

Group Report

Assessed Coursework



Contributors:

Maha Albalushi – H00245141

Nicolas Wiecek H00156036

Gita Permonaite H00260666

Abdelraman Yaseen – H00281389

CONTENTS

1. Introduction

2. Requirements’ checklist

3. Design Considerations

4. UML class diagrams

5. Data structures

6. Functionality decisions

7. Testing

8. Summary

9. Appendixes

10. Code screenshots

1. **INTRODUCTION**

The purpose of this report is to develop and present simple check-in system at an airport. At this stage we only will develop the basic functionality of this system.

This report will go through development process of this program by presenting data structures, UML diagrams, decisions about program functionality and testing.

At the end of this report all members will be presented with their contribution to this coursework.

In our group we have these members:

* Maha Albalushi – H00245141

Worked on the Flight class and class diagram.

* Nicolas Wiecek – H00156036

Worked on CheckinView and CheckinController as well as on activity diagram.

* Gita Permonaite – H00260666

Worked on CheckinModel as well as on data structures on report.

* Abdelraman Yaseen – H00281389

Worked on Passenger class as well as on test class and excess baggage calculations.

We all worked together on the final report.

1. **REQUIREMENT'S CHECKLIST**

The following table will list of all delivered requirement for this check in system:

|  |  |
| --- | --- |
| Functional requirements | Delivered: Yes/No/Partly Delivered |
| Text file1 with all bookings including (name of passenger, booking reference code, flight code and status of check-in) | Yes |
| Text file2 with all flight details (destination airport, the carrier, capacity of flight, maximum number of passengers, maximum baggage weight, maximum baggage volume) | Yes |
| GUI display representing electronic check-in kiosk | Yes |
| Option to enter passenger’s last name and booking reference code | Yes |
| Baggage dimensions and weight insertion fields for only 1 luggage | Yes |
| Excess baggage fee calculation | Yes |
| Passenger’s last name and booking ref number checked in against flight booking list | Yes |
| GUI kiosk closes after all passengers checks in | Yes |
| Exceptions thrown if passenger was not found on the list | Yes |
| Report generation | Yes |

1. **DESIGN CONSIDERATIONS**

For this application we decided to use the current classes, Passenger, PaseengerList, Name, Manager, Flight, FlightList, CheckInModel, CheckInView and CheckInController, Test, GUI

However, after starting coding we were also considering adding an additional Luggage class, but finally we agreed not to at the moment. Other than that, our program met the required specification.

1. **UML CLASS DIAGRAMS**

Please see our UML class diagrams as appendixes below:

* Appendix 1 – class diagram
* Appendix 2 – activity diagram
* Appendix 3 – sequence diagram

1. **DATA STRUCTURES**

For this program, we used array list as data structure in the classes below: plz add where lists are added

We used ArrayList because they support dynamic arrays, so that they can be expanded automatically and if objects are removed, lists cab be shrinked.

1. **FUNCTIONALITY DECISIONS**

Our group had a few discussions regarding functionality for this system.

We already mentioned that our application has met required specifications for this system in section 2.

* Format for booking reference codes

2 alphabetical letters + 3 numbers ???

* Details about calculating the fees for access baggage

please add info

1. TESTING

Please explain which methods have been tested using Junit?

Exceptions used:

1. SUMMARY

For the stage 1 of this coursework we were discussing different groups of classes and their data types.

Therefore, we finally agreed on our development plan by week 4.

We divided all tasks between us and development process went well with a few issues we had while coding.

For example: agreeing on the right code for baggage calculation and

Overall our system met all required specifications

**APPENDIXES**

Appendix1:

Appendix2:

Appendix3: