|  |  |
| --- | --- |
| Client | Healthcare company |
| User | * Receptionist |
| Requirements | * **R1:** The system must be able to load the data base with all the previously registered patients. * **R2:** The system must allow the receptionist to either search for an already registered patient or register them in case they aren’t. Of a patient the receptionist registers name, surname, id, gender, age, and a special status if they have any. * **R3:** The system must have more a queue per unit of the laboratory and must be able to assign a priority to every patient waiting in queue, this priority will depend on their age and their special status. * **R4:** The system must be able to separate patients depending on whether they have priority or not, assigning them a different priority depending on their characteristics. * **R5:** The system must be able to undo any recently made change to any of its subsystems, whether it is the data base or either one of the queues. * **R6:** The system must give the option to the receptionist to visualize at any given moment the patients found in the facilities, be it in general or per unit, the order in which they are to be called, and the option to register the exit of any of the patients. * **R7:** The system must be able to take any patient out of either queue in the case they are called, let other patients go on to their turns, this must be simulated by taking the time each patient is waiting and taking them out when around two minutes have passed. |
| Context | A health care institution needs the first version of a system that allows them the management of the entrance and exit of their patients. The receptionist will use the program. |
| Non-Functional requirements | * **Nr1:** The action of search and registration must be carried out in the most efficient way possible. |
| Process requirements | * **Pr1:** The system must consist of proprietary implementation of all the main algorithms to be implemented. |

1. **Entity identification**
   * **Database** 
     + **Patient**
   * **Queue**
   * **Time management**
2. **Subsystem hierarchy**

1. **First level of requirement assigning**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Database | Queue | Time management |
| R1 | X |  |  |
| R2 | X |  |  |
| R3 | X | X |  |
| R4 |  | X |  |
| R5 | X | X |  |
| R6 |  | X |  |
| R7 |  | X | X |

1. **Requirements sub specification.**
   * **R1\_Dat\_1:** As the program initializes, the system must load all the information relating to the patients that have at one point been into the facilities.
   * **R2\_Dat\_1:** The system must be able to search any of the patients in the data base given their id.
   * **R2\_Dat\_2:** The system must be able to register a new patient into the database. Of a patient the receptionist registers name, surname, id, gender, age, and a special status if they have any.
   * **R3\_Dat\_1:** The system must be able to use the data of a patient to assign them with a priority.
   * **R3\_Queue\_1:** The system must have a different queue per unit of the laboratory.
   * **R3\_Queue\_2:** The system must be able to assign a patient with a priority the moment they are inside the facilities.
   * **R4\_Queue\_1:** The system must be able to differentiate between patients depending on their priority.
   * **R4\_Queue\_2:** The system must have different base priorities depending on the conditions of the patient.
   * **R5\_Dat\_1:** The system must be able to unregister any patient were the user to request it, this only applies to changes made not only recently but also in the same session. The order in which the changes are to be undone follows that the last done is the first undone.
   * **R5\_Queue\_1:** The system must be able to take a patient out of any of the queues given that the user request so, this applies only to changes made recently and in the same session. The order in which the changes are to be undone follows that the last done is the first undone.
   * **R6\_Queue\_1:** The system must be able to report to the user the patients that are currently in a queue and their order in said queue.
   * **R6\_Queue\_2:** The system must be able to allow the user to take a patient out of either any of the queues, or the registry of access to the facilities.
   * **R7\_Queue\_1:** The system must be able to take any patient out of a waiting queue given an external condition is met.
   * **R7\_Time\_1:** The system must be able to track the time each patient is in the waiting queue to know when to tick the exit condition of said patient.
2. **Organizar por entidades los requerimientos: Ya**
3. **Second level partitioning.**
   * **Database** 
     + **Patient**
   * **Queue**
     + **Waiting list**
     + **Entry registry**
   * **Time management**
4. **Second level sub requirements analysis**
   * **R1\_Dat\_1:** As the program initializes, the system must load all the information relating to the patients that have at one point been into the facilities.
   * **R2\_Dat\_1:** The system must be able to search any of the patients in the data base given their id.
   * **R2\_Dat\_2:** The system must be able to register a new patient into the database. Of a patient the receptionist registers name, surname, id, gender, age, and a special status if they have any.
   * **R3\_Dat\_1:** The system must be able to use the data of a patient to assign them with a priority.
   * **R3\_Unit\_1:** The system must have different queues for each unit of the lab.
   * **R3\_Ent\_1:** The system must be able to assign a patient with a priority the moment they are registered as inside the facilities.
   * **R4\_Ent\_1:** The system must be able to differentiate between patients depending on their priority.
   * **R4\_Unit\_1:** The system must have separate queues for patients withs different priority.
   * **R5\_Dat\_1:** The system must be able to unregister any patient were the user to request it, this only applies to changes made not only recently but also in the same session. The order in which the changes are to be undone follows that the last done is the first undone.
   * **R5\_Unit\_1:** The system must be able to take a patient out of any of the queues given that the user request so, this applies only to changes made recently and in the same session. The order in which the changes are to be undone follows that the last done is the first undone.
   * **R5\_Ent\_1:** The system must be able to take a patient out of any of the Entry registration given that the user request so, this applies only to changes made recently and in the same session. The order in which the changes are to be undone follows that the last done is the first undone.
   * **R6\_Unit\_1:** The system must be able to report to the user the patients that are currently in a queue and their order in said queue.
   * **R6\_Unit\_2:** The system must be able to allow the user to take a patient out of the waiting queues.
   * **R6\_Ent\_1:** The system must be able to allow the user to take a patient out of the registry of access to the facilities.
   * **R7\_Unit\_1:** The system must be able to take any patient out of a waiting queue given an external condition.
   * **R7\_Time\_1:** The system must be able to track the time each patient is in the waiting queue to know when to tick the exit condition of said patient.
5. **Segundo nivel de asignación de requerimientos a subsistemas.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Database | Queue | | Time management |
|  | Patient | Unit management | Entry registration | Time management |
| R1 | R1\_Dat\_1 |  |  |  |
| R2 | R2\_Dat\_1  R2\_Dat\_2 |  |  |  |
| R3 | R3\_Dat\_1 | R3\_Unit\_1 | R3\_Ent\_1 |  |
| R4 |  | R4\_ Unit \_1 | R4\_Ent\_1 |  |
| R5 | R5\_Dat\_1 | R5\_ Unit \_1 | R5\_Ent\_1 |  |
| R6 |  | R6\_ Unit \_1  R6\_ Unit \_2 | R6\_Ent\_1 |  |
| R7 |  | R7\_ Unit \_1 |  | R7\_Time\_1 |