**Really Overpowered Unicorns Save Everyone (R.O.U.S.E.) Hospital**

**Simulation Management Proposal**

Proposed by

Ancel Carson, Jacob Kelsey, Spencer Davis

10-05-2018

ROUSE Hospital program specifications

Basic Overview:

We are proposing to create three programs to complete this project. First a StopWatch class to record the times that a patient arrived, current amount of time they have been waiting, and total wait time until they were put in a room. Second a Patient class that randomizes and holds all the patient’s relevant information. Finally an ERSimulator class that brings in patients on a random interval, sorts them into different queues, and places them in a room depending on severity of injury and time they have been waiting.

StopWatch:

The stop watch class will contain a single constructor. And a series of methods that retrieve the time from the system to keep track of how long different operations have been running.

Methods:

There will be four methods in the Stopwatch Class. The first and second methods start and stop a timer respectively. The third method will return the time from start to the current time. The fourth method will return how much time there was between the start and stop methods being called.

Patient:

The patient class will contain a single constructor that sets the patient’s rating, how long they will take up a hospital room, and start a timer to count how long they have been waiting.

Private Methods:

There will be four private methods in the Patient class. The first will be to get a patients rating based on the percentages specified. Using Math.random() a percentage value will be retrieved that, though a switch statement will assign the rating. The Second method will assign the base amount of time that a patient will be in a hospital room. This number is assigned from the rating given in the first method. Values assigned are again taken from the given estimations from the hospital staff. The Third method will create two percentages. The first percentage will be on the chance of the patient needing more time in the hospital room. The second percentage will be by how much the room time should be increased to a maximum of 50%. As described by the staff, there is a 20% chance that any one patient may need more time in the hospital room. The final private method will assign a single value that assists in determining the patient’s priority based on their rating.

Public Methods:

There will be three public methods in the Patient class. The first method will return a patient’s priority. This number will be how long they have been waiting multiplied by the number assigned in the fourth private method. The second method will stop the timer and return how long the patient should be in their room. The third method will return how long the patient stayed inside of the waiting room.

ERSimulator:

The ERSimulator class will handle patients coming in and sorting them into their correct groups. The groups will be made up of DoublyLinkedList queues that hold a patient of a correct rating whenever it enters the program. There are four queues, one for each rating. At the beginning of the program the user will be prompted to enter the number of rooms they want available for the current simulation. With that choice, an array of Patient objects will be created to represent the usable rooms and an array of times to hold how long the room will be in use. The remainder of the main method will be inside of a while loop that runs as long as the simulation has not been running for 10 minutes and there are patients in the queue. If 10 minutes have not passed, the program will continue to bring in patients at intervals of 5-10 seconds and sort them into the correct queue. The program will attempt to place a patient into an empty room if any are available. After 10 minutes have passed, the program will stop receiving patients but will continue running until all four of the queues have been emptied. When that happens the output specified by the client will be given and the program will end.

Methods:

There are two extra methods in the ERSimulator class. The first controls the spacing between patients entering the hospital in seconds. This is done using Math.random(). The second method decides which queue currently holds the highest priority value and returns the result to the main method.