Engineering Method

Problem solving application.

Airline

Members:

- Alexis Jaramillo (A00395655)
- David Molte (A00368867)
- Juan Daniel Reina (A00394352)

Problem Context: The airline needs to improve the process of passenger boarding and deplaning to ensure a more efficient and comfortable travel experience for its customers. To this end, it has been requested to develop a system that allows the loading of passenger information, the identification and registration of their arrival at the boarding lounge, the establishment of an order of entry and exit of the aircraft, and the prioritization of the entry of first-class passengers. The system must be efficient, secure, scalable, flexible, reliable, and easy to use and maintain.

Solution development: To solve this problem, we will use a series of steps to follow, which will help us to have a systematic, effective, and efficient process to solve the need raised, which are:

1. Identification of the needs and symptoms of the problem: We can infer and inquire deeply into the problem that is presented to us, so we can highlight what are the aspects that are present in our case, we can lead to an effective solution:

Some of the needs and symptoms of the problem that can be identified in the statement are:

- Need to improve the efficiency of the process of getting passengers on and off the plane.

- Need to reward the punctuality of passengers entering the aircraft on a first-come, first-served basis.

- Need to prioritize the entry of first-class passengers according to certain special criteria.

- Symptom of long waiting times in the boarding lounge due to lack of order and efficiency in the process.

- Symptom of confusion and disorganization in the check-in and check-out process.

- Symptom of lack of clear information for the airline personnel in charge of operating the process.

- Symptom of lack of integration of the airline's systems and tools for information exchange.

2. Compilation of the information: For greater clarity in our search for a solution, we must have cleared all the elements that we are going to use, within which they can be:

- Data structures: they are ways of organizing and storing data in a computer in an efficient way so that they can be manipulated and recovered easily. Some of the data structures that can be used in this problem are arrays, linked lists, queues, stacks, trees, and graphs.

- Algorithms: are a series of instructions that are used to solve a specific problem. Some of the algorithms that can be used to solve the mentioned problem are sorting algorithms, search algorithms, tree and graph traversal algorithms, dynamic programming algorithms and recursion algorithms.

- Recursion: is a programming technique in which a function calls itself to solve a problem. It is a useful technique for solving problems that can be broken down into smaller, identical problems.

- Sorting: is the technique of arranging a set of data in a specific order. Sorting algorithms are used to arrange data so that it can be searched and retrieved efficiently.

- Searching: is the technique of finding a specific element in a data structure. Search algorithms are used to search for elements in a data structure and determine whether they are present or not.

- Trees: are data structures used to represent hierarchies of information. A tree is made up of nodes and edges that connect the nodes. Trees can be used to represent relationships between data and to organize and store information.

- Networks: are data structures that are used to represent relationships between objects. A graph consists of vertices and edges connecting the vertices. Networks can be used to represent networks, social relationships, and transportation routes, among other things.

- Dynamic programming: is a programming technique used to solve problems by optimizing a cost function. Dynamic programming is useful for problems that can be divided into smaller subproblems and can be solved optimally by combining solutions to the subproblems.

3. Search for creative solutions: Next, several solutions will be presented with which the problem can be developed, they were obtained through a brainstorming that resulted in the following options:

- Binary search algorithm: One way to improve the efficiency of the search for passengers in the database could be by implementing the binary search algorithm, which would reduce the search time in the case of very large databases.

- Sorting algorithm by mixture: One way to sort the passenger data by different criteria, such as accumulated miles, could be by using the sorting algorithm by mixture, which allows sorting the data efficiently and accurately.

- Tree data structure: To store passenger information, a tree data structure could be a good option, as it allows quick access to the data and efficient searching.

- Use of networks: One way to find the most efficient route for passengers to exit the aircraft in set order could be to use graphs and shortest path search algorithms, such as Dijkstra's algorithm.

- Optimized bubble sort algorithm: To sort the passenger list in order of arrival at the boarding lounge, the optimized bubble sort algorithm could be a good option, as it performs well for small lists and can be optimized to perform better on large lists.

- Use of HashMap’s: To store passenger information and perform efficient searches, a data structure such as HashMap’s could be used to map passenger data to unique keys, which would allow fast and efficient searching of the information.

4. Transition from Ideas to Preliminary Designs: Although the creative solutions proposed may be valid to solve the problem, the use of HashMap's is considered the best option for the following reasons, thus discarding the rest of the creative solutions that emerged in the brainstorming:

- Efficiency: HashMap's offer high efficiency in searching and accessing stored items. Compared to the previously proposed solutions, the use of HashMap's is faster and more efficient, which translates into better system performance.

- Flexibility: HashMap's offer great flexibility in structuring and organizing data. They can be used to store and retrieve data of different types and formats, which makes them very versatile and adaptable to different situations and needs.

- Ease of implementation: the implementation of a HashMap is relatively simple and does not require advanced programming knowledge. This facilitates the development and maintenance of the code and reduces the possibility of errors.

- Support and documentation: HashMap's are a widely used data structure in programming and are widely supported and documented online. This means that there are numerous resources and tools available for use and troubleshooting.

In conclusion, although there are different creative solutions to solve the problem, the use of HashMap's is considered the best option due to its efficiency, flexibility, ease of implementation and support.

5. Evaluation and Selection of the Best Solution: For the justification of the previous selection, in which the use of HashMap was the winner, a series of criteria were made where all the possible solutions found are evaluated, where the best solution to develop is evidenced.

The selected criteria are the following:

- Criterion A. Accuracy of the solution. The alternative delivers a solution:
  [2] Exact (an exact solution is preferred).
  [1] Approximate

- Criterion B. Efficiency. A solution with better efficiency than the others considered is preferred. The efficiency can be:
  [4] Constant
  [3] Greater than constant

[2] Logarithmic
[1] Linear

- Criterion C. Completeness. A solution that finds all solutions is preferred. How many solutions it delivers:
  [3] All
  [2] More than one if any, but not all.
  [1] Only one or none

- Criterion D. Ease of algorithmic implementation:
  [2] Compatible with the basic arithmetic operations of a modern computer hardware.
  [1] Not fully compatible with the basic arithmetic operations of a modern computer system.

Creative solutions	Criterion A	Criterion B	Criterion C	Criterion D	Total points
Binary Search	2	2	1	2	7
Sorting by Mixing	2	2	1	1	6
Tree	2	3	3	2	10
Graphs	1	2	3	1	7
Optimized Bubble Sorting	2	1	1	2	6
HashMap’s	2	4	3	2	11

According to the table, the most effective solution is the use of HashMap's. It obtains the highest score in three of the four criteria and the second highest score in the remaining criterion. It provides an exact solution, with constant time efficiency (O (1)) and can find all solutions. In addition, it is compatible with the basic arithmetic operations of a modern computer.

6. Preparation of Reports and Specifications: Here the possible inputs and outputs that can be presented for the solution of the problem are contemplated. In this way we anticipate possible cases during the program.

Inputs:

- Plain text file with passenger information, including at least full name, document number, seat number, seat class, seat class, miles accrued, whether they require special attention and whether they are a senior citizen.

- Identifier of the flight to which the passengers belong.

- Special call orders for first class, indicating the priority criteria to be considered (e.g., accrued miles, special attention required, senior citizen).

Outputs:

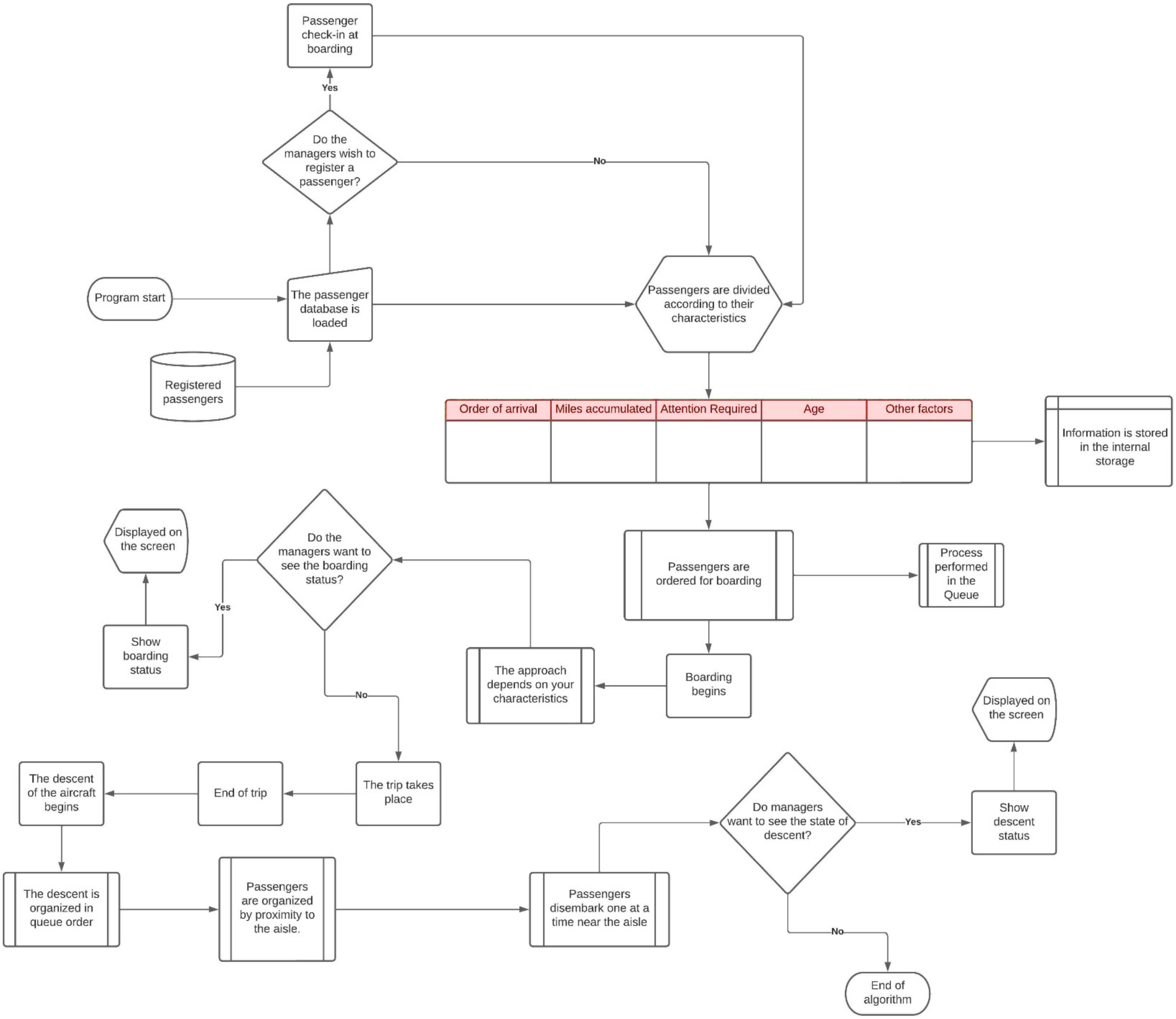
- Confirmation of passenger data upload.

- Order of arrival of passengers at the boarding lounge.

- Order of entry to the aircraft, considering the priority of the first class and the order of arrival for the rest of the passengers.

- Order of departure from the aircraft, considering the distribution of seats.

Algorithm base flowchart


Passenger check-in at boarding
Yes
Do the managers wish to register a passenger?
No
Program start
The passenger database is loaded
Passengers are divided according to their characteristics
Registered passengers
Order of arrival
Miles accumulated
Attention Required
Age
Other factors
Information is stored in the internal storage
Displayed on the screen
Do the managers want to see the boarding status?
Yes
Show boarding status
Passengers are ordered for boarding
Process performed in the Queue
The approach depends on your characteristics
Boarding begins
No
Displayed on the screen
The descent of the aircraft begins
End of trip
The trip takes place
Do managers want to see the state of descent?
Yes
Show descent status
The descent is organized in queue order
Passengers are organized by proximity to the aisle.
Passengers disembark one at a time near the aisle
No
End of algorithm