**Emergency Room Simulator**

**Requirement Specification:**

This is an emergency room simulator that will simulate a week of an emergency room by minutes. The simulator will be able to take a constant arrival rate of patients, number of doctors, and number of nurses and then output the average wait time and visit time per patient treated. It will also keep record and output how many patients arrived at the emergency room compared to how many were actually treated. The user will be able to display all patients that were treated and search for patients by first name after the simulation is over.

**Use Cases (User input/Clock ticks):**

User Enters Arrival Rate

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1. | User enters an integer for the arrival rate per hour of patients. | If invalid integer system prompts for user to try again. |
| 2. |  | System stores the value as the arrival rate per hour |

User Enters Number of Nurses and Doctors

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1. | User enters an integer for the number of doctors. | If invalid integer system prompts for user to try again. |
| 2. |  | System stores the value as the number of Doctors, and prompts for user to enter the number of nurses. |
| 3. | User enters an integer for the number of nurses. | If invalid integer system prompts for user to try again. |
| 4. |  | System stores the value as the number of Nurses. |

Emergency Room Update

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1. | Simulator gets a random number and compares it to the arrival rate. | System checks if the number is less than the arrival rate and returns true or false. |
| 2. |  | If the system returns false then update ends. If system returns true system creates a new patient. |
| 3. | Simulator takes the illness level of the patient and checks if it is less than 11. | System returns true if value is less than 11 or false if value is greater than 11. |
| 4. |  | If the system returns true the patient is put into a nurse priority queue, but if false the patient is put into a doctor priority queue. |

Nurse Queue Update

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1. | Simulator checks if the nurse does not have a patient. | System returns true if there is no patient or false if there is a patient. |
| 2. |  | If true the system checks if the nurse’s priority queue has any patients ready to be treated. If false nothing happens. |
| 3. |  | If system returns true it gives that nurse the top patient from the nurse priority queue. If system returns false nothing occurs. |
| 4. | Simulator checks if the nurse does have a patient. | System returns true if there is a patient or false if there is not a patient. |
| 5. |  | If system returns true then it checks whether the patient is done being treated. |
| 6. |  | If system returns true it removes the patient and if false the system does nothing. |

Doctor Queue Update

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1. | Simulator checks if the doctor does not have a patient. | System returns true if there is no patient or false if there is a patient. |
| 2. | Simulator checks if there are any patients in the doctor priority queue. | If true the system checks if the doctor’s priority queue has any patients ready to be treated. If false nothing happens. |
| 3. |  | If system returns true it gives that doctor the top patient from the doctor priority queue. If system returns false nothing occurs. |
| 4. | Simulator checks if the doctor does not have a patient again. | System returns true if there is no patient or false if there is a patient. |
| 5. | Simulator checks if there are any patients in the nurse priority queue. | If true the system checks if the nurse’s priority queue has any patients ready to be treated. If false nothing happens. |
| 6. |  | If system returns true it gives that doctor the top patient from the nurse priority queue. If system returns false nothing occurs. |
| 7. | Simulator checks if the doctor does have a patient. | System returns true if there is a patient or false if there is not a patient. |
| 8. |  | If system returns true then it checks whether the patient is done being treated. |
| 9. |  | If system returns true it removes the patient and if false the system does nothing. |

User Displays Treated Patients

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1. | User issues the command to display all the treated patients. |  |
| 2. |  | System outputs every patient that has been treated that week at the hospital. |

User Searches for a Patient

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1. | User issues the command to search for a specific patient. |  |
| 2. |  | System prompts for name. |
| 3. | User enters name. | If user cancels entry of name, process terminates. If invalid entry system prompts for user to try again. |
| 4. |  | System outputs all patients with that name and their illness level. |

**Pseudo – Code:**

* Enter\_data () in Simulator.h takes in the arrival rate, number of doctors, number of nurses. Then creates the nurse/doctor queues and sets all the queues up.
* Run\_simulation () in Simulator.h loops through the queues and updates them with each clock tick.
* Show\_list () in Simulator.h outputs all the patients that have been treated by the emergency room and their illness level.
* Search (name) in Simulation.h searches through treated patients to find a person by their name.
* End\_data () in Simulation.h displays the average wait time, visit time, patients that enter the emergency room and patients that were served.
* End\_menu () in Simulation.h lets the user choose whether they want to display all treated patients, search for a patient by name or exit the simulation.
* Update () in DoctorQueue.h, NurseQueue.h, and EmergencyRoomQueue.h updates all the queues by either giving them a new patient or removing treated patients.
* Add\_patients () in Patient.h takes all the patients from a file and puts them into an accessible array.
* Next\_int () in Random.h gets a random integer from 0 to a selected number.
* Next\_double () in Random.h gets a random double from 0 to 1.
* Get () in all header files to get specific private variables from certain classes.

|  |
| --- |
| **Emergency Room** |
| - arrival\_rate : double  - num\_of\_doctors : int  - num\_of\_nurses : int  - num\_of\_patients : int |
| + set\_patient ()  + set\_arrival\_rate ()  + get\_num\_of\_patients ()  + set\_num\_of\_doctors ()  + set\_num\_of\_nurses ()  + illness\_level ()  + update () |

**UML Diagram:**

|  |
| --- |
| **Simulator/Hospital** |
| - num\_of\_doctors : int  - num\_of\_nurses : int  - total\_time : int  - average\_visit\_time : double  - average\_wait\_time: double  - total\_patients\_served : int  - clock : int  - read\_int () |
| + enter\_data ()  + run\_simulation ()  + end\_data ()  + end\_menu ()  + search (name)  + show\_list () |

|  |
| --- |
| **Nurse** |
| - treatment\_time : int  - service\_time : int  - visit\_time : double  - nurse\_wait\_time : double  - patients served : int |
| + set\_emergency\_queue ()  + get\_num\_of\_patients\_served ()  + get\_treatment\_time ()  + get\_visit\_time ()  + get\_nurse\_wait\_time ()  + update () |

|  |
| --- |
| **Doctor** |
| - treatment\_time : int  - service\_time : int  - visit\_time : double  - doctor\_wait\_time : double  - patients served : int |
| + set\_emergency\_queue ()  + get\_num\_of\_patients\_served ()  + get\_treatment\_time ()  + get\_visit\_time ()  + get\_doctor\_wait\_time ()  + update () |

|  |
| --- |
| **Patient** |
| - arrival\_time : int  - illness\_level : int  - start\_treatment\_time : int  - first\_name : string  - last\_name : string |
| + add\_patients ()  + get\_first\_name ()  + get\_last\_name () |

|  |
| --- |
| **Random** |
| + next\_double ()  + next\_int () |