Pseudo-Code Program:

prompts user for num of patients per hour

User: user inputs num of patients

User: enters inputs out of range

Program: throws exception

Program: prompts user for num of doctors per hour

User: enters inputs out of range

User: inputs num of doctors

Program: throws exception

Program: prompts user for num of nurses per hour

User: inputs num of nurses

User: enters inputs out of range

Program: throws exception

Program: shows stats, and prompts user to choose from menu.

User selects option number one

User selects out of range

Program clears input and displays menu again

Program displays all patients treated, shows menu again

User selects option number two

Program prompts user for patient name

User enters patient name

User enters invalid name

Program: throws exception

Program displays that patients record, displays menu again

User: selects option three

Program ends the program

enterInfo() {

1 add people to vector

2 input patients per hour

3 divided input by 60 (per hour)

4 input num of docs

5 input num of nurses

6 go to in waitingroom queue and pass through patPerHour in setRate(double)

7 go to in waitingroom queue and pass through numOfNur in setNur(Nurses)

8 go to in waitingroom queue and pass through numOfDoc in setDoc(Doctors)

9 go to in waitingroom queue and pass through the vector in Citizens class in setFirstName(vector<string>)

10 go to in treatmentRoom queue and pass through the dischargeRoom Queue pointer in setDischargeeRoomQueue(DischargeRoomQueue \* d)

11 go to in treatmentRoom queue and pass through the waitintgRoom Queue pointer in setWaitingRoomQueue(WaitingRoomQueue \* d)

12go to hospitalrecords and pass through the hospitalrecords pointer in setHospitalRecords(hospitalRecords \* h)

}

runSimulation() {

1 for each minute In a week

2 go to waitingRoomQueue and update wainting room queue

3 go to treatmentRoomQueue and update treatment room queue

4 go to dischargeRoomQueue and update discharge room queue

}

2a

waitingRoomQueue->update(int clock)

{

1 update random location in vector

2 update random patient’s illness level

3 update random treatment time for doctors

4 update random treatment time for nurses

5 if random number is less than patPerHour

{

5a Create temp patient record

5b if(there are docs or nurs avaible)

{

5b.1 If(if illness lvlfits docs needs)

{

5b.1.1 add patient to doc queue

` if(patient name is not a copy)

5b.1.2 add patient info to patient records

}

5b.2 else if(fits nurs need)

Add patient top nurses queue

If(name is not a copy)

5b.2.1 add patient info to patient records

}

}

}

3a

TreatmentQueue(int clock)

{

0 open up docUpDate()

1 if queue is not empty

2 then check for patients that are don’t being treated

3 if any patients then point to the patient in the front of the queue

4 remove patient from queue

5 add doctor to waitingroom

6 set discharge time to clock

7 if waiting room queue has patients

8 while there are doctors and and waiting patients

9 point to patint in waiting room queue

10 remove patient from waitingroom

11 take away one doctor

12 set patient start timem to clock

13 add patint to doc queue

0 open up nurUpdate()

1 if queue is not empty

2 then check for patients that are don’t being treated

3 if any patients then point to the patient in the front of the queue

4 remove patient from queue

5 add nurse to waitingroom

6 set discharge time to clock

7 if waiting room queue has patients

8 while there are nurses and and waiting patients

9 point to patint in waiting room queue

10 remove patient from waitingroom

11 take away one nurse

12 set patient start time to clock

13 add patint to nur queue   
}

4a

DischargeRoomQueue->update(clock){

1 if doc queue is not empty

2point to front patient

3 set discharge time to clock

4 remove patient

5Add visit time to total visit tie

6 increment total number of patients

7 delete patient pointer

8if nurseQueue is not empty())

9 point to front patient

10 set discharge time to clock

11 remove patient

12Add visit time to total visit tie

13 increment total number of patients

14 delete patient pointer

}

5 showStats()

{

1 output num of patients treated

2 out put avg time to treat patients

3 display menu

}

6 statsMenu()

{

1 cin choice

2 open switch statement

Case 1: listPatients()

{

Output all the patients name

}

Case 2: findPatient()

{

Cin patient name

Output patients hospital records

}

Case 3: endProgram()

{

Return 0;

}

}