Software Requirements Specification

for

Ultra Air

Version 1.0 approved

Prepared by:
Ivanov Timofey
Zarudko Ignat
Parhomenko Heorhi

BSU, MMF

17.05.2020

Minsk 2020

Contents

1. General information

Application "**Ultra Air**" - a service that provides the opportunity to buy tickets for the flight you need.

This site is intended for people who value their time and want to buy tickets for the aircraft of interest to them with maximum comfort.

Having bought a ticket, you will instantly receive it by e-mail, and you can also add it to your electronic wallet and you will not waste time searching for it in the future.

2. Application purposes

The application provides the following features:

1. Authorization and registration;
2. View all available tickets;
3. Selecting the ticket, you are interested in;
4. Online payment for tickets.

3. Types of users and subsystem functions

System users

№	User	Description
1.	Guest(unregistered user)	Can view available tickets, but can't buy them, can also register or login
2.	Client	Can view and order tickets
3.	Administrator	Can do all what Client does and also has the ability to control flights (add, delete).

4. Users and their description

4.2 Use case diagram
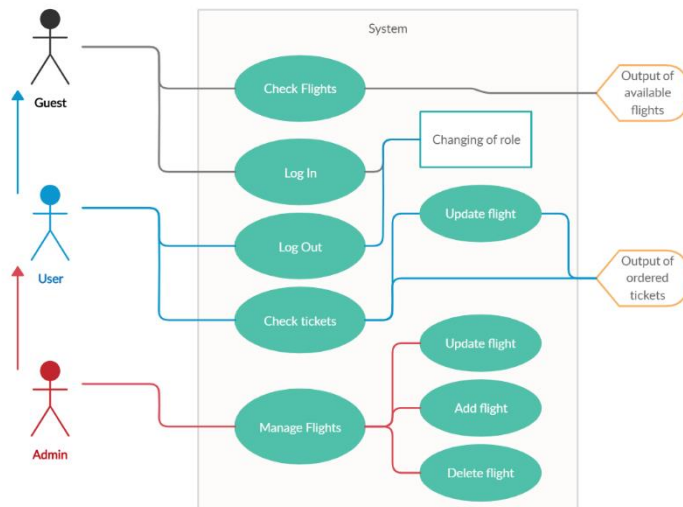
Pic. 4.1 Use case diagram

4.2 Use cases

A. Guest

Use Case – Register (Guest)

Briefly

Property	Description
Actor	Guest
Goal in context	Register in system
Trigger event	Need to register
Pre-condition	-
Success	New User is Registered
Fail	Refusal to register with reasons

Main scenario

Step	Actor	Description
1	Guest	Presses the "Register" button
2	System	Gives a registration form

3	Guest	Fills in the fields:
		1. Login
		2. E-mail
		3. Password
		4. Repeat password
		Press the "Submit" button
4	System	Validating fields. If success, new user is registered
5	System	Sends confirmation email

Extension

№	Variations	Possible variations
3.1	Required fields are empty	Application highlights empty fields, asks to fill them
3.2	Fields are filled incorrectly	Application highlights incorrect fields, offers to fix:
		a. Such login already exists
		b. This email doesn't exist/is already taken
		c. The re-password doesn't match the password itself

Use case – View air tickets (Guest)

Briefly

Property	Description
Actor	Guest
Goal in context	Choose a suitable ticket
Trigger event	View available tickets
Pre-condition	-
Success	Ticket found
Fail	Ticket not found

Main scenario

Step	Actor	Description
1	Guest	Presses the "Catalog" button
2	System	Redirects to a catalog page
3	Guest	Chooses a ticket
4	System	Redirects to a Login/Register page (only authorized user can order tickets)

B. Client
Use Case – Login (Client)

Briefly

Property	Description
Actor	Client
Goal in context	Login to the system
Trigger event	Need to login
Pre-condition	Need to book a ticket/View booked
Success	Client is authorized
Fail	Refusal to login with reasons

Main scenario

Step	Actor	Description
1	Client	Presses the "Login" button
2	System	Gives a login form
3	Client	Fills in the fields:
		1. Login or E-mail
		2. Password
		Press the "Login" button
4	System	Validating fields. If success, Client is authorized

Extension

№	Variations	Possible variations
3.1	Required fields are empty	Application highlights empty fields, asks to fill them
3.2	Fields are filled incorrectly	Application highlights incorrect fields, offers to fix:
		a. Login or E-mail and/or Password are incorrect

Use Case – View air tickets (Client)

Briefly

Property	Description
Actor	Client
Goal in context	Choose a suitable ticket
Trigger event	View available tickets
Pre-condition	-
Success	Ticket found
Fail	Ticket not found

Main scenario

Step	Actor	Description
1	Client	Presses the "Catalog" button
2	System	Redirects to a catalog page
3	Client	Chooses a ticket
4	System	Redirects to a ticketing page

Use Case – Ticketing (Client)

Briefly

Property	Description
Actor	Client
Goal in context	Order a ticket
Trigger event	Need to order a ticket
Pre-condition	Client is authorized
Success	Ticket is booked
Fail	Refusal of order with reasons/Cancelled by client

Main scenario

Step	Actor	Description
1	Client	Presses the "Order" button on concrete ticket
2	System	Gives a form with info about ticket
3	Client	Fills in the fields:
		1. If client want to be first on registration, sitting
5	System	Calculates ticket's price based on entered data
4	Client	Presses the "Order" or "Cancel" button
5	System	Confirms/Cancels booking

Use Case – View ordered tickets (Client)

Briefly

Property	Description
Actor	Client
Goal in context	View ordered tickets
Trigger event	Need to view ordered tickets
Pre-condition	Client is authorized
Success	Displaying ordered tickets
Fail	Error

Main scenario

Step	Actor	Description
1	Client	Presses the "My tickets" button at his profile page
2	System	Returns a list of ordered tickets

Extension

№	Variations	Possible variations
3.1	Ticket list is empty	Application returns a message "Empty list" instead of list itself

Use Case – Logout (Client)

Briefly

Property	Description
Actor	Client
Goal in context	Client becomes a Guest
Trigger event	Need to logout
Pre-condition	Client is authorized
Success	Client is logged out
Fail	Error

Main scenario

Step	Actor	Description
1	Client	Presses the "Logout" button at his profile page
2	System	Authorizes Client as a Guest

C. Admin
Use Case – Login (Admin)

Briefly

Property	Description
Actor	Client
Goal in context	Login to the system
Trigger event	Need to login
Pre-condition	Need to User abilities + flight management (Add, Delete, Edit)
Success	Admin is authorized
Fail	Refusal to login with reasons

Main scenario

Step	Actor	Description
1	Admin	Presses the "Login" button
2	System	Gives a login form
3	Guest	Fills in the fields:
		1. Login or E-mail
		2. Password
		Press the "Login" button
4	System	Validating fields. If success, Admin is authorized

Extension

№	Variations	Possible variations
3.1	Required fields are empty	Application highlights empty fields, asks to fill them
3.2	Fields are filled incorrectly	Application highlights incorrect fields, offers to fix:
		a. Login or E-mail and/or Password are incorrect

Use Case – Flight Management (Admin)

Briefly

Property	Description
Actor	Admin
Goal in context	Go to the flight management menu
Trigger event	Need to manage flights
Pre-condition	Admin is authorized
Success	Redirecting to Flight management page
Fail	Error

Main scenario

Step	Actor	Description
1	Admin	Presses the "Manage Flights" button
2	System	Returns flights list with buttons "delete" for each flight, and button "Add flight" at the end
3. Adding		
3.1	Admin	Presses "Add Flight" button
3.2	System	Returns adding flight form
3.3	Admin	Fills in the fields:
		a. Departure city
		b. Arrival city
		c. Departure date
		d. Arrival date
3.4		Presses "Add" button
4. Delete		
4.1	Admin	Presses "Delete Flight" on certain flight
4.2	System	Deletes certain flight
5. Edit		
5.1	Admin	Presses "Edit Flight" on certain flight
5.2	System	Transforms "Add flight" form into "Edit flight" form, fulfills fields with info about current flight
5.3	Admin	Edits fields
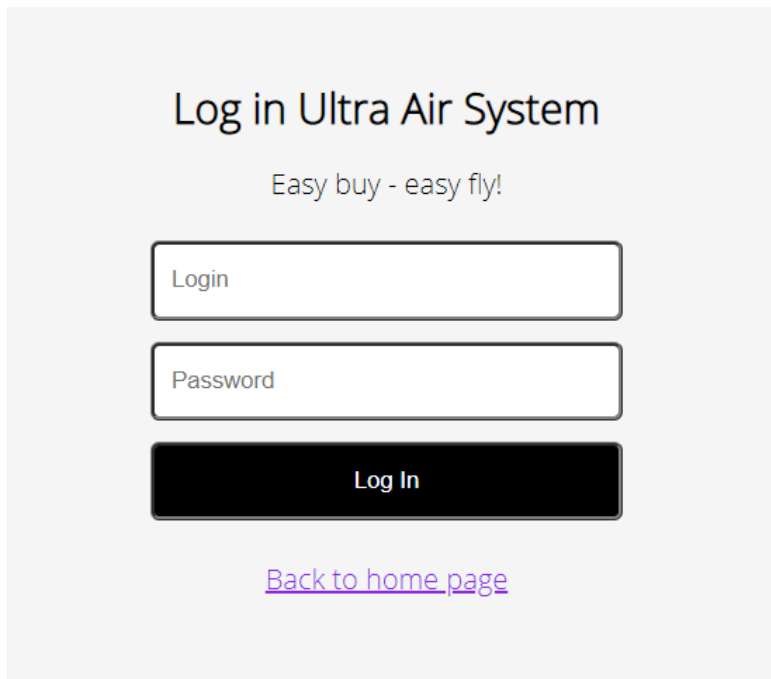
| 5.4 | Admin | Presses "Edit" button |
| 5.5 | System | Edit certain flight |

Extension

№	Variations	Possible variations
3.1/5.1	Required fields are empty	Application highlights empty fields, asks to fill them
3.2/5.2	Fields are filled incorrectly	Application highlights incorrect fields, offers to fix:
		a. City in unavailable for flights

5. Application Layouts:
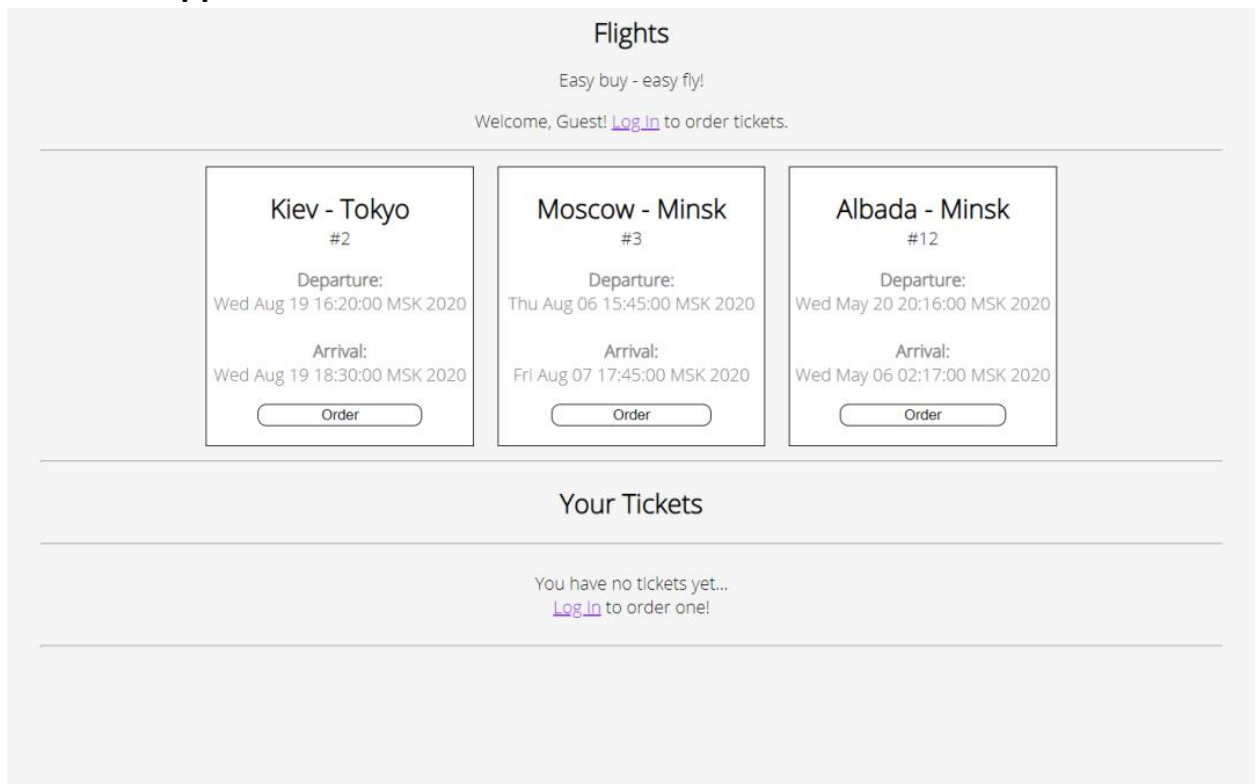
5.1 Application Login Page

Pic. 5.1

5.2 Application Home

Pic. 5.2

5.3 Filling in information when buying a ticket

Ultra Air

Ticket ordering

Username:

admin

Flight:

3

First on registration:

First on board:

Order

Back to home page

Pic. 5.3

5.4 Confirm ticket page

Ultra Air

Ticket ordering

New Ticket

Username = admin
Flight Num = 3
First on board = false
First on registration = false
Departure date = Thu Aug 06 15:45:00 MSK 2020

Ticket price = 210.0
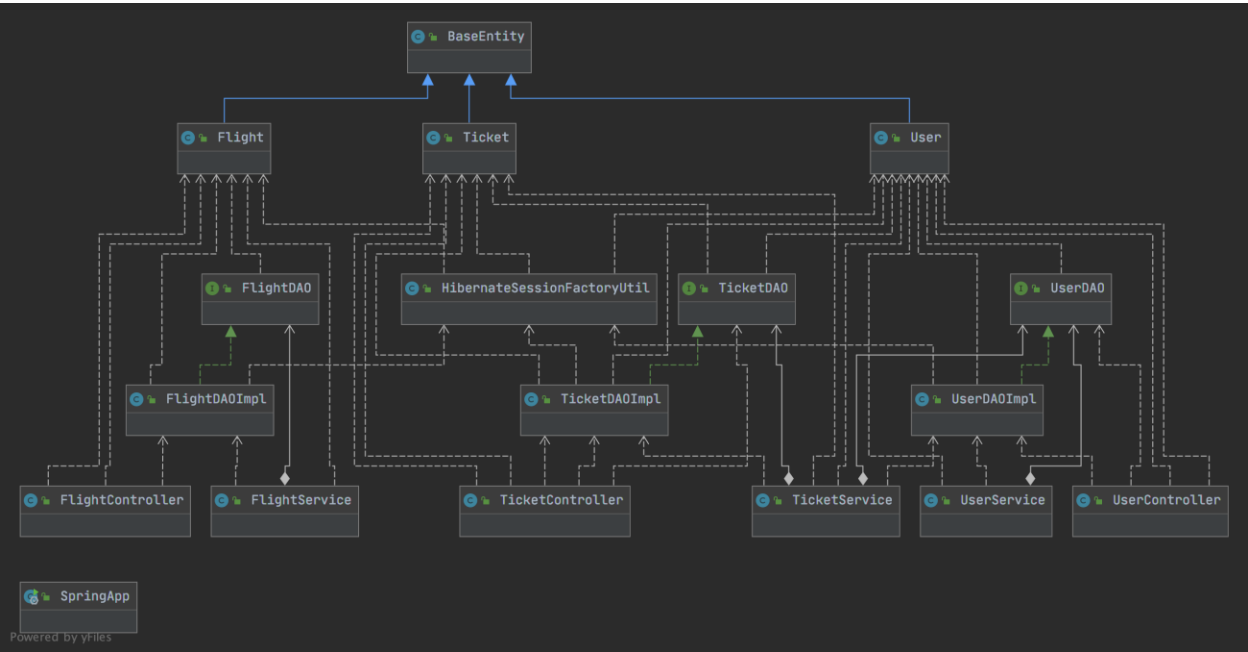
Order

Back to home page

Pic. 5.4

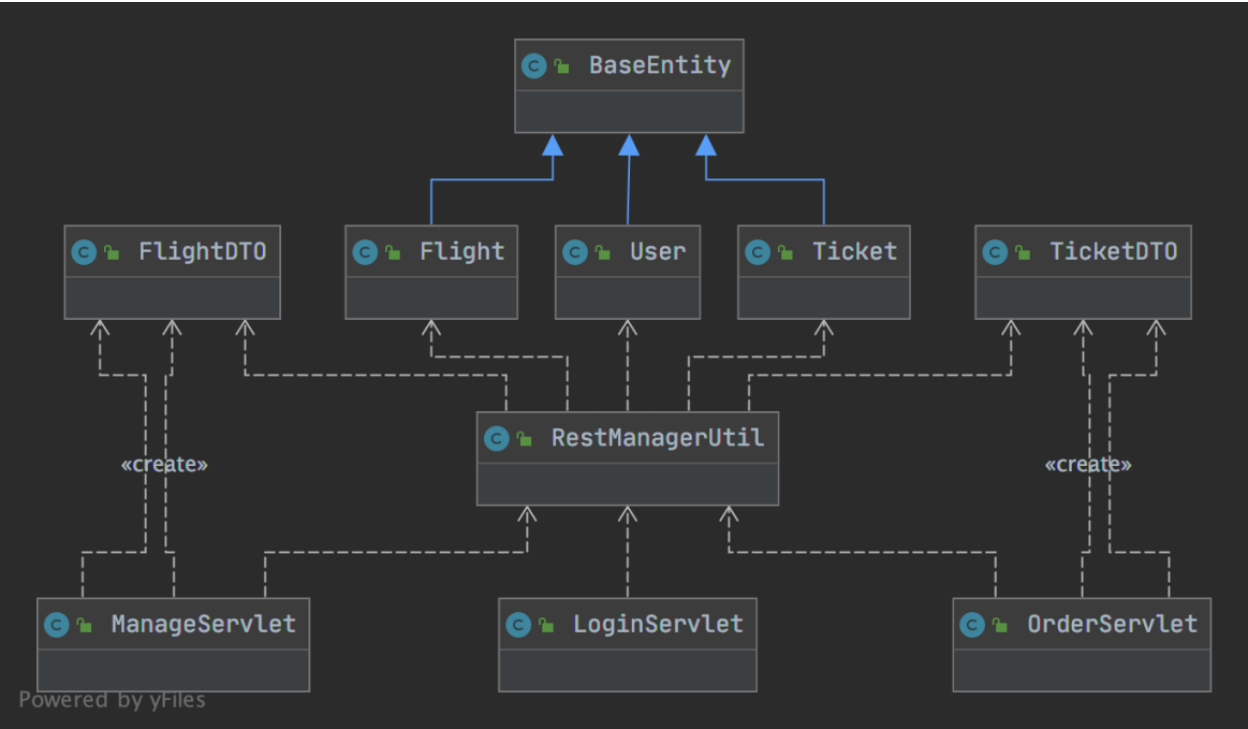## 5.5 Manage flights page

Pic. 5.5

6. System description

6.1 Backend class diagram

Pic. 6.1

6.2 Frontend class diagram

Pic. 6.2

7. Deadlines and execution plan

Development stage	Deadline
Basis	29.03.2020
Diagram	12.04.2020
Use Stories	12.04.2020
CI	19.04.2020
Unit Testing	19.04.2020
Design	26.04.2020
Clean Code	26.04.2020
UML	17.05.2020
Presentation	17.05.2020
Docs	17.05.2020