Airplane Tickets

Requirements

Functional requirements:

* Login customer
* Create new customer account
* Addind data of customer to account
* Change data of customer to account
* Deleting of customer account
* Customer subscribing to a direction
* Create new order of tickets for customer
* Getting a list of siutable options of tickets for customer
* Selection of the flight date and direction
* Selection of the number of passengers
* Selection of the age of passengers
* Selection of the minimum and maximum price of the ticket
* Selection of the payment method for the order
* Selection of payment currency
* Selection of the seat on the flight board
* Selection of ticket refund option
* Selection of the possibility to change the date of departure
* Selection of the ticket cabin class
* Selection the number of transfers
* Selection of transfer durations
* Selection of departure and arrival times
* Selection of departure and arrival airports
* Selection of transfer airports
* Selection of menu on the flight board
* Selection of the number and weight of suitcases in hand luggage and baggage
* Selection of participation in the airline loyalty program
* Getting a order with tickets and payment confirmation for customer by email
* Getting a information about the status of flight
* Getting a newsletter by e-mail about the status of the customer's flights
* Online check-in
* Online check-out
* Adding extra luggage
* Adding insurance by flight or baggage
* Create a airline request in customer account
* Changing the flight's date if the ticket has such an oppotunity
* Cancelling flight if the ticket has such an oppotunity

Non-functional requirements:

* Available 24/7
* Multi-language system
* PC and mobile phone version os system

Use Cases

* Use case 1
  * Title: Login customer
  * Primary actor: Customer
  * Success scenario: The customer can enter his personal account by filling out a special form and entering his email and password.

* Use case 2
  * Title: Create new customer account
  * Primary actor: Customer
  * Success scenario: The customer can create a personal account by filling out a special form and entering the minimum required data: last name, first name, mail, password. Confirmation will be emailed to him.

* Use case 3
  * Title: Addind data of customer to account
  * Primary actor: Customer
  * Success scenario: The customer can add data in his personal account. Passport data, phone, date of birth. Confirmation will be emailed to him.

* Use case 4
  * Title: Change data of customer to account
  * Primary actor: Customer
  * Success scenario: The customer can change his data in his personal account: last name, first name, passport data, phone, mail, password. Confirmation will be emailed to him.

* Use case 5
  * Title: Deleting of customer account
  * Primary actor: Customer
  * Success scenario: The customer can delete his personal account. Confirmation will be emailed to him.

* Use case 6
  * Title: Customer subscribing to a direction
  * Primary actor: Customer
  * Success scenario: The customer can subscribe in his personal account to receive advantageous offers from the airline to the destinations he needs by mail.

* Use case 7
  * Title: Create new order of tickets for customer
  * Primary actor: Customer
  * Success scenario: To select the optimal flight, the customer enters the required minimum information (date of departure, city of departure, city of arrival, number of passengers, age of passengers, seat class) into the system and starts the search.

* Use case 8
  * Title: Getting a list of siutable options of tickets for customer
  * Primary actor: Customer
  * Success scenario: TAfter the customer enters the primary information, the system displays all available options on the screen, and also suggests using a filter for a more detailed flight search.

* Use case 9
  * Title: Selection of the flight date and direction
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer indicates the date of the flight.

* Use case 10
  * Title: Selection of the number of passengers
  * Primary actor: Customer
  * Success scenario: When placing an order, the customer specifies the number of passengers.

* Use case 11
  * Title: Selection of the age of passengers
  * Primary actor: Customer
  * Success scenario: When placing an order, the customer indicates the age of the passengers.

* Use case 12
  * Title: Selection of the minimum and maximum price of the ticket
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can

select the minimum and maximum price of the tickets to be displayed.

* Use case 13
  * Title: Selection of the payment method for the order
  * Primary actor: Customer
  * Success scenario: When placing an order, the customer can choose the method of paying for the order by card or in cash at the office of the airline or at partner companies.

* Use case 14
  * Title: Selection of payment currency
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can choose the currency for displaying ticket prices.

* Use case 15
  * Title: Selection of the seat on the flight board
  * Primary actor: Customer
  * Success scenario: When placing an order or later, during check-in for a flight, the customer can choose a seat from those available on the plane.

* Use case 16
  * Title: Selection of ticket refund option
  * Primary actor: Customer
  * Success scenario: When placing an order, the customer can choose a fare with the option to return the ticket.

* Use case 17
  * Title: Selection of the possibility to change the date of departure
  * Primary actor: Customer
  * Success scenario: When placing an order, the customer can choose a fare with the ability to change the date of departure.

* Use case 18
  * Title: Selection of the ticket cabin class
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can filter flights with seats of a certain class.

* Use case 19
  * Title: Selection the number of transfers
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can filter flights with a certain number of transfers.

* Use case 20
  * Title: Selection of transfer durations
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can filter flights with the duration of transfers he needs.

* Use case 21
  * Title: Selection of departure and arrival times
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can filter flights with departure and arrival at a specific time.

* Use case 22
  * Title: Selection of departure and arrival airports
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can filter flights with departures and arrivals at a specific airport.

* Use case 23

  * Title: Selection of transfer airports
  * Primary actor: Customer
  * Success scenario: When searching for the optimal flight, the customer can filter flights with transfers at the airports he needs.

* Use case 24
  * Title: Selection of menu on the flight board
  * Primary actor: Customer
  * Success scenario: When placing an order, the customer can choose meals during the flight from the proposed options or refuse it.

* Use case 25
  * Title: Selection of the number and weight of suitcases in hand luggage and baggage
  * Primary actor: Customer
  * Success scenario: When placing an order, the customer can choose the amount of luggage that he plans to take on board and checks into the luggage compartment.

* Use case 26
  * Title: Selection of participation in the airline loyalty program
  * Primary actor: Customer
  * Success scenario: The customer in his personal account can register in the airline's loyalty program by noting his desire in a special window. After confirmation of registration, the client is assigned an identification number, which he can enter every time he makes a new order and receive privileges from the airline.

* Use case 27
  * Title: Getting a order with tickets and payment confirmation for customer by email
  * Primary actor: Customer
  * Success scenario: After successful payment of the order, the customer receives a confirmation of payment and a file with the reservation by mail.

* Use case 28
  * Title: Getting a information about the status of flight
  * Primary actor: Customer
  * Success scenario: The customer can check the status of his flight in the system without being in his personal account, if he indicates the reservation number and last name in a special window: check-in is open, flight cancellation, change in departure time.

* Use case 29
  * Title: Getting a newsletter by e-mail about the status of the customer's flights
  * Primary actor: Customer
  * Success scenario: The customer receives informational letters with the status of his flight by mail: check-in is open, flight cancellation, change of departure time.

* Use case 30
  * Title: Online check-in
  * Primary actor: Customer
  * Success scenario: The customer enters his personal account, selects the desired order, registers for the flight. Confirmation comes by mail.

* Use case 31
  * Title: Online check-out
  * Primary actor: Customer
  * Success scenario: The customer enters his personal account, selects the desired order, cancels the check-in for the flight. Confirmation comes by mail.

* Use case 32

```
  * Title: Adding extra luggage
  * Primary actor: Customer
  * Success scenario: The customer enters his personal account, selects the
desired order that he wants to change and adds additional pieces of baggage.
Chooses a payment method, pays for the service. Confirmation comes by mail.

* Use case 33
  * Title: Adding insurance by flight or baggage
  * Primary actor: Customer
  * Success scenario: The customer enters his personal account, selects the
desired order that he wants to change and adds the service of self-insurance
during the flight or baggage. Chooses a payment method, pays for the service.
Confirmation comes by mail.

* Use case 34
  * Title: Create a airline request in customer account
  * Primary actor: Customer
  * Success scenario: The customer can send his question to the airline by
filling out a special form. The answer will come by mail.

* Use case 35
  * Title: Changing the flight's date if the ticket has such an oppotunity
  * Primary actor: Customer
  * Success scenario: The customer enters his personal account, selects the
desired order, which he wants to change and selects a new flight date, if such
an option is available for his tariff. After confirmation, a new reservation
comes to the mail.

* Use case 36
  * Title: Cancelling flight if the ticket has such an oppotunity
  * Primary actor: Customer
  * Success scenario: The customer enters his personal account, selects the
desired order, which he wants to cancel, if such an option is available for his
tariff, chooses the method of receiving funds. After confirmation, recalculation
and return rules come by mail.


## Resource

### Authentification

* Login
  * Protocol - HTTPS

### Order

* Creating a order
  * Protocol - HTTP
  * End Point - `http://servername.com:8080/orders` <!-- plural -->
  * URI Design - `http://servername.com:8080/orders`
                 `http://servername.com:8080/orders?
limit=<limit>&offset=<offset>` <!-- pagining -->
                 `http://servername.com:8080/orders?limit=<limit>`
* Searching a order
  * Protocol - HTTP
  * End Point - `http://servername.com:8080/orders`
  * End Point - `http://servername.com/dictonary/
  * URI Design - `http://servername.com/dictonary/<english>/<word>`
  * <english> - america or uk
  * URI Design - `http://servername.com:8080/orders?title=<title>`
  * Representation
  * HTML
* Represation
  * JSON
```

```json
{ "orders" :
  [{ "id" : 1,
     "title" : "order 1",
     "feed"  : "http://servername.com:8080/?name=order%20{id}",
     ...
   },
   { "id" : 2,
     "title" : "order 2",
     "feed"  : "http://servername.com:8080/?name=order%20{id}",
     ...
   }]
}
```
     * Content type - `application/json`
     * Query parameter - `type`
     * HTTP method - `GET`
     * HTTP headers - `Expires: ...`
                      `Cache-Control: ...`
     * HTTP method - `POST`
       * POST BODY - `title=<title>`
     * HTTP status code - `200` - `OK`, `400`, `500`
  * XML
```xml
<orders>
  <order>
     <id>1</id>
     <title>"1BC123A"</title>
     <customerId></customerId>
     <feed>"http://servername.com:8080/?name=order%20{id}"</feed>
  </order>
  <order>
     <id>2</id>
     <title>"1BA1111F"</title>
     <feed>"http://servername.com:8080/?name=order%20{id}"</feed>
     ...
  </order>
</orders>
```
     * Content type - `text/xml`
     * Query parameter - `type`
     * HTTP method - `GET`, `POST`
     * HTTP status code - `200` - `OK`, `400` - `Bad Request`, `500` - `Internal Server Error`

Tickets

* Creating a ticket
  * Protocol - `HTTP`
  * End Point - `http://servername.com:8080/tickets`
  * URI Design - `http://servername.com:8080/tickets`
 * Updating a ticket
  * Protocol - `HTTP`
  * End Point - `http://servername.com:8080/tickets`
  * URI Design - `http://servername.com:8080/tickets`
* Representation
  * JSON
```json
{ "tickets" :
  [ { "id" : 1BC123A,
      "flightNumber" : "SU123",
      "departureDate" : "27.02.2023 11:05"
      "from" : "Moscow"
      "to" : "Cancun"
```

~~~
      "seatClass" : "econom"
      "seatNumber" : "17F",
      "refundOption" : "true",
      "changeOption" : "true",
      "typeOfMenu" : "regular",
      "loyaltyProgramNumber" : "123123123",
      "insurance" : "true",
      "checkIn" : "true",
      "exit" : "1A",
      "dateAndTimeOfCheckIn" : "25.02.2023 11:05",
      "handLuggages" : [ "countOfOfHandLuggages" : "1",
                         "weightOfHandLuggageKg" : "12" ]
      "baggages" : [ "countOfOfBaggages" : "2",
                     "weightOfBaggageKg" : "23" ]
      "countOfTransfers" : "2",
      "transfers" : [ { "cityOfTransfer" : "Istanbul",
                        "durationOfTransfer" : "8:40"} ,
                      { "cityOfTransfer" : "Madrid",
                        "durationOfTransfer" : "5:10"} ]
      "passengers" : [
"http://servername.com:8080/order/{id}/ticket/{id}/customer/{id}",

"http://servername.com:8080/order/{id}/ticket/{id}/customer/{id}" ]
    },
    { "id" : 1BA1111F,
      "flightNumber" : "R133",
      "departureDate" : "29.02.2023 11:05"
      "from" : "Mexico"
      "to" : "Bogota"
      "seatClass" : "econom"
      "seatNumber" : "1D",
      "refundOption" : "false",
      "changeOption" : "false",
      "typeOfMenu" : "vegetarian",
      "loyaltyProgramNumber" : "0",
      "insurance" : "true",
      "checkIn" : "false",
      "exit" : "0",
      "dateAndTimeOfCheckIn" : "27.02.2023 11:05",
      "handLuggages" : [ "countOfOfHandLuggages" : "1",
                         "weightOfHandLuggageKg" : "12" ]
      "baggages" : [ "countOfOfBaggages" : "1",
                     "weightOfBaggageKg" : "23" ]
      "countOfTransfers" : "1",
      "transfers" : [ { "cityOfTransfer" : "Cancun",
                        "durationOfTransfer" : "1:10"} ]
      "passengers" : [
"http://servername.com:8080/order/{id}/ticket/{id}/customer/{id}" ]
    }]
}
```
     * Content type - `application/json`
     * Query parameter - `type`
     * HTTP method - `GET`
     * HTTP method - `PUT`
     * HTTP headers - `Expires: ...`
                      `Cache-Control: ...`
     * HTTP method  - `POST`
       * POST BODY - `title=<title>`
     * HTTP status code - `200` - `OK`, `400` - `Bad Request`, `500` - `Internal
Server Error`
  * XML
```xml
<tickets>
~~~

```
  <ticket>
     <id>1BC123A</id>
     <flightNumber>SU123</flightNumber>
     <departureDate>27.02.2023 11:05</departureDate>
     <from>Moscow</from>
     <to>Cancun</to>
     <seatClass>econom</seatClass>
     <seatNumber>17F</seatNumber>
     <refundOption>true</refundOption>
     <changeOption>true</changeOption>
     <typeOfMenu>regular</typeOfMenu>
     <loyaltyProgramNumber>123123123</loyaltyProgramNumber>
     <insurance>true</insurance>
     <checkIn>true</checkIn>
     <exit>1A</exit>
     <dateAndTimeOfCheckIn>25.02.2023 11:05</dateAndTimeOfCheckIn>
     <handLuggages>
      <countOfOfHandLuggages>1</countOfOfHandLuggages>
      <weightOfHandLuggageKg>12</weightOfHandLuggageKg>
     </handLuggages>
     <baggages>
      <countOfOfBaggages>2</countOfOfBaggages>
      <weightOfBaggageKg>23</weightOfBaggageKg>
     </baggages>
     <countOfTransfers>2</countOfTransfers>
     <transfers>
      <transfer>
        <cityOfTransfer>Istanbul</cityOfTransfer>
        <durationOfTransfer>8:40</durationOfTransfer>
      </transfer>
      <transfer>
        <cityOfTransfer>Madrid</cityOfTransfer>
        <durationOfTransfer>5:10</durationOfTransfer>
      </transfer>
     </transfers>
     <passengers>
       <passenger>"http://servername.com:8080/order/{id}/ticket/{id}/customer/
{id}"</passenger>
       <passenger>"http://servername.com:8080/order/{id}/ticket/{id}/customer/
{id}"</passenger>
     </passengers>
  </ticket>
    <ticket>
     <id>1BA1111F</id>
     <flightNumber>R133</flightNumber>
     <departureDate>29.02.2023 11:05</departureDate>
     <from>Mexico</from>
     <to>Bogota</to>
     <seatClass>econom</seatClass>
     <seatNumber>1D</seatNumber>
     <refundOption>false</refundOption>
     <changeOption>false</changeOption>
     <typeOfMenu>vegetarian</typeOfMenu>
     <loyaltyProgramNumber>0</loyaltyProgramNumber>
     <insurance>true</insurance>
     <checkIn>false</checkIn>
     <exit>0</exit>
     <dateAndTimeOfCheckIn>27.02.2023 11:05</dateAndTimeOfCheckIn>
     <handLuggages>
      <countOfOfHandLuggages>1</countOfOfHandLuggages>
      <weightOfHandLuggageKg>12</weightOfHandLuggageKg>
     </handLuggages>
     <baggages>
      <countOfOfBaggages>1</countOfOfBaggages>
```

~~~~
         <weightOfBaggageKg>23</weightOfBaggageKg>
        </baggages>
        <countOfTransfers>1</countOfTransfers>
        <transfers>
         <transfer>
           <cityOfTransfer>Cancun</cityOfTransfer>
           <durationOfTransfer>1:10</durationOfTransfer>
         </transfer>
        </transfers>
        <passengers>
          <passenger>"http://servername.com:8080/order/{id}/ticket/{id}/customer/
{id}"</passenger>
        </passengers>
  </ticket>
</tickets>
```
     * Content type - `text/xml`
     * Query parameter - `type`
     * HTTP method - `GET`, `POST`, `PUT`
     * HTTP status code - `200` - `OK`, `400` - `Bad Request`, `500` - `Internal
Server Error`


### Customers

* Creating a customer
  * Protocol - `HTTP`
  * End Point - `http://servername.com:8080/customers`
  * URI Design - `http://servername.com:8080/customers`
* Updating a customer
  * Protocol - `HTTP`
  * End Point - `http://servername.com:8080/customers`
  * URI Design - `http://servername.com:8080/customers`
* Deliting a customer
  * Protocol - `HTTP`
  * End Point - `http://servername.com:8080/customers`
  * URI Design - `http://servername.com:8080/customers`
* Customer subscribing to direction
  * Protocol - `HTTP`
  * End Point - `http://servername.com:8080/customers`
  * URI Design - `http://servername.com:8080/customers/{id}/direction/{id}`
* Representation
  * JSON
```json
{ "customers" :
  [ { "id" : 1,
      "firstName" : "Victoria",
      "lastName" : "Nasonova",
      "email" : "nasoviva@gmail.com"
      "documentType" : "passport"
      "documentNumber" : "511554453"
      "birthDay" : "02.11.1988"
      "phone" : "+529843672222"
      "orders" : [ "http://servername.com:8080/customer/{id}/order/{id}",
                   "http://servername.com:8080/customer/{id}/order/{id}" ]
    },
    { "id" : 2,
      "firstName" : "Roman",
      "lastName" : "Nasonov",
      "email" : "roman@gmail.com"
      "documentType" : "passport"
      "documentNumber" : "721554453"
      "birthDay" : "24.07.1980"
      "phone" : "+529841111111"
~~~~

~~~
      "orders" : [ "http://servername.com:8080/customer/{id}/order/{id}" ]
    }]
}
```
     * Content type - `application/json`
     * Query parameter - `type`
     * HTTP method - `GET`
     * HTTP method - `PUT`
     * HTTP method - `DELETE`
     * HTTP headers - `Expires: ...`
                      `Cache-Control: ...`
     * HTTP method  - `POST`
       * POST BODY - `title=<title>`
     * HTTP status code - `200` - `OK`, `400` - `Bad Request`, `500` - `Internal
Server Error`
  * XML
```xml
<customers>
  <customer>
     <id>1</id>
     <firstName>"Victoria"</firstName>
     <lastName>"Nasonova"</lastName>
     <email>"nasoviva@gmail.com"</email>
     <documentType>"passport"</documentType>
     <documentNumber>"511554453"</documentNumber>
     <birthDay>"02.11.1988"</birthDay>
     <phone>"+529843672222"</phone>
     <orders>
       <order>"http://servername.com:8080/customer/{id}/order/{id}"</order>
       <order>"http://servername.com:8080/customer/{id}/order/{id}"</order>
     </orders>
  </customer>
  <customer>
     <id>2</id>
     <firstName>"Roman"</firstName>
     <lastName>"Nasonov"</lastName>
     <email>"roman@gmail.com"</email>
     <documentType>"passport"</documentType>
     <documentNumber>"721554453"</documentNumber>
     <birthDay>"24.07.1980"</birthDay>
     <phone>"+529841111111"</phone>
     <orders>
       <order>"http://servername.com:8080/customer/{id}/order/{id}"</order>
     </orders>
  </customer>
</customers>
```
     * Content type - `text/xml`
     * Query parameter - `type`
     * HTTP method - `GET`, `POST`, `PUT`, `DELETE`
     * HTTP status code - `200` - `OK`, `400` - `Bad Request`, `500` - `Internal
Server Error`
~~~