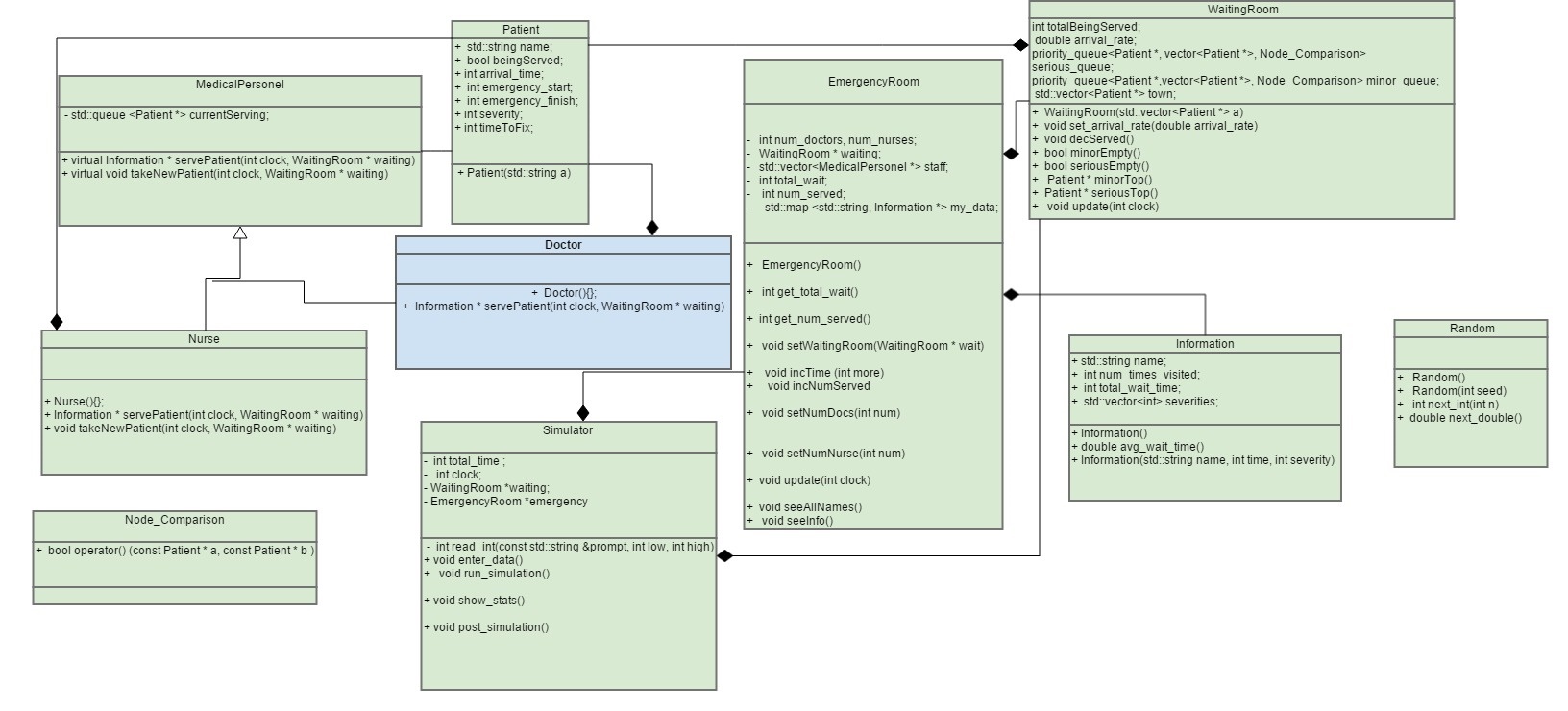
**Requirement Specifications**

Write an emergency room simulator that examines a week in life of emergency room on minute by minute basis. When a patient arrives at the hospital, s/he is assigned a number indicating the seriousness of illness. Patients assigned a number between 1 and 10 are the lowest priority of in emergency room. The probability of receiving a patient with an assigned number between 1 and 10 is 70 percent. There is 20 percent probability of receiving a patient that is assigned a number between 11 and 15. These patients have higher priority than any one assigned a number between 1 and 10. The highest priority is given to patients that are assigned a number between 16 and 20. There is 10 percent probability of receiving a patient with highest priority.

The system should be capable of recording all the visits by patients in the emergency room; the records should store number of visits of a particular patient and severity of illness. User of this system should be able to input these values:

1. Average hourly patient arrival rate.
2. Number of doctors and nurses working in emergency room.

Given these values the system should calculate the average visit time from arrival to discharged time. Lastly the system should give users the option to displaying names of patients treated and access records of resident by names.



Use Cases

Use Case for Main Menu

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1 |  | User is welcomed and asked for the arrival rate of patients (patients/hour) |
| 2 | User enters the number of patients per hour |  |
| 3 |  | If the number entered is less than 1 or greater than 60, the user is informed that the given rate is out of range and then is prompted for a new number. |
| 4 |  | System divides result by 60 and then sets the arrival rate field in WaitingRoom class (patients/min) |
| 5 |  | User is prompted for the number of doctors that are working |
| 6 | User enters the number of doctors that are working |  |
| 7 |  | If the number is less than 1 or greater than 20, the user is informed that the given number is out of range and then is prompted for a new number. |
| 8 |  | System changes number of doctors in the EmergencyRoom class to the user specified value |
| 9 |  | User is prompted for the number of nurses that are working |
| 10 | User enters the number of doctors that are working |  |
| 11 |  | If the number is less than 1 or greater than 20, the user is informed that the given number is out of range and then is prompted for a new number. |
| 12 |  | System changes number of nurses in the EmergencyRoom class to the user specified value |
| 13 |  | System runs simulation |
| 14 |  | System outputs the number of people served, the total wait time of all patients, and the average wait time per patient |
| 15 |  | System displays menu with options to list the names of all residents that were treated, and retrieve the record of a resident by name |
| 16 | User picks an option |  |

Use case for post-simulation menu (list names)

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1 | User chooses to display the names of all the residents that were treated (i.e. user chooses 1) |  |
| 2 |  | System lists the names of all those that were treated |

Use case for post-simulation menu (retrieve record)

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1 | User chooses to retrieve a record of a patient (i.e. user chooses 2) |  |
| 2 |  | System displays all the times the individual was in the hospital (or says they weren’t if they were not). System also displays the severity of the individual’s injuries and time spent waiting in the hospital by that patient |

Use case for post-simulation menu (exit)

|  |  |  |
| --- | --- | --- |
| Step | User’s Action | System’s Response |
| 1 | User chooses to exit (i.e. user chooses 3) |  |
| 2 |  | System terminates program |

Use Case for During Simulation

|  |  |  |
| --- | --- | --- |
| Step | User’s Action (clock) | System’s Response |
| 1 | Clock is increased |  |
| 2 |  | System checks to see if clock has exceeded total time, if so the simulation terminates |
| 3 |  | System randomly chooses number between 0 and 1, if that number is less than the arrival rate for patients then a patient is added to one of the WaitingRoom priority queues. The queue is determined by a second randomly generated number between 0 and 9. If the second number is less than 7 the patient is assigned a severity between 1 and 10 and is placed in the minor\_queue (priority queue) on WaitingRoom. If the number is 7 or 8 then the patient is assigned a severity of 11-15 and is placed in the serious\_queue (priority queue) in WaitingRoom. If the number is 9 the patient is assigned a severity of 16-20 and is placed in the serious\_queue (priority queue) in WaitingRoom. |
| 4 |  | System checks to see if any of the nurses’ queues are empty, if there is an empty queue, a patient is moved out of the minor\_queue in WaitingRoom and into that nurse’s queue in the EmergencyRoom class assuming that the minor\_queue has someone in it. The patient is also given a number 1-10 that indicates how long the patient will need to be treated for. |
| 5 |  | If a nurse’s queue has someone in it, the system checks to see if enough time has elapsed to fully treat the patient. If not, nothing happens. However, if enough time has elapsed the system updates the patients exit time, adds the patient’s total wait time to the overall total wait time, adds one to the amount of people the hospital has served, and adds one to the number that the nurses have served. Also, push the patient’s data into a map using the patient’s name as the key for record keeping. |
| 6 |  | System checks to see if any of the doctors’ queues are empty, if there is an empty queue, a patient is moved out of the serious\_queue in WaitingRoom and into that doctor’s queue in the EmergencyRoom class. If the serious\_queue is empty a patient from the minor\_queue will be moved into the doctor’s queue unless the minor\_queue is empty as well. The patient is also given a number 1-20 that indicates how long the patient will need to be treated for. |
| 7 |  | If a doctor’s queue has someone in it, the system checks to see if enough time has elapsed to fully treat the patient. If not, nothing happens. However, if enough time has elapsed the system updates the patients exit time, adds the patient’s total wait time to the overall total wait time, adds one to the amount of people the hospital has served, and adds one to the number that the doctors have served. Also, push the patient’s data into a map using the patient’s name as the key for record keeping. |
| 8 | Clock is increased |  |
| 9 |  | Process repeats |

Pseudo Code

For creating a vector of all the names in the town

Create an input file stream

Open the file stream

While not the end of the file

{

Get a name

Push a new patient with that name into a vector

}

Close the file stream

For entering new Patients into the waiting room

Create a random double between 0-1

If that number is less than the arrival rate for patients select a random Patient in the vector of patients and add them to the waiting room if they are not being served

Change that patient to “Being Served”

Choose a random number 1-10

If that number is less than 7 give that patient a severity of 0-9 and add them to the minor emergency priority queue

If the number is 7 or 8 give that patient a severity of 11-15 and place them in the serious emergency priority queue

If the number is 9 give the patient a severity of 16-20 and place them in the serious emergency priority queue

For inside the emergency room

For each nurse check if their queue is empty

If it isn’t check to see if their patient has spent enough time to be healed

If the patient hasn’t spent enough time do nothing

If the patient has been treated fully set his/her emergency room finish time

Then add their total wait time to the overall total wait time for the hospital

Then add one to the amount the hospital has helped overall

Then push the info of the patient’s visit into a database (map)

Then set the patient to not being served

Then pop the patient from the nurse’s queue

If a nurse isn’t serving anyone check to see if the minor priority queue has someone in it

If it does pop the top patient from that queue and put them in the nurse’s queue

Update the patients start emergency room time

Give the patient a time needed in the emergency room between 1-10

For each doctor check if their queue is empty

If it isn’t check to see if their patient has spent enough time to be healed

If the patient hasn’t spent enough time do nothing

If the patient has been treated fully set his/her emergency room finish time

Then add their total wait time to the overall total wait time for the hospital

Then add one to the amount the hospital has helped overall

Then push the info of the patient’s visit into a database (map)

Then set the patient to not being served

Then pop the patient from the doctor’s queue

If a doctor isn’t serving anyone check to see if the serious priority queue has someone in it

If it does pop the top patient from that queue and put them in the doctor’s queue

Update the patients start emergency room time

Give the patient a time needed in the emergency room between 1-20

If the serious priority queue is empty check to see if there is anyone in the minor priority queue

If there is pop the top patient from that queue and put them in the doctor’s queue

Update the patients start emergency room time

Give the patient a time needed in the emergency room between 1-20

For getting all names in the database

Make an iterator of the map

Initialize the iterator to the start of the map

Loop the iterator through the whole map outputting the maps key

For getting info on a name

Find a name in the map

Output all the data in the Information class that is mapped to by that name