplete or incorrect data
      * The system again displays a form with selected fields in which errors were found
   2. Customer has not been logged in yet
      * System redirects to “Login Page”.
   3. Customer clicks “Back“ button
      * System returns back the “Make Reservation Page” but all inputs have already filled.
   4. Customer clicks “Cancel” button
      * System returns back the “Make Reservation Page” but all inputs are blank.
2. Time Dependencies
   1. The frequency of execution: ~500-750 times a day
   2. The anticipated stagnation: N/A
   3. The typical execution time: ~ 2 min
   4. The maximum execution time: unlimited
3. Values obtained by the actors after use case finishing
   * A message indicating the success or failure of the operation

**Exchange Ticket**

1. Participating Actors
   * Costumer
2. Basic Flow

* The system authenticates customer and redirects to “Ticket Management Page”.
  + Customer picks a flight out of all tickets.
  + By clicking “Exchange” button, system redirects to “Make Reservation Page”.
  + After the end of reservation, old ticket will be canceled.
  + The system informs the operations carried out by displaying an adequate message.

1. Alternative Flows
   1. Customer clicks “Cancel” button during reservation.
      * The system returns back “Ticket Management Page”.
   2. Customer tries to exchange ticket before 36 hours of flight.
      * This situation is not acceptable, so system will not go ahead with “Reservation Page”.
      * The system informs customer by displaying an adequate message
2. Time Dependencies
   1. The frequency of execution: ~10-20 times a day
   2. The anticipated stagnation: N/A
   3. The typical execution time: ~ 2 min
   4. The maximum execution time: unlimited
3. Values obtained by the actors after use case finishing
   * A message indicating the success or failure of the operation

**Cancel Ticket**

1. Participating Actors
   * Costumer
2. Basic Flow

* The system authenticates customer and redirects to “Ticket Management Page”.
  + Customer picks a flight out of all tickets.
  + By clicking “Cancel” button, system displays “Approval Message”.
  + The system informs customer about canceling ticket by displaying an adequate message.

1. Alternative Flows
   1. Customer does not approve “Approval Message”.
      * The system redirects to “Ticket Management Page”.
   2. Customer tries to cancel ticket before 36 hours of flight.
      * This situation is not acceptable, so system will not displays “Approval Message”.
      * The system informs customer by displaying an adequate message.
2. Time Dependencies
   1. The frequency of execution: ~10-20 times a day
   2. The anticipated stagnation: N/A
   3. The typical execution time: ~ 1 min
   4. The maximum execution time: unlimited
3. Values obtained by the actors after use case finishing
   * A message indicating the success or failure of the operation

**Send E-mail**

1. Participating Actors
   * Costumer
2. Basic Flow
   * The system authenticates customer and redirects to “Ticket Management Page”.
   * Customer picks a flight out of all tickets.
   * By clicking “Send E-mail” button, system sends an e-mail that includes ticket.
   * The system informs customer by displaying an adequate message.
3. Time Dependencies
   1. The frequency of execution: ~900-1000 times a day
   2. The anticipated stagnation: N/A
   3. The typical execution time: ~ 1 min
   4. The maximum execution time: unlimited
4. Values obtained by the actors after use case finishing
   * A message indicating the success or failure of the operation.
   * An e-mail that includes ticket.

**Print Ticket**

1. Participating Actors
   * Costumer
2. Basic Flow
   * The system authenticates customer and redirects to “Ticket Management Page”.
   * Customer picks a flight out of all tickets.
   * By clicking “Print Ticket” button, system prints selected ticket.
   * The system informs customer by displaying an adequate message.
3. Time Dependencies
4. The frequency of execution: ~300-400 times a day
5. The anticipated stagnation: N/A
6. The typical execution time: ~ 1 min
7. The maximum execution time: unlimited
8. Values obtained by the actors after use case finishing
   * A message indicating the success or failure of the operation

**Add Flight**

1. Participating Actors
   * Manager
2. Basic Flow
   * The system authenticates manager and redirects to “Management Page”.
   * Manager clicks “Add Flight” button.
   * The system displays a form allowing for adding flight.
   * Manager chooses date and route of new flight.
   * By clicking “Add” button, flight will be added.
   * System will returns back “Management Page” and show a message about operation.
3. Alternative Flows
   1. Manager fills incorrect data
      * System show error and return back.
   2. Manager clicks “Cancel” button
      * System returns back the “Management Page”.
4. Time Dependencies
5. The frequency of execution: ~2-3 times a week
6. The anticipated stagnation: N/A
7. The typical execution time: ~ 1 min
8. The maximum execution time: unlimited
9. Values obtained by the actors after use case finishing
   * A message indicating the success or failure of the operation.

**Cancel Flight**

1. Participating Actors
   * Manager
2. Basic Flow
   * The system authenticates manager and redirects to “Management Page”.
   * Manager clicks “Cancel Flight” button.
   * The system displays a form allowing for canceling flight.
   * Manager chooses date and route of the flight will be deleted, then system shows all flights.
   * Manager selects a flight to cancel.
   * By clicking “Cancel” button, flight will be canceled.
   * System sends an e-mail to passengers about cancelation.
   * System will returns back “Management Page” and show a message about operation.
3. Alternative Flows
4. Manager fills incorrect data
   * + System show error and return back.
   1. Manager clicks “Cancel” button
      * System returns back the “Management Page”.
   2. Manager clicks “Back“ button
      * System returns back the “Cancelation Page” but all inputs have already filled.
5. Time Dependencies
6. The frequency of execution: ~2-3 times in 3 months
7. The anticipated stagnation: N/A
8. The typical execution time: ~ 1 min
9. The maximum execution time: unlimited
10. Values obtained by the actors after use case finishing
    * A message indicating the success or failure of the operation.
    * Customers will receive an e-mail.

**Get Information**

1. Participating Actors
   * Manager
2. Basic Flow
   * The system authenticates manager and redirects to “Management Page”.
   * Manager clicks “Get Information” button.
   * The system displays a form allowing for selecting flights. (Multi-selection acceptable)
   * Manager chooses date and route of the flights, then system shows all flights.
   * Manager selects flights to get information.
   * By clicking “Report” button, report will be created.
   * System will returns back “Management Page” and show a message about operation.
3. Alternative Flows
4. Manager fills incorrect data
   * + System show error and return back.
   1. Manager clicks “Cancel” button
      * System returns back the “Management Page”.
   2. Manager clicks “Back“ button
      * System returns back the “Information Page” but all inputs have already filled.
5. Time Dependencies
6. The frequency of execution: ~1-2 times a week
7. The anticipated stagnation: N/A
8. The typical execution time: ~ 1 min
9. The maximum execution time: unlimited
10. Values obtained by the actors after use case finishing
    * A message indicating the success or failure of the operation.