Development Plan

Syd Colvin, Vanesa Alonso, Sheron Tong & Enrique Espinosa

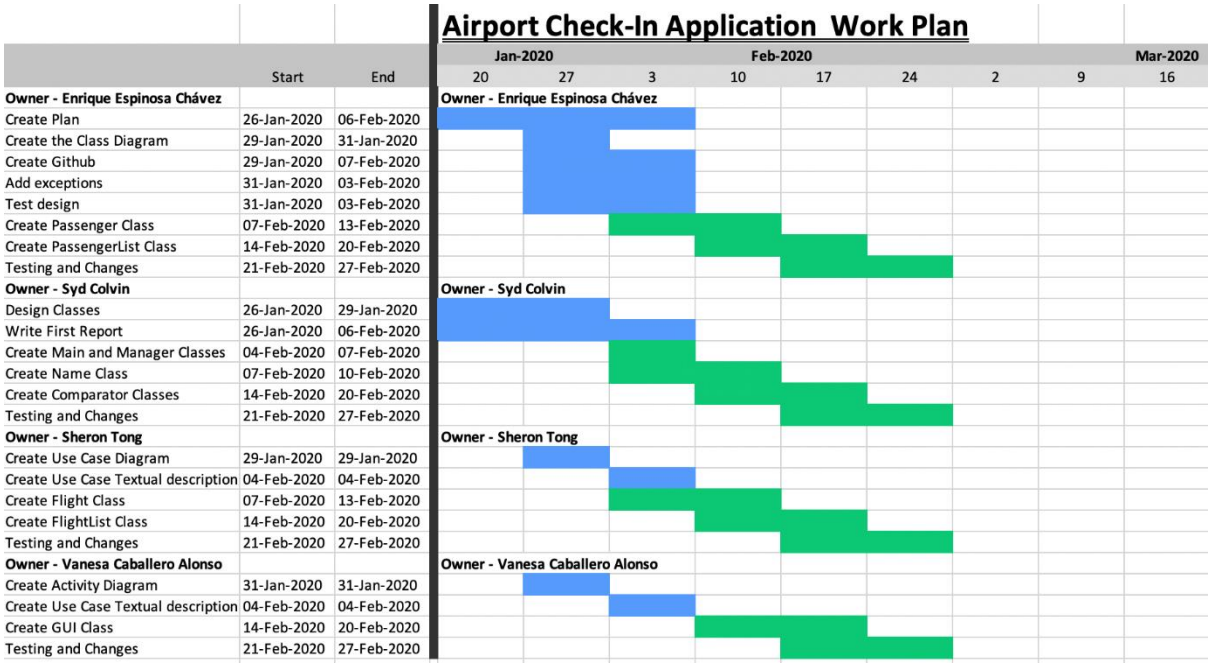
Chosen Data Structures

The Passengers and Flights will be created by there own classes to become objects, which are then stored in their respective container. Details such as baggage weight and volume and fees, etc, will be initialised as zero with their values modified when relevant. Check in will be initialised as false.

We plan to store the Passenger list as a HashSet data type. Passengers should allow duplicates as there is no unique value, multiple passengers can be on the same booking, and passengers can have multiple bookings (e.g. return flights). The key will be the booking reference, and the value the full name of the passenger.

The Flights will be stored as a TreeSet because it is secure against duplicates and attempting to re-add an existing flight will not change anything. Sets are also known for fast searching, which is ideal for an interface which will be used with a queue of waiting users.

Work Plan



	Start	End
Owner - Enrique Espinosa Chávez		
Create Plan	26-Jan-2020	06-Feb-2020
Create the Class Diagram	29-Jan-2020	31-Jan-2020
Create Github	29-Jan-2020	07-Feb-2020
Add exceptions	31-Jan-2020	03-Feb-2020
Test design	31-Jan-2020	03-Feb-2020
Create Passenger Class	07-Feb-2020	13-Feb-2020
Create PassengerList Class	14-Feb-2020	20-Feb-2020
Testing and Changes	21-Feb-2020	27-Feb-2020
Owner - Syd Colvin		
Design Classes	26-Jan-2020	29-Jan-2020
Write First Report	26-Jan-2020	06-Feb-2020
Create Main and Manager Classes	04-Feb-2020	07-Feb-2020
Create Name Class	07-Feb-2020	10-Feb-2020
Create Comparator Classes	14-Feb-2020	20-Feb-2020
Testing and Changes	21-Feb-2020	27-Feb-2020
Owner - Sheron Tong		
Create Use Case Diagram	29-Jan-2020	29-Jan-2020
Create Use Case Textual description	04-Feb-2020	04-Feb-2020
Create Flight Class	07-Feb-2020	13-Feb-2020
Create FlightList Class	14-Feb-2020	20-Feb-2020
Testing and Changes	21-Feb-2020	27-Feb-2020
Owner - Vanesa Caballero Alonso		
Create Activity Diagram	31-Jan-2020	31-Jan-2020
Create Use Case Textual description	04-Feb-2020	04-Feb-2020
Create GUI Class	14-Feb-2020	20-Feb-2020
Testing and Changes	21-Feb-2020	27-Feb-2020

Planned Testing

Planned Testing

Our tests will include:

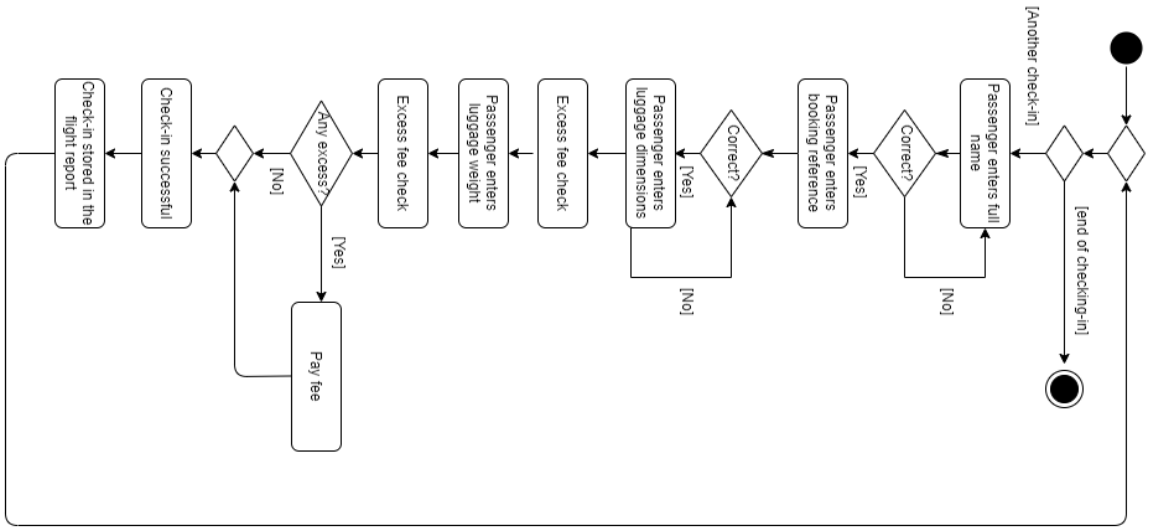
- The correct reading of the input files for both passengers and flights and the respective line processing
- The GUI works correctly while searching for passengers' last names and booking reference
- The baggage weight and size can be entered in the respective fields
- The volume calculation is done correctly
- The fees for overweight and oversized baggage are calculated correctly
- The final report is generated with all the required information

Exceptions

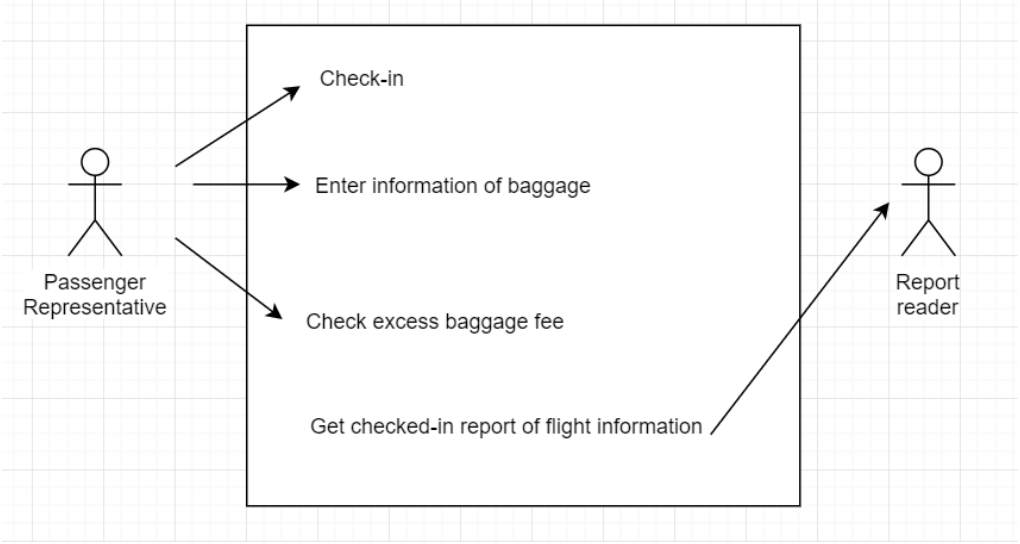
- File Exceptions:
  - If any files are missing, the GUI won't run and will show an error instead
  - If any of these events happen, the application will show an error and skip the lines:
    - Unique booking reference code is not a number
    - Flight capacity is not a number
- In the GUI:
  - If a passenger is not found, the GUI asks to re-enter details.
  - If the baggage weight and size details are not a number, the GUI will ask to enter them correctly.
  - If the passenger doesn't accept the calculated fees, then the passenger won't be able to check in.
  - If customer enters not valid payment details, user is informed of error, tries again.
  - If flight is closed, check-in is refused, message is sent to customer.
  - If customer has already checked in, the GUI will show warning.

Diagrams

Activity Diagram



Use Case Diagram



USER CASE: Check in
GOAL: To check in an already booked flight
ACTORS: Customer, Staff

Pre-conditions: Customer has already bought the flight ticket and has a booking reference.
Main success scenario:

1. Customer enters his/her full name and booking reference
2. Customer can view the booked flight reference
3. Customer enters dimensions and weight of his/her luggage and submit them
4. The customer has successfully checked in
5. The check in is stored in the list of checked-in flights and staff can access it
6. Staff get the checked-in report for each flight

Post-conditions: The check-in has been added to the list of already checked-in flights

Class Diagram