**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Sno.** | **Content** |
| **1.** | Abstract |
| **2.** | Introduction |
| **3.** | Literature |
| **4.** | Objective |
| **5.** | Methodology |
| **6.** | System Requirements |
| **7.** | Pert Chart |
| **8.** | Source Code |
| **9.** | Output |
| **10.** | Limitations |
| **11.** | References |

**ABSTRACT**

This system handles the patients that are victims of mass casualties such as War zones, transportation accidents, bomb blast in a public place, during this time, it is important to utilize the limited medical resources available in the hospital. Following this the system prioritizes the patients pertaining to the categories and fields inherited from Glasgow Coma Scale Study and assigning the patients dynamically to the appropriate Specialist.

Assignment of patients to the particular specialist is done using scheduling algorithms which have been implemented using Activity Scheduling algorithm. This System also asks the number of wards and dynamically assigns the highest priority patients to the next empty war

**INTRODUCTION**

In our day to day life there are certain happening of mass casualties like natural calamities, disasters, injuries during a sports match, bus or train accidents. This disrupts the health of large number of people which need to be operated in parallel. Now in rural areas the problem occurs when the availability of resources is scarce with respect to patients affected and injured due to the disaster happened. As a result, there is need to modify the traditional approach of handling patients in such a manner that the resources available are not overwhelmed due to sporadic arrival of victims affected.

In the early history some scales were introduced to handle such problem like GCS or Pediatric GCS but the problem that arose with these scales was that they were much time consuming. Also, when the number of victims increases, the time along increases with a larger scale. This system solves this problem by automating the task of storing and prioritizing the patients according to the GCS scale be generating a ready queue which is then passed on the activity scheduler function which further allots them to the corresponding wards.

# LITERATURE REVIEW

Day after day rate of accidents and mass casualties has been raised that results in a huge damage to people and their loved ones. That is why, it is very necessary to save the precious lives of these people with limited available resources in a fast and methodical manner in order to minimize the loss of life caused to the patients.

The damage is more severe when traumatic brain injuries are involved and multiple patients have to be handled in parallel. Here there is a need to prioritize between the patients on the basis of their criticality. In order to tackle this difficulty, the Glasgow coma scale was introduced.

GLASGOW COMA SCALE –FOR ADULTS

**TABLE 1.1**

**GLASGOW COMA SCALE –FOR ADULTS**

|  |  |  |
| --- | --- | --- |
| **FEATURE** | **RESPONSE** | **SCORE** |
| **BEST EYE RESPONSE** | Open Spontaneously | 4 |
|  | Open to Verbal Command | 3 |
|  | Open to Pain | 2 |
|  | No Eye Opening | 1 |
| **BEST VERBAL RESPONSE** | Oriented | 5 |
|  | Confused | 4 |
|  | Inappropriate Words | 3 |
|  | Incomprehensible Sounds | 2 |
|  | No Verbal Response | 1 |
| **BEST MOTOR RESPONSE** | Obeys Commands | 6 |
|  | Localizing Pain | 5 |
|  | Withdrawal from Pain | 4 |
|  | Flexion to Pain | 3 |
|  | Extension to Pain | 2 |
|  | No Motor Response | 1 |

**TABLE 1.2**

**PEDIATRIC GLASGOW COMA SCALE –For children.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FEATURES** | **>1 YEAR** | | **< 1 YEAR** | **SCORE** |
| **EYE OPENING** | Spontaneously | | Spontaneous | 4 |
|  | To Verbal Command | | To Shout | 3 |
|  | To Pain | | To Pain | 2 |
|  | No Response | | No Response | 1 |
| **MOTOR RESPONSE** | Obeys | | Spontaneous | 6 |
|  | Localizes Pain | | Localizes Pain | 5 |
|  | Flexion Withdrawal | | Flexion Withdrawal | 4 |
|  | Flexion Abnormal(decorticate rigidity\_) | | Flexion Abnormal(decorticate rigidity\_) | 3 |
|  | Extension (decerebrate rigidity) | | Extension (decerebrate rigidity) | 2 |
|  | No Response | | No Response | 1 |
| **VERBAL RESPONSE** | **>5 YEARS** | **2-5 YEARS** | **0-23 MONTHS** |  |
|  | Oriented | Appropriate words/Phrases | Smiles/Coos Appropriately | 5 |
|  | Disoriented/Confused | Inappropriate words | Cries and is Consolable | 4 |
|  | Inappropriate words | Persistent Cries/Screams | Persistent Inappropriate Crying | 3 |
|  | Incomprehensible sounds | Grunts | Grunts, Agitated and restless | 2 |
|  | No Response | No Response | No Response | 1 |

The Glasgow Coma Scale is different for both children and adult. This is because the verbal and motor response of a child is not as well developed as an adult.

According to this scale, patients are assessed based on movement, speech and eye opening. After the examination, the values given corresponding to the patient’s actions to the 3 responses are summed up.

The classification of brain injury is classified as-

Severe - (GCS <=8)

Moderate - (GCS 9-12)

Mild - (GCS >=13)

The lowest GCS is 3 which means the patient is in deep comma or is dead. The patients with GCS score 3-8 are considered to be in comma. The highest GCS is 15 which indicates that the patient is completely oriented and alert. In both these cases the patient is given low priority for treatment. The ones who are in the range of 8 to 14 are given high priority.

After prioritizing the patient’s wards are allotted to them and the whole process takes place in a chronological manner. Here, Priority scheduling and Activity selection algorithms are used for the same purpose.

**Case Study:**

In Sassoon General Hospitals, Pune, 45 of the patients admitted in the general surgery ward were selected. They had traumatic brain injury. 36 patients were above 50 years of age while 9 were below 9. The GCS score of 7 patients was between 3 to 5, 12 patients was 6 to 9 and 26 patients was between 10 to 15. Analysis was made for all these 45 people and it was found that the all the parameters included in the GCS scale are independent predictors of the patient’s survival after a traumatic brain injury.

# OBJECTIVES

Natural calamities, disaster or terrorist bombing, splash a sudden spike in demand for the emergency response in the affected area. As the resources are limited in such an environment, it becomes pivotal to allocate the available limited resources to the patients in the best possible way. The initial goal is to provide immediate care to those who can be saved and need immediate lifesaving interventions.

**METHODOLOGY**

**Mass Casualties incidents like earthquake, train accidents, terrorist attacks etc. lead to increment in number of patients, that need treatment as soon as possible. This may lead to overwhelming of available resources very rapidly.**

**The project focuses on solving the problem of delivering these deficit resources to the patients in the best possible manner.**

* The system works on the basis of Glasgow coma scale and the very first task of it is to collect the number of patients affected.
* A child is given priority over an adult thus it is essential for the system to mark the number of children. This leads to the use of Data Structures where the child patients are stored in an array.
* Each child is examined and on basis of his/her response score from the pediatric Glasgow coma scale is assigned and summed up.
* On the basis of the summed up score the children are placed into three categorize- Severe, Moderate and Mild. Three lists or containers are created to store the following category separately.
* Inside the array of Moderate, Mild or Severe Category the Patient is prioritized and sorted on the basis of their summed up score.

E.g.- Three children with PGCS score as 11,10,9 respectively, the one with the least score is given highest priority.

* The main Data Structure used is the ready queue, all the patients will we stored in it after sorting, based on prioritization.
* The children having the moderate type are given the highest priority, thus are flushed into the ready queue i.e. our original queue from the array.
* The adult patients are considered, the same method is followed to place them into Mild, Moderate and Severe groups on the basis of Glasgow Coma Scale, later all three categorizes are prioritized and stored in three arrays separately.
* The first priority was given to the Children in moderate group. The next priority is provided to the adult in the same group.

Thus, the adult (moderate) victims are pushed into the original ready queue that is after the children.

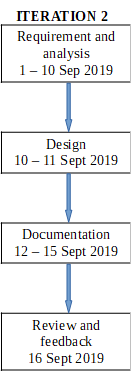
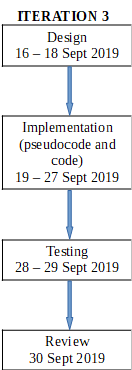
* Now the priority followed in the ready queue will be child mild over adult mild and child severe over adult severe.
* Therefore, the next group to be in the ready queue will be child mild followed by adult mild. Similarly done for the severe case.
* The previous arrays will be flushed and cleared from the screen.
* The screen is left with the ready queue containing all the affected patients in a prioritized manner.
* Original queue is passed to the Activity Scheduler function.
* The first four patients of the queue will be allocated to the wards by the Activity Scheduler.
* The Scheduler will ask for the free ward next, as soon as the ward is emptied, next patient will be granted to the respective ward.
* The whole process moves forward in a **sequential automated** manner.
* The system will provide resources to the patients who are in danger and whose life can be saved.

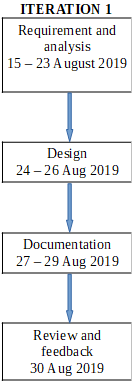
**System Requirements (Software/Hardware)**

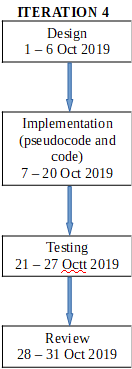
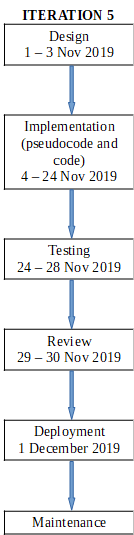
**Hardware Interface:**

* 32 bits processor architecture supported by windows.
* Minimum RAM requirement for proper functioning is 512 MB. Software Interface:
* This system is developed in C programming language.
* G++ compiler

# PERT Chart







# Source Code

# #include <stdio.h>

# #define MAX 50

# struct structure

# {

# int gcs\_sum;

# int patient\_id;

# };

# void insert(int);

# int length\_of\_queue = 0;

# int delete(int);

# int queue\_array[MAX];

# void ward\_allotment(int queue\_array[MAX]);

# int rear = - 1;

# int front = - 1;

# void insert(int add\_item)

# {

# if (rear == MAX - 1)

# printf("Queue Overflow \n");

# else

# {

# if (front == - 1)

# /\*If queue is initially empty \*/

# front = 0;

# rear = rear + 1;

# queue\_array[rear] = add\_item;

# length\_of\_queue+=1;

# }

# } /\* End of insert() \*/

# int delete(int ward\_no)

# {

# if (front == - 1 || front > rear)

# {

# printf("Queue Underflow \n");

# return 0;

# }

# else

# {

# front = front + 1;

# return queue\_array[front-1];

# }

# } /\* End of delete() \*/

# void ward\_allotment(int queue\_array[MAX])

# {

# int q[50];

# int i,n, ward\_no[4];

# label:

# printf("\n");

# printf("Number of wards empty: "); //we assume we have 4 wards only

# scanf("%d",&n);

# if(n!=0)

# printf("Enter the ward number of %d wards which are empty(Ward 0,1,2,3):\n",n); // ward 1,2,3 or 4 (assistant will enter this)

# for(i = 0; i<n; i++)

# {

# scanf("%d",&ward\_no[i]);

# q[i] = delete(ward\_no[i]);

# }

# if(n!=0)

# {

# printf("\n");

# printf("|----------------------------------|\n");

# printf("| Patient No. |Ward No. allotted|\n");

# printf("|----------------------------------|\n");

# for(i = 0; i<n; i++)

# printf("| %d | %d |\n",q[i],ward\_no[i]);

# printf("|----------------------------------|\n");

# printf("\n\n");

# }

# 

# if(front<=rear)

# {

# printf("waiting queue of patients:\n");

# printf(" Patient Id\n");

# for(i = front; i<=rear; i++)

# {

# printf("|-----------|\n");

# printf("| %d |\n", queue\_array[i]);

# }

# printf("|-----------|\n");

# 

# sleep(6);

# goto label;

# }

# 

# }

# int main()

# {

# int victims,i;

# printf("\n +++++++++++++\n");

# printf(" + Welcome +\n");

# printf(" +++++++++++++\n\n");

# printf("Enter how many total victims are there:");

# scanf("%d",&victims);

# printf("\n");

# 

# struct structure patients[victims];

# struct structure severe[victims+1];

# struct structure moderate[victims+1];

# struct structure mild[victims+1];

# 

# 

# int eye\_open,motor,verbal\_response, sum;

# int j, temp1, temp2;

# int index1=0, index2=0, index3=0; //number of patients in severe, moderete and mild queues

# 

# 

# //patient number and GCS sum

# for(i = 0; i<victims;i++)

# {

# printf("Enter the details of patient number %d:\n", i);

# printf("Scale of eye opening:");

# scanf("%d",&eye\_open);

# printf("Scale of motor response:");

# scanf("%d",&motor);

# printf("Scale of verbal response:");

# scanf("%d",&verbal\_response);

# sum = eye\_open + motor + verbal\_response;

# patients[i].gcs\_sum = sum;

# patients[i].patient\_id = i;

# printf("\n");

# 

# }

# 

# 

# 

# for(i = 0 ;i<victims-1; i++) // sorting

# {

# for(j = 0 ;j<victims-i-1; j++)

# {

# if (patients[j].gcs\_sum > patients[j+1].gcs\_sum)

# {

# temp1 = patients[j].gcs\_sum;

# temp2 = patients[j].patient\_id;

# patients[j].gcs\_sum = patients[j+1].gcs\_sum;

# patients[j].patient\_id = patients[j+1].patient\_id;

# patients[j+1].gcs\_sum= temp1;

# patients[j+1].patient\_id= temp2;

# }

# }

# }

# 

# printf("Final sorted list:\n"); //print final sorted queue

# printf("patient\_id\tgcs\_sum\n");

# for(i = 0 ;i<victims; i++)

# {

# printf("%d \t\t%d\n", patients[i].patient\_id, patients[i].gcs\_sum);

# 

# }

# 

# 

# for(i=0; i<victims; i++) //queues for moderete

# {

# if(patients[i].gcs\_sum>=9 && patients[i].gcs\_sum<=12)

# {

# moderate[index2].gcs\_sum = patients[i].gcs\_sum;

# moderate[index2].patient\_id = patients[i].patient\_id;

# index2 = index2 + 1;

# insert(patients[i].patient\_id);

# }

# }

# 

# for(i=0; i<victims; i++) //queues for mild

# {

# if(patients[i].gcs\_sum>=13)

# {

# mild[index3].gcs\_sum = patients[i].gcs\_sum;

# mild[index3].patient\_id = patients[i].patient\_id;

# index3 = index3 + 1;

# insert(patients[i].patient\_id);

# }

# 

# }

# 

# 

# for(i=0; i<victims; i++) //queues for severe

# {

# if(patients[i].gcs\_sum<=8)

# {

# severe[index1].gcs\_sum = patients[i].gcs\_sum;

# severe[index1].patient\_id = patients[i].patient\_id;

# index1 = index1 + 1;

# insert(patients[i].patient\_id);

# }

# 

# }

# 

# 

# printf("\n"); // print different queues

# printf("\nsevere:\n");

# printf("patient\_id\tgcs\_sum\n");

# for(i = 0 ;i<index1; i++)

# {

# printf("%d \t\t%d\n", severe[i].patient\_id, severe[i].gcs\_sum);

# }

# 

# printf("\n");

# printf("\nmoderate:\n");

# printf("patient\_id\tgcs\_sum\n");

# for(i = 0 ;i<index2; i++)

# {

# printf("%d \t\t%d\n", moderate[i].patient\_id, moderate[i].gcs\_sum);

# }

# 

# printf("\n");

# printf("\nmild:\n");

# printf("patient\_id\tgcs\_sum\n");

# for(i = 0 ;i<index3; i++)

# {

# printf("%d \t\t%d\n", mild[i].patient\_id, mild[i].gcs\_sum);

# }

# 

# 

# ward\_allotment(queue\_array);

# printf("\nNo more Patients\n");

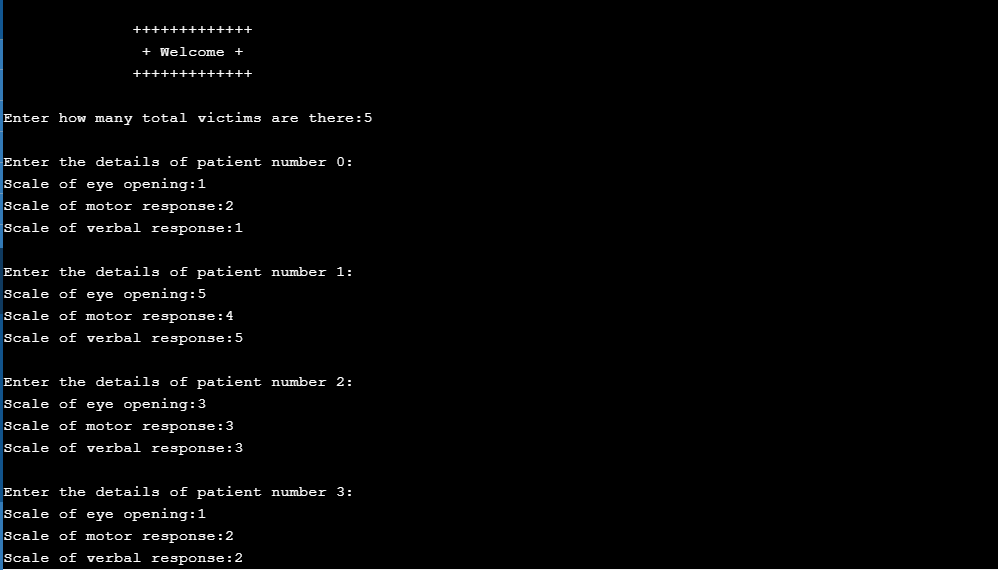
# printf("-----THANK YOU------");

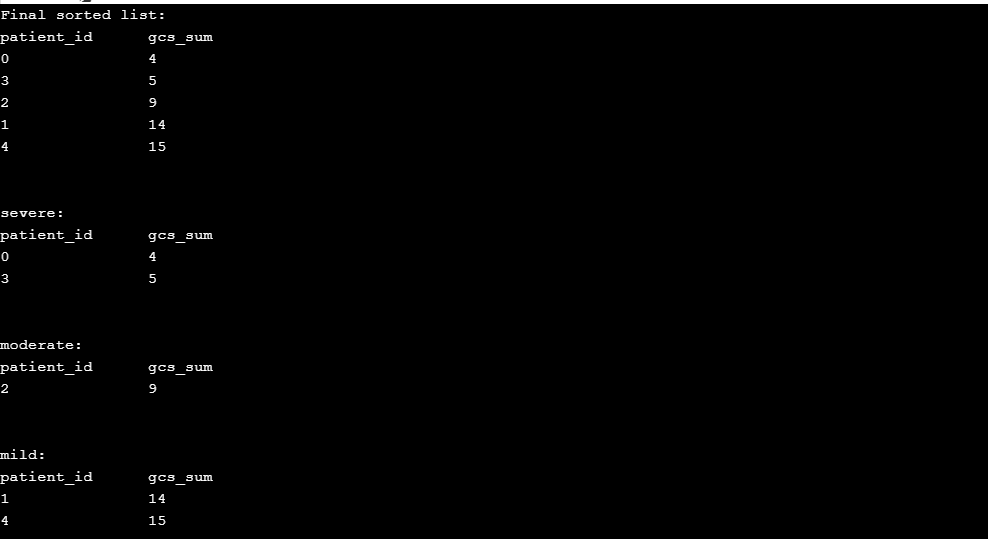
# 

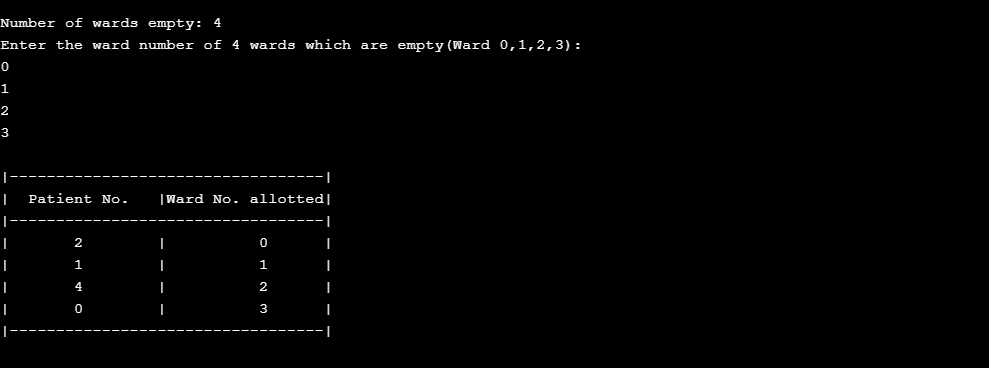
# }

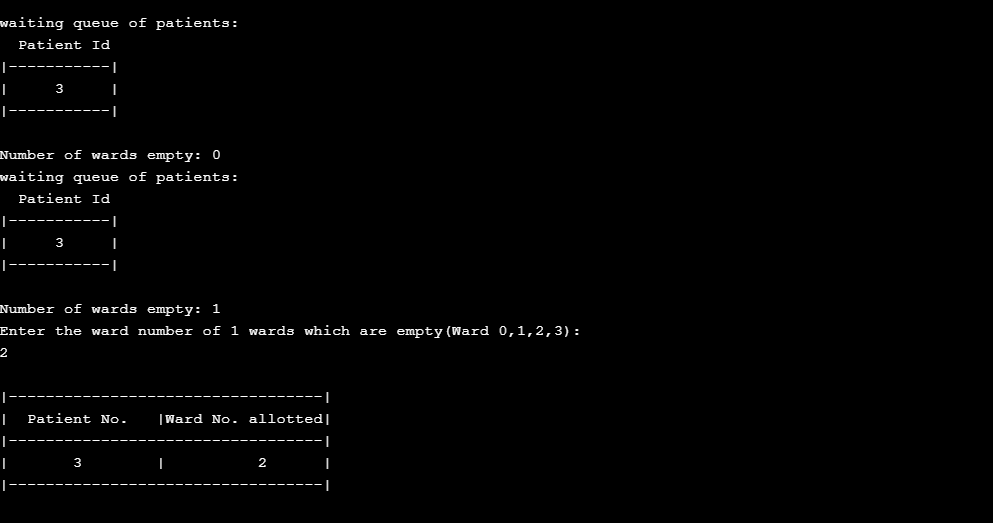
# Output

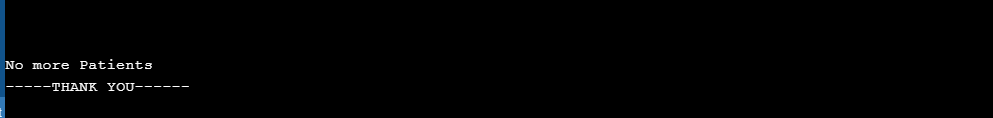
# 











# LIMITATIONS

* This project is applicable for small scale hospitals where medical resources and doctors are limited.
* We have assumed that in each ward only one patient can be treated at a time.
* This project is only for trauma patients of triage category.
* In this project the user has to input the values based on Glass Gow comma scale.

# REFERENCES

[1] Committee of Trauma, American College of Surgeons. Resources for Optimal Care of the Injured Patient. 6th ed. Chicago,2014.

[2] Traumatic Brain Injury in the United States: Emergency Department Visits, Hospitalizations and Deaths 2002–2006. Atlanta: Centers for Disease Control and Prevention. National Center for Injury Prevention and Control; 2010. http://www.cdc.gov/traumaticbraininjury/pdf/blue\_book.pdf.

[3] Effective Patient Prioritization in Mass Casualty Incidents using Hyperheuristics and the Pilot Method. http://www.lcc.uma.es/~ccottap/papers/cotta11effective.pdf

[4] Assessment of coma and impaired consciousness. A practical scale. Lancet. 1974 Jul 13;2(7872):81–4. [PubMed]

[5] Singh B, Murad MH, Prokop LJ, et al. Meta-analysis of Glasgow Coma Scale and Simplified Motor Score in predicting traumatic brain injury outcomes. Brain Inj. 2013;27(3):293–300. [PubMed] [CrossRef]