Integrating Task I

(Engineering method)

Juan José López

Juan David Patiño

Juan Yustes

Computation and Discrete Structures I

Jeison Mejia Murillo

Faculty of Engineering

ICESI University

2022

Phase 1: Problem identification

We have a clinic which has problems for the organization of their patients, because they have difficulty registering the entry and exit of customers and assign them to a department, they also have problems for those people who require more instant attention, because those people can not go to the end of the line, this clinic does not have a defined customer system.

Phase 2: Gathering the necessary information

Once we have the problem defined and we know all the needs to be covered, we are going to list each thing that is needed:

R1 - Enter customer.
R2 - Assign shift.
R3 - Lead person.
R4 - Customer search.
R5 - Create customer.
R6 - Exit customer.
R7 - Undo login/logout.
R8 - Monitoring.
R9 - Upload clients from a text file.

Phase 3: Search for creative solutions

Once we know the needs that we must satisfy, we must think of a solution that seeks to solve all the requirements that were identified, the following ideas were obtained:

Option 1: Personal.

For this solution it is proposed to hire different people to take on the role of logistics, each of which is assigned one or a couple of specific tasks, to meet all the needs, for example: We place a person who takes note and assigns a turn to each patient who enters, another to help the person to be directed to their respective department, etc..

Option 2: Software.

We have an idea to develop a software program which manages to systematize in the best way the order of the people who are entering and leaving the clinic, so that the person who enters has to enter their data, a process is followed and at the end their respective exit is recorded, clearly it would cover all the needs of the clinic.

Option 3: Tickets.

The last idea that was given was a ticket machine to order, and have been seen in different places that kind of methodology, where a person / customer enters the establishment and to be served claims a physical ticket from a dispenser, according to that ticket assigned waiting to be called and directed, is an idea for the order that is needed in the queue of patients that has the clinic.

Phase 4: Choice of solution

Many variables were taken into account when making a decision, because all the ideas were thought with a purpose thinking that it would be the right solution, but finally we realized that the most optimal and complete is a specific one, first we will mention the reasons why the others were discarded and finally we will say the reasons why that solution is the best.

We have:

Tickets: This solution was discarded because it has many disadvantages, first of all, being a physical ticket, it would waste an excessive amount of paper, totally unnecessary and would contribute to the overexploitation of natural resources that are harmful to nature, besides that, it would not cover all the needs, because each person who enters would have a turn at the end of the line, Therefore, priority would not be given to those people who need it, such as people with illnesses or pregnant women. In addition, no record would be kept of the people who enter the clinic, and consequently the exit of people would not be taken into account, therefore, within the clinic there would be an order but that order would not be registered in any way, which would be affected in the long term because there would be no record of each patient.

Personnel: Here also affect different factors such as economics, hiring more people to the clinic would imply a much higher cost since a salary would have to be assigned, apart from that it would not be profitable enough to record people manually, because, although there is a record of patients, physical records are usually more prone to loss, damage, etc.

Software: This was the chosen idea, since it was considered the most complete, unlike the other two ideas, it does not have unnecessary paper expenses, since everything will be saved by the program that will be developed, and it would be much cheaper than paying more people, besides that it covers all the needs required by the clinic, the patient would be registered when entering, he would be assigned a shift, he would be guided to the respective department and then his discharge would be registered, it would be easier to search for a person in the database, among other things.

Phase 5: Design and implementation

Table of requirements

FUNCTIONAL REQUIREMENTS	R1 - Enter customer. R2 - Assign shift. R3 - Lead person. R4 - Customer search. R5 - Exit customer. R6 - Undo login/logout. R7 - Monitoring. R8 - Upload clients from a text file.

NAME OR IDENTIFIER	R1 - Enter customer.		
SUMMARY	Glue the customer into the queue.		
TICKETS	Name of entry	Data type	Selection or repetition condition
	name	String	
	id	String	
	genre	String	
	pregnant	int	
	elderly	int	
	illness	int	
OUTCOME OR POSTCONDITION	Message confirming registration.		

EXITS	Output name	Data type	Selection or repetition condition
	message	String	

NAME OR IDENTIFIER	R2 - Assign shift		
SUMMARY	The person will be given an appointment once registered.		
TICKETS	Name of entry	Data type	Selection or repetition condition
	id	String	
	priority	int	
	lab	int	
OUTCOME OR POSTCONDITION	Message saying that you are in the queue.		
EXITS	Output name	Data type	Selection or repetition condition
	Message	String	

NAME OR IDENTIFIER	R3 - Lead person.		
SUMMARY	The user must enter the required data and the event will be registered.		
TICKETS	Name of entry	Data type	Selection or repetition condition
	lab	int	
OUTCOME OR POSTCONDITION	Message confirming that the person has been addressed.		
EXITS	Output name	Data type	Selection or repetition condition
	message	String	

NAME OR IDENTIFIER	R4 - Customer search.		
SUMMARY	By means of the ID the person is searched in the customer data and it is confirmed whether he/she is registered or not.		
TICKETS	Name of entry	Data type	Selection or repetition condition
	id	String	
OUTCOME OR POSTCONDITION	The information of the searched person or a message saying that he/she is not found is displayed.		
EXITS	Output name	Data type	Selection or repetition condition
	patient	Patient	

NAME OR IDENTIFIER	R5 - Exit customer.		
SUMMARY	Once the person has passed through the corresponding laboratory, his or her discharge is recorded.		
TICKETS	Name of entry	Data type	Selection or repetition condition
	id	String	
OUTCOME OR POSTCONDITION	Message stating that the person has been graduated.		
EXITS	Output name	Data type	Selection or repetition condition
	message	String	

NAME OR IDENTIFIER	R6 - Undo login/logout.

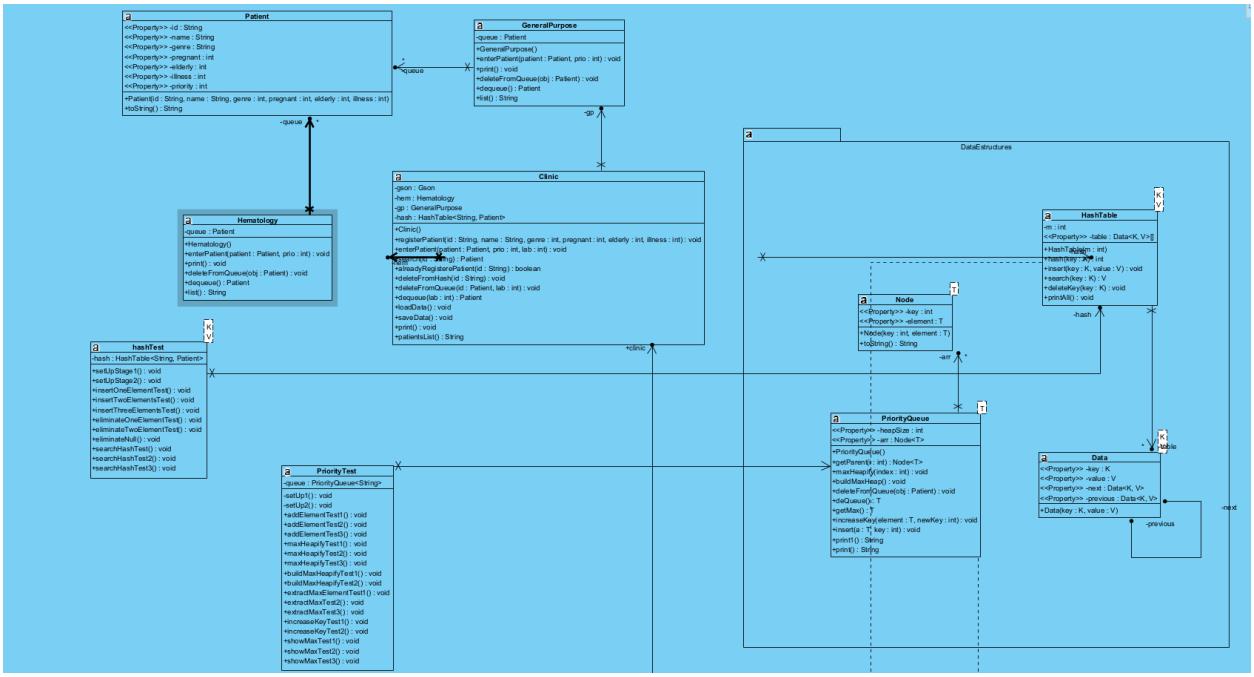
SUMMARY	Each time one of these options is made, at the end of the process you are asked if you want to undo it.		
TICKETS	Name of entry	Data type	Selection or repetition condition
	undo	int	
OUTCOME OR POSTCONDITION	Returns to the initial menu.		

NAME OR IDENTIFIER	R7 - Monitoring.		
SUMMARY	Selecting this option should display the summary of all patients.		
OUTCOME OR POSTCONDITION	Message displaying general customer information.		
EXITS	Output name	Data type	Selection or repetition condition
	patientList	String	

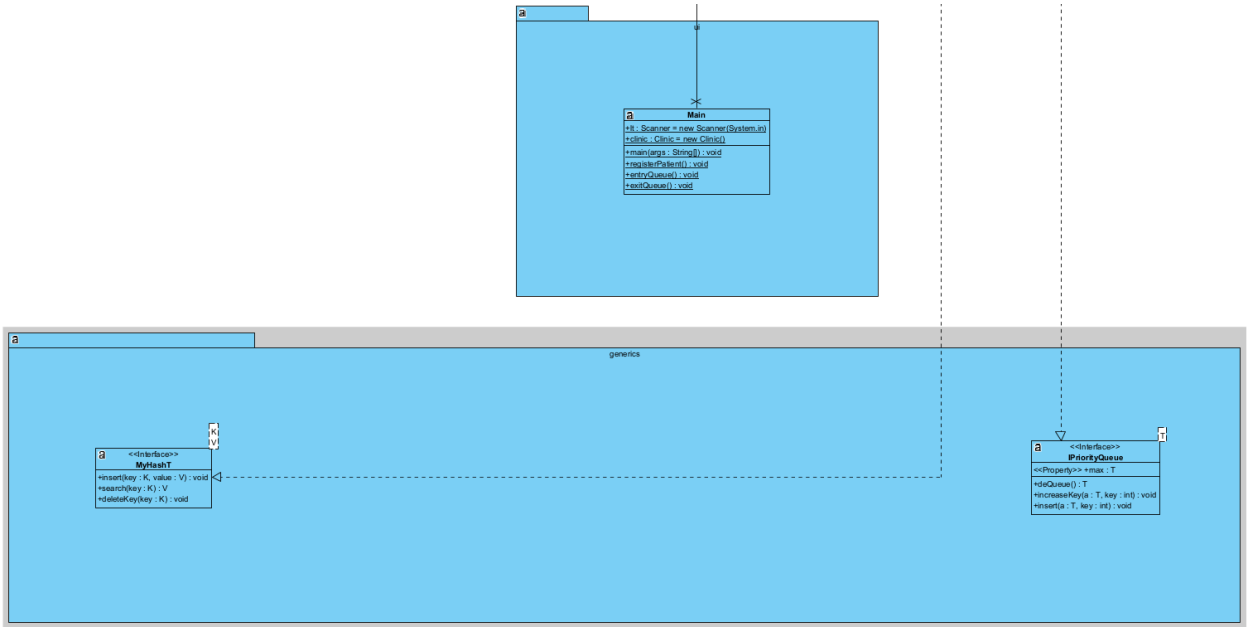
NAME OR IDENTIFIER	R8 - Upload clients from a text file.
SUMMARY	Maintain the list of registered patients in an external file.
OUTCOME OR POSTCONDITION	.txt" file with the patients in JSON format.

Class diagram

Model:



UI and Generics:



TAD

TAD Hash-Table

Abstract Object: Hash Table Invariant: A key, contains one value
Primitive Operations insert: K key, V value void search: K key V: object deleteKey: K key void

TAD Priority Queue
Abstract Object: Priority Queue Invariant: The value associated as the highest, is the first to exit.
Primitive Operations deQueue T value getMax: T value increaseKey:T a, int key void insert: T a, int key void

Complexities

Method 1: Save Data.

Spatial:

The cost of each line is commented on the respective side.

```
public void saveData(){
    ArrayList<Patient> pat=new ArrayList<>(); //1
    try {
        FileOutputStream fos = new FileOutputStream(new File( pathname: "eps.txt")); //1
        for(Data p:hash.getTable()){ //n
            if(p!=null){  //n-1
                pat.add((Patient) p.getValue()); //n-1
            }
        }
        String json=gson.toJson(pat); //    1
        fos.write(json.getBytes(StandardCharsets.UTF_8)); //1
        fos.close(); //1


    } catch (FileNotFoundException e) {
        e.printStackTrace(); //1
    } catch (IOException e) {
        e.printStackTrace(); //1
    }

}
```

Temporal: 6+n+2(n-1)=O(n).

Method 2: Load Data

Spatial:

The cost of each line is commented on the respective side.

```
public void loadData(){
    try {
        File file=new File( pathname: "eps.txt");      //1
        FileInputStream fis=new FileInputStream(file);   //1
        BufferedReader reader= new BufferedReader(new InputStreamReader(fis));  //1
        String json=""; //1
        String line;    //1
        while ((line=reader.readLine())!=null){ //n
            json+=line; //n-1
        }
        fis.close();    //1
        Gson gson=new Gson();   //1
        Patient[] patients=gson.fromJson(json, Patient[].class); //n

        ArrayList<Patient> arrayPatients=new ArrayList<>(); //1
        Collections.addAll(arrayPatients, patients); //n

        for (Patient p: patients){ //n
            hash.insert(p.getId(),p);  //n-1
        }

    } catch (IOException e) {
        e.printStackTrace(); //1
    }
```

Time Complexity: 9+4n+2(n-1) = O(n)