

# Group Contribution

The members of the group consisted of Nadia Abulhawa, Haikah Ghoghari, Jamie Hill, Amy McFarland and Sam Muir.

In respect to the coding aspect of the group project, each member was assigned particular classes which was exhibited in the group planning schedule in the development report. Jamie was assigned the File IO and report generator classes, Nadia was assigned the flight classes, Haikah coded the passenger and baggage classes, Sam was assigned the Booking class and Amy was designated the GUI classes. Throughout this part of the project, each group member assisted other members in their classes and testing which proved beneficial in completing the classes on schedule. Jamie offered assistance with the testing and flight classes and providing advice for other members classes. Nadia, Amy and Haikah worked together frequently helping each other complete their classes. Sam also provided assistance when it was required. Additionally, every group member read through each class that was committed to GitHub. This ensured that before each class was committed the group was happy that the classes included everything required. The integration of all the code was completed by \*\* which involved correcting any bugs that occurred in the program. After reading the report and ensuring the code worked effectively, the submission was done as a group to ensure everyone was satisfied with the work that had been accomplished.

As a group, we worked effectively together and often met to assist other members with their classes and discuss the objectives that had to be fulfilled for the next meeting. Each group member worked and contributed equally to the project.

In regard to the group report, each section was completed as a group with each member contributing equally. Amy wrote the summary and group contribution section of the report which was checked by the entire group. The class diagram restructuring was discussed as a group and individually completed by Nadia. The data structures section of the report was completed by Amy and checked by the group. The program functionality decisions were discussed a group during the coding part of the project and written by Amy. Finally, the testing section was completed as a group. Before submission, the report was read by each group member to ensure everyone was satisfied with the completed work.

# Repository Link

https://github.com/jamiefhill/hw-ase-stage-one-check-in-system

# Status Report

This program meets the specification fully.

# Class Diagram

Activity Diagram (for calculating the excess baggage fees)

# Data Structures

The data structures that were discussed in the development report have not changed and the decisions on which structures to use have remained the same.

The booking collection class used a HashMap structure, for the same reasons that were provided in the development report. For example, it would allow the finding of a booking quickly for the user when their booking code was entered into the GUI. This proved beneficial and effective as it did not disrupt the flow of the program for the user.

The flight collection class implemented a TreeSet structure as it provided the optimum functionality for the final output report as it iterated through the set of flights. Additionally, it offers a sorted list of flights for the report.

# Program Functionality

In regard to the functionality of the program, the booking reference codes followed a structure of two letters followed by three numbers with a – and then three numbers. For example, a booking coded used in the program was **BA123-121.** The excess baggage fees were calculated by firstly, subtracting the weight that was inputted by the user in the GUI from the allowed baggage weight per passenger. The allowed baggage weight per passenger value is included in the flight csv file and remains a constant number. This provided the excess baggage weight per passenger. The excess baggage weight was multiplied by the excess charge to calculate the excess fees. The excess charge value exists in in the flight csv file and fluctuates per flight.

# Testing

JUnit testing was completed for the