Identification of the problem:

In general, the problem relates to the organization and management of data and processes to ensure efficient and satisfactory service for the airline's customers. The problem focuses on how to load passenger data into the system, how to search and record information of passengers arriving at the boarding lounge, and how to sort passengers for boarding the aircraft. In addition, there is an additional problem of how to handle first class passengers in a special way at the time of boarding.

gathering the necessary information:

|  |  |
| --- | --- |
| CLIENT | airline |
| USER | airline administrator |
| FUNCTIONAL REQUIREMENT | RF1: load passenger information.  RF2: search passenger information  RF3: register passenger arrive  RF4: show order entry of the passengers  RF5: assign priority |
| CONTEXTO DEL PROBLEMA |  |
| NO FUNCTIONAL REQUIREMENT | RNF1: The system must be efficient in searching for passenger information, even when the amount of data is significantly very large.  RNF2: The system must be easy to use for the crew member in charge.  RNF3: The system must be secure and reliable, guaranteeing the privacy of passenger information.  RNF4: The system must be scalable,  permitiendo la adición de nuevas funcionalidades en el futuro.  RNF5: The system should be developed using good programming practices and industry standards to ensure its quality and maintainability. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF1: load passenger information. | | |
| Summary | The system must be able to load passenger information for a flight by means of a plain text file. | | |
| Inputs | **input name** | **Datatype** | **Selection or repetition condition** |
| data | txt (String) |  |
| General activities necessary to obtain the results | 1. Load the txt file through the loadData() method.  2. Read the lines of the txt file (each line corresponds to the information of a passenger).  3. Create the passengers in the program using the information read.  4. Add the passengers to the hashtable.  5. Print the information read. | | |
| Result or postcondition | New passenger loaded or error message | | |
| Outputs | **output name** | **Datatype** | **Selection or repetition condition** |
| message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF2: search passenger information | | |
| Summary | The system must be alble to efficiently search for complete passenger information once a passenger arrives at the boarding lounge. | | |
| Inputs | **input name** | **Datatype** | **Selection or repetition condition** |
| key | K |  |
| General activities necessary to obtain the results | 1.type the key in the getValue() method  2. the system compares all the keys in the information table to see if it matches any of them.  3a. if there is a match the system will return the corresponding V value of the key K.  3b.if there is no match, the system will display a "not found" message. | | |
| Result or postcondition | 1. the value of the corresponding key 2. not found message. | | |
| Outputs | **output name** | **Datatype** | **Selection or repetition condition** |
| a. value | V |  |
| b. message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF3: register passenger arrive | | |
| Summary | The system must be able to record the arrival of a passenger at the boarding lounge. | | |
| Inputs | **input name** | **Datatype** | **Selection or repetition condition** |
| dateArrave | date |  |
| key | K |  |
| General activities necessary to obtain the results | 1. the system use the searchPassanger() method to check if the passenger exists.  2. the system calculate the passenger's arrival time  3. the system boards passengers on a first-come, first-served or first-served basis.  4. systeme shows basic information of the passenger as they go along | | |
| Result or postcondition | basic passenger information for boarding the aircraft | | |
| Outputs | **output name** | **Datatype** | **Selection or repetition condition** |
| info | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF4: load passenger information. | | |
| Summary | The system must be able to show the crew member in charge in which order passengers should enter the aircraft. | | |
| Inputs | **input name** | **Datatype** | **Selection or repetition condition** |
| option | int |  |
| General activities necessary to obtain the results | 1. The user select the option of the method showPassengerListEntry()  2. The system acquires passenger information and sorts it into a list.  3. Create the passengers in the program using the information read.  4. The system displays this sorted list on the console | | |
| Result or postcondition | passenger entry list | | |
| Outputs | **output name** | **Datatype** | **Selection or repetition condition** |
| list | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF5: assign priority | | |
| Summary | The system should allow priority to be given to first class passengers based on special attention required, age or other relevant data. | | |
| Inputs | **input name** | **Datatype** | **Selection or repetition condition** |
| passengerInfo | txt (String) |  |
| General activities necessary to obtain the results | 1. System compare attributes of the passenge info to assignet e priority  2. depending on which priority characteristics it meets the program will assign it a priority number | | |
| Result or postcondition | number passenger priority. | | |
| Outputs | **output name** | **Datatype** | **Selection or repetition condition** |
| priority | int |  |

+

SEARCH FOR CREATIVE SOLUTIONS:

Integrate the system with the airline's databases to obtain additional passenger information.

1. Saving  passenger’s information using an two arraylist,then ordering the arraylist using an algorithm of sorting to compare them by priority represented as a number result of the sum of  numerics values assigned  to the priority characteris like being first class,pregnant.While being selection,time of arrive,etc
2. Develop a computer system that allows for the loading of passenger information and registration of their arrival at the boarding gate, as well as prioritize the order of boarding and deplaning based on various criteria such as accumulated miles, special needs, elderly passengers, among others. This will require a database and the development of an efficient search and sorting algorithm.
3. Design a passenger identification system using facial recognition or fingerprint technology, which would allow for a faster and more secure registration process.
4. Implement a passenger notification system through a mobile application, which would allow passengers to be notified in real-time of the boarding process and their designated boarding time based on the order established by the system.
5. Provide training to boarding gate staff to effectively operate the system and improve efficiency in the boarding and deplaning process.
6. Create an online feedback system for passengers to evaluate the system's efficiency, allowing for the identification of continuous improvement opportunities.

TRANSITION FROM IDEA FORMULATION TO PRELIMINARY DESIGNS

one of the most viable options, taking into account the knowledge acquired in the university classes, would be to

Develop a simple computer system using a programming language such as Python or Java, which allows the loading and registration of passenger information, as well as the prioritization of the order of entry and exit of the aircraft according to the established criteria.

Since we could make use of all the algorithms structures seen in the following sections, which allow us to make the indicated ordering of passengers by priority, we could develop a simple computer system using a programming language such as Python or Java.