**Requirements Specification:**

Program is supposed to simulate a week long hospital emergency room on a minute by minute base. Whenever a patient arrives to the hospital they should be given a number to show how severe their visit is. The hospital has two caregiver’s doctors and nurses. User should be able to enter the average hourly patient arrival rate including the number of doctors and nurses that are going to be working at that time. Patients with numbers assigned from 1 to 10 can occur 70% of the time and these patients are the less priority. Patients with numbers from 11 to 15 can occur 20% of the time. Patients with numbers 16 to 20 have a probability of 10%. Program should be able to store information of the number of times the patient visits and the severity. Finally, the system should be able to calculate the average visit time for the emergency room. Program will be able to show on the screen all of the patients name that got served along with searching for a patient by name on the database.

UML Diagrams:

Class EmergencyRoom

+Int num\_treated

* Waiting\_room room

+Int total\_wait

+int num\_doctors

+int num\_nurses

-Map<string,Information\*>

Class Medical\_team

-queue<Patient\*>

+Virtual treat\_patient()

+Virtual treat\_time()

Class Information

-int num\_of\_visits

- int total\_wait\_time

-int injury

Class Nurse

+void treat\_patient()

+int treat\_time()

Class Patient

+String name

-int severity\_illness

-int arrival\_time

-Int num\_of\_visits

+int fix\_time

+int start

+int end

+string getname()

+int getillness()

+int getvisits()

+int getarrival()

Class Simulator

-int clock

-int total\_time

+ void enter\_data()

+void Run\_simulator()

+void final\_report()

+void post\_sim()

Class Random:

+Random()

+Random(int seed)

+int next\_int(int n)

+double next\_double()

Class Doctor

+void treat\_patient()

+ int treat\_time()

Class Get\_Names

-vector<string> names

+vector<string> get\_names()

Class Waiting Room

-double arrival\_rate

-int to\_be\_served

Priotry\_queue<\*patient,vector<Patient\*>,Compare> minor

-Priotry\_queue<\*patient, vector<Patient\*> , Compare> major

-vector<Patient\*> town

-void update(int clock)

**Use Cases:**

**Display Menu:**

|  |  |
| --- | --- |
| User Action | System’s Response |
| 1. User enters the command to see the name of all patients served at the hospital |  |
| 2) | Takes the user to the function and outputs the data saved of every patient |
| 1. User enters to search by name every patient that has gotten treated |  |
|  | Responds by display all the records for the given name. |
| 1. User enters 0 to exit the program |  |
|  | Exits the program |

**Simulator:**

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| 1. **Command to run the simulator** |  |
|  | **Responds by promoting the user to enter average hourly patient arrival time** |
| 1. **User enters arrival time** |  |
| **4)** | **System prompts for the number of doctors** |
| 1. **User enters number of doctors** |  |
|  | **System prompts the number of nurses** |
| 1. **User enters number of nurses** |  |
|  | **Simulator runs** |

Pseudo Code:

**Void run\_simulator()**

Begin:

1. Starts the simulator
2. Do
3. Updates clock for queues
4. While the clock is less than total time

End

**Void enter\_data()**

Begin:

1. Prompt user to enter average patient arrival rate time
2. Reads command
3. Prompts user for number of doctors
4. Reads command
5. Prompts user for number of nurses
6. Read commands
7. Sends appropriate message to simulator

End

**Void search\_name()**

Begin:

1. User enters the name they want to search
2. System tries to locate that name in the map
3. Do
4. Iterate through every element
5. Output all the records for the name that matches the users request
6. While it does not reach the end of the map
7. End

Void update(int clock)

Begin:

1. Random number is generated