**Project Report**

**Fly High – Airline Management System**

**Created by:**

|  |  |
| --- | --- |
| **Cristina Ailoaei (266543)** | **Dragoș Sîrbu (266500)** |
|  |  |
| **Michał Jurewicz (266892)** | **Michał Podgórni (267128)** |
|  |  |

**Supervisors:**

**Ib Havn**

**Jens Cramer Alkjærsig**

**Mona Wendel Andersen**

**Steffen Vissing Andersen**

**[Name and logo of educational institution]**

**[Logo of companies included]**

**[Number of characters]**

**Information and Communication Technology** **Engineering**

**2nd Semester**

**IT-SEP2Y-A17**

**Group I**

**07.07.2018**

****

**Table of content**

[List of figures and tables v](#_Toc515579587)

[Executive summary vi](#_Toc515579588)

[1 Introduction 1](#_Toc515579589)

[2 Theory/literature survey 2](#_Toc515579590)

[3 Methods 3](#_Toc515579591)

[3.1 Analysis 4](#_Toc515579592)

[3.1.1 Requirements 4](#_Toc515579593)

[3.1.2 Use case diagram 4](#_Toc515579594)

[3.1.3 Use case descriptions 4](#_Toc515579595)

[3.1.4 Activity diagrams 4](#_Toc515579596)

[3.1.5 Analysis class diagram 4](#_Toc515579597)

[3.2 Design 4](#_Toc515579598)

[3.2.1 Design class diagram 4](#_Toc515579599)

[3.2.2 Sequence diagrams 4](#_Toc515579600)

[3.2.3 GUI design 4](#_Toc515579601)

[3.3 Implementation 4](#_Toc515579602)

[3.4 Testing 4](#_Toc515579603)

[3.4.1 Test cases 4](#_Toc515579604)

[3.4.2 Junit testing 4](#_Toc515579605)

[4 Results/findings and Discussion 5](#_Toc515579606)

[4.1 Results 5](#_Toc515579607)

[4.2 Discussion 5](#_Toc515579608)

[5 Conclusions c6](#_Toc515579609)

[6 Sources of information 7](#_Toc515579610)

Appendices

# List of figures and tables

# Executive summary

For content see Appendix 3 “Project Report – VIA Engineering Guidelines”.

# Introduction

For content see Appendix 3 “Project Report – VIA Engineering Guidelines”.

# Theory/literature survey

For content see Appendix 3 “Project Report – VIA Engineering Guidelines”.

# Methods

For content see Appendix 3 “Project Report – VIA Engineering Guidelines”.

## Analysis

### Requirements

#### Functional requirements

1. An administrator should be able to add airports to the system. While adding a new airport, the administrator has to specify the code, name, city, postcode, country, number of gates.
2. An administrator should be able to add airplanes to the system. While adding a new plane, the administrator has to specify the number, model, number of seats.
3. An administrator should be able to add crew members to the system. While adding a new crew member, the administrator has to specify the name, position, address, birthdate, id, phone number, e-mail.
4. An administrator should be able to add flights to the system. While adding a new flight, the administrator has to specify the number, departure time, arrival time, departure place, arrival place, plane, crew, price.
5. A customer booking a flight should specify all of the following: name, birthdate, nationality, type of ID, ID number, expiration date.
6. An administrator should be able to delete data from the system.
7. A head administrator should be able to cancel flights.
8. An administrator should be able to change data for club members, crew, flights, airplanes and airports.
9. An administrator should be able to select date/time range for flights in order to get flights in a specified range.
10. An administrator should be able to select cities for flights in order to get flights from/to the specified cities.
11. A customer should be able to choose a seat number, luggage size, payment method in order to book a ticket.
12. A customer should be able to select departure and destination airport and the departure and return date (or departure only) for flights in order to get the available flights.
13. An administrator should be able to get a list of all flights and club members.
14. An administrator should be able to set the annual fee for club members.
15. A customer should receive the ticket via email.
16. A customer should be able to become a club member in order to get discounts.
17. A club member should be able to search only for cheap flights from his/her city.
18. A customer should be able to subscribe to the newsletter in order to receive new information regarding flights and offers via email.
19. An administrator should be able to log in the system in order to manage data.
20. A head administrator should be able to see the profiles of all administrators.
21. A head administrator should be able to create or delete an administrator account in order to ease the management of accounts.

#### Non-functional requirements

1. The system has to use the client-server architecture.
2. The system has to store persistent data using a database.
3. The system has to have a GUI.
4. The system has to provide a log in.
5. The system has to be implemented in Java.
6. The system and the system development process have to be documented.

### Use case diagram

Figure 1 - Use case diagram

The use case shown above (Figure 1) presents all functional feature that every user of  
the FlyHigh application can perform. The use cases are following:

* **Add an element** – The administrator or the head administrator can add an airplane, airport, crew member or flight into the system.
* **Find an element** – The administrator or the head administrator can search for an existing airplane, airport, crew member or flight in the system.
* **Edit an element** – The administrator or the head administrator can edit the data of an airplane, airport, crew member or flight.
* **Delete an element** – The head administrator can delete an airplane, airport or crew member from the system.
* **Cancel a flight** – The head administrator can cancel a flight.
* **Find a flight** – The customer can search for an existing flight in the system.
* **Book a flight** – The user can book an existing flight.

### Use case descriptions

|  |  |
| --- | --- |
| UseCase | Book a flight |
| Summary | A customer books a flight |
| Actor | Customer |
| Precondition | None. |
| Postcondition | The flight becomes booked, the changes are stores in the database. |
| Base Sequence | 1. The person goes through find a flight use case.  2. The person enters all the required personal data: name, birthdate, nationality, type of ID, ID number, expiration date, seat number, size of luggage, method of payment.  3. The person confirms the decision to book the given flight.  4. If one or more of the entered data is not valid then go to step 2 else the decision is confirmed and the given flight becomes booked, changes are stored in the database, person is redirected to another site in order to make a payment and the use case ends. |
| Branch Sequence |  |
| Exception Sequence | The entered data could not be valid:  4 as base sequence  The system informs that the entered data is not valid |
| Sub UseCase | Find a flight |
| Note |  |

Figure 2 - Book a flight use case description

### Activity diagrams

Figure 3 - Booking a flight activity diagram

One of the crucial functionalities, booking a flight, is

### Analysis class diagram

## Design

### Design class diagram

### Sequence diagrams

### GUI design

## Implementation

## Testing

### Test cases

### Junit testing

# Results/findings and Discussion

## Results

## Discussion

# Conclusions

For content see Appendix 3 “VIA Engineering - Project Report Guidelines”

# Sources of information

For content see Appendix 3 “VIA Engineering - Project Report Guidelines”

**Appendices**

For content see Appendix 3 “VIA Engineering - Project Report Guidelines”