Client	Airline Company
User	Airline Employees
Functional requirements	The system must allow to: R1. Passenger loading system. R2. Check-in at the boarding lounge. R3. Prioritization of the first class. R4. Order of entry to the aircraft. R5. Aircraft departure order.
Context	The airline needs to improve the process of passenger check-in and check-out, so it is necessary to develop a first version of a system to load passenger information, register their arrival at the boarding lounge, establish an order of entry and exit of the aircraft, and prioritize the entry of first-class passengers. In addition, it must be considered that the database will be simulated by means of a plain text file and that the system must be efficient to handle large amounts of data.
Non-functional requirements	• Develop the project using TDA. • Using a repository from day 1. • Make at least 10 commits spread equip-temporally. • For each of the 10 commits, report 3 simple quality indicators in the repository readme (bug-density, reliability, and completeness). The completion indicator is expected to be greater than 5 at the end of the product implementation. • The system must be able to handle large amounts of data efficiently and quickly. • The system must be secure and protect passenger information from unauthorized access. • The system must be scalable to handle many flights and passengers in the future. • The system must be easy to use and understandable for the airline personnel in charge of operating it.

<table>
<tr><td>Name or identifier</td><td colspan="3">R1. Passenger loading system.</td></tr>
<tr><td>Summary</td><td colspan="3">The system shall allow the uploading of passenger information for a particular flight.

The uploading of passenger information shall be done through a user-generated plain text file.</td></tr>
<tr><td rowspan="3">Inputs</td><td>Input name</td><td>Datatype</td><td>Selection or repetition condition</td></tr>
<tr><td>dataBase</td><td>String</td><td>If there is an existing database with correctly registered passengers</td></tr>
<tr><td>flight_Number</td><td>String</td><td>If it is assigned a correct flight number</td></tr>
<tr><td>General activities necessary to get the results</td><td colspan="3">1. Read the plain text file with passenger information.

Validate that the plain text file has the correct format and that it contains the necessary information to load the passengers.

3. Create a "Flight" object and assign it the corresponding flight number.

4. For each passenger in the plain text file, create a "Passenger" object with the corresponding information and add it to the "Flight" object.

5. Return the "Flight" object with the loaded passenger list.</td></tr>
</table>

Result or postcondition	A "Flight" object with the passenger list loaded.		
Outputs	Output name	DataType	Selection or repetition condition
	flight	Flight	In case the plain text file is incorrectly formatted or missing necessary information, an exception must be thrown indicating the error (Type: Exception).

Name or identifier	R2. Check-in at the boarding lounge.
Summary	The system must be able to search for a passenger's complete information once he/she arrives at the corresponding boarding lounge. The system must register the arrival of a passenger at the boarding lounge. The system must allow rewarding the punctuality of passengers by entering the aircraft on a first-come, first-served basis.

Inputs	Input name	Datatype	Selection or repetition condition
	namePassenger	String	If the data is recorded correctly
	flightNumber	Int	
	timeArrival	LocalDateTime	
General activities necessary to get the results	1. Complete passenger information is searched in the system database. 2. The time of arrival at the boarding lounge is recorded in the system database.		
Result or postcondition	The system must record the passenger's arrival time at the boarding lounge corresponding to the flight indicated.		
Outputs	Output name	DataType	Selection or repetition condition
	None		

Name or identifier	R3. Prioritization of the first class.

Summary	The system must consider special data such as mileage accrual, special care required, senior citizenship or other relevant data to prioritize the admission of first-class passengers.		
Inputs	Input name	Datatype	Selection or repetition condition
	list	PassengerList	If they have been properly registered
General activities necessary to get the results	1. The method goes through the list of passengers and evaluates whether they meet any special conditions to prioritize their entry to the aircraft. 2. A priority order is established based on the special conditions and the passenger list is sorted accordingly. 3. The sorted passenger list is returned.		
Result or postcondition	The list of passengers ordered according to the special conditions that give them priority access to the aircraft.		
Outputs	Output name	DataType	Selection or repetition condition
	list	PassengerList	If the list entered complies with the required parameters.

<table>
<tr><td>Name or identifier</td><td colspan="3">R4. Order of entry to the aircraft.</td></tr>
<tr><td>Summary</td><td colspan="3">The system must show the crew member in charge in which order passengers must enter the aircraft, following the established order.

The order in which passengers enter the aircraft must follow the call by sections of the aircraft, starting with those furthest from the entrance door to the one closest to it.</td></tr>
<tr><td rowspan="4">Inputs</td><td>Input name</td><td>Datatype</td><td>Selection or repetition condition</td></tr>
<tr><td>sections</td><td>ListOfSections</td><td rowspan="3">If they were registered with the correct parameters in the system.</td></tr>
<tr><td>list</td><td>PassengerList</td></tr>
<tr><td>featuresPassengers</td><td>String</td></tr>
<tr><td>General activities necessary to get the results</td><td colspan="3">1. An empty list is created to store the order in which passengers enter.

2. Iterate over the aircraft sections, starting with the one furthest from the entry door and ending with the one closest to the entry door.

3. For each section, passengers are iterated over in the corresponding order of entry and added to the list in step 1.

4. Priorities or special conditions, if any, are applied to modify the order of entry for some passengers. For example, if there are passengers with special needs, they are given priority entry.

5. The final list is returned with the order in which passengers are admitted.</td></tr>
</table>

Result or postcondition	A list of passengers in the corresponding order of entry to the aircraft, following the order established by sections and applying the priorities or special conditions.		
Outputs	Output name	DataType	Selection or repetition condition
	passengerListInOrder	PassengerList	If they were registered with the correct parameters in the system.

Name or identifier	R5. Aircraft departure order.		
Summary	The system must allow to establish the order of departure from the aircraft. The order of departure from the aircraft must be established by rows, where those who leave first are those who are in the first rows and for each row the order is established by proximity to the aisle or order of arrival as the last instance. The flight crew person in charge of the system will be able to see in which order the passengers must exit.		
Inputs	Input name	Datatype	Selection or repetition condition

	list	PassengerList	Must be pre-loaded
	numberOfRows	int	Must be a positive integer
	NumberOfSeatsPerRow	int	Must be a positive integer
General activities necessary to get the results	1. The passenger list is sorted by priority, considering special data such as accumulated miles, special care required, senior citizens or other relevant data to prioritize the entry of first-class passengers. 2. An empty list is created to add passengers in the order in which they must depart. 3. The list of passengers is scrolled through, and passengers are added to the empty list according to the established order: passengers from the first rows are added first, and for each row, the one closest to the aisle is added first, then the one furthest from the aisle. 4. The list of passengers is displayed in the order in which they must leave.		
Result or postcondition	The list of passengers is shown in the order in which they must exit the aircraft, according to the order established by rows and proximity to the aisle.		
Outputs	Output name	DataType	Selection or repetition condition
	list	PassengerList	If it is correctly registered