

Hospital Management System

A Minor Project Report

Submitted in partial fulfillment of the requirements for the
Award of degree of Bachelor of Computer applications (BCA)

2021-2024



Submitted by

Amandeep Singh 0211BCA038

Guided by

Ms. Preeti Taneja



STUDENT UNDERTAKING

This is to certify that the project titled Hospital Management System submitted to Bharati Vidyapeeth (Deemed to be University), Pune in partial fulfillment of the requirement for the award of the degree of Bachelor of Computer Applications is an original work carried out under the guidance of Ms. Preeti Taneja. The matter embodied in this project is a genuine work done by me and has been submitted neither to this University nor to any other University for the fulfillment of the requirement of the course of study.

Amandeep Singh

ERPID : 0211BCA038

PRN No:2120100272



**Bharati Vidyapeeth
(Deemed to be University)**
Institute of Management & Research, New Delhi
A Grade Status Awarded by MHRD, Govt of India, re-accredited with Grade A+ by NAAC
An ISO 9001:2015 14001:2015 Certified Institute
A-4, Paschim Vihar, New Delhi – 110063

Ref:

Date:

CERTIFICATE FROM GUIDE

This is to certify that the minor project report titled “**Hospital Management System**” Submitted to Bharati Vidyapeeth (Deemed to be University) Institute of Management and Research, New Delhi in partial fulfillment of the requirement for the award of the Bachelor of Computer applications is an original work carried out by “**Amandeep Singh**” under my guidance. To the best of my knowledge and belief the matter embodied in this project is genuine work done by the student and has been submitted neither to this University nor to any other University for the fulfillment of the requirement of the course of study.

Ms. Preeti Taneja
(Project Guide)



Bharati Vidyapeeth

(Deemed to be University)

Institute of Management & Research, New Delhi

A Grade Status Awarded by MHRD, Govt of India, re-accredited with Grade A+ by NAAC

An ISO 9001:2015 14001:2015 Certified Institute

A-4, Paschim Vihar, New Delhi – 110063

Ref:

Date:

CERTIFICATE FROM DIRECTOR

This is to certify that the Project titled “**Hospital Management System**” is an academic work done by “**Amandeep Singh**” submitted in partial fulfillment of the requirement, for the award of the Degree of Bachelor of Computer applications, from **Bharati Vidyapeeth (Deemed to be University), Pune**. It has been completed under the guidance of Ms. Preeti Taneja.

The authenticity of the project work will be examined by the viva-voce examiner, which includes data verification, checking duplicity of information etc., and it may be rejected due to nonfulfillment of quality standards set by the Institute.

Dr. Yamini Agarwal

Director

ACKNOWLEDGEMENT

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report. A special thanks to our Minor project Guide Ms. Preeti Taneja whose help, stimulating suggestions and encouragement, helped me to coordinate our project especially in writing this report topic and achieving the goal as well as his encouragement to maintain our progress in track. I would appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills by their comment and tips.

Index

CHAPTER 1 INTRODUCTION		Page no.
1.1	Introduction about Project	2
1.2	Need of Computerization of System/ Problem in existing system	3
1.3	Proposed Software (What would S/W accomplish?)	3
1.4	Importance of the Work/project	4
 CHAPTER 2 SYSTEM ANALYSIS		
2.1	Analysis Methodology/ Requirement gathering techniques	6
2.2	Feasibility Study	7
2.3	Choice of Platforms	9
2.3.1	Software used	9
2.3.2	Hardware used	9
 CHAPTER 3 SYSTEM DESIGN		
3.1	Process Model used	12
3.2	Database Design (ERD/Data Dictionary/ Table Design)	13
3.3	Functional Design (DFD)	14
3.4	Interface Design (Screen shots of forms)	16
3.5	Output Design (Screen Shots of Report)	41

CHAPTER 4 TESTING AND IMPLEMENTATION

4.1	Testing Methodology (Types)	65
4.2	Testing Methodology applied (Why)	66
4.3	Test Cases	66
4.4	Test Data	73
4.5	Gap Analysis (Planned Vs Achieved)	73
4.6	Rework/ Retest	73
4.7	Hardware & Software Requirement	75
4.7.1	Hardware Requirement	75
4.7.2	Software Requirement	75

CHAPTER 5 CONCLUSION AND REFERENCES

5.1	Conclusion	77
5.2	Limitation of System	77
5.3	Future Scope for Modification	78
5.4	References/ Bibliography	78

ANNEXURES

A-1	Menu Flow Diagram	80
A-2	Sample Input	81
A-3	Sample Output	83
A-4	Program Code	
A-5	Mentor Feedback Report	
A-6	Plagiarism certificate	

CHAPTER 1

INTRODUCTION

- 1.1 Introduction**
- 1.2 Present state of the art**
- 1.3 Problem in existing system**
- 1.4 Proposed Software**
- 1.5 Importance of the Work/project**

1.1 Introduction about Project-

This project introduces HOSPITAL MANAGEMENT SYSTEM. It explains how Patients, Doctors, Staff, Medicines and Billing records are added, modified, listed, searched or deleted. The project is developed in c language and some other files are also used in this project. The project is well maintained by programming. Hospital Management System can easily be accompanied with the help of this.

It aims at developing software that can be used to easily manually do the tasks of Adding, Editing, Deleting, Searching or Listing Patients, Doctors, Staff, Medicines and Bill records. The main merit of this project is that it converts all manual work which is time taken and error taken to fully automatic system which helps in eliminating all the paper work, saves time and improves customer services thus providing convenience to the workers as well as customers.

The project starts with 5 modules namely Patients, Doctors, Staff, Medicines and Billing. For every module that the user choose there are several functionality such as Add, Edit, List, Search or Delete a record/records.

1.2 Present state of the art –

Hospital management system is an attempt to make the basic concepts of modification of Patients, Doctors, Staff, Medicines and Bill records. The system enables to perform the following functions:

- Add Patients, Doctors, Staff, Medicines and Bill records
- Editing Patients, Doctors, Staff and Medicines records
- List Patients, Doctors, Staff, Medicines and Bill records.
- Search Patients, Doctors, Staff, Medicines and Bill records
- Delete Patients, Doctors, Staff and Medicines records

- Patient/Customer Billing System
- Project helps to provide flexibility due to which everyone can operate it easily and efficiently.
- Project also provides a complete set of solutions for some common and specific areas of work in the Patients, Doctors, Staff, Medicines and Bill records.

1.3 Need of Computerization of System/ Problem in existing system-

In existing system, all the work is done manually, so it takes more time by the users. Another disadvantage of the existing system is that search, delete, add, edit done manually is difficult and time consuming. This will take more time, currently a one day process for verifying all records. So after conducting the feasibility study we decide to make a computerized Hospital management system.

1.4 Proposed Software

The purpose of this source is to describe the Hospital management system which provides all the Patients, Doctors, Staff, Medicines and Bill Details.

- It adds records
- It views all records
- It searches any records
- It modifies any records
- It delete any records

Of Patients, Doctors, Staff, Medicines and Bills.

1.5 Importance of the work/project-

The project is very simple and easy in design. The project requires very low system properties. The system will work in almost all patterns.

It has the following importance:

Enhancement:

Project concentrations to improve and upgrading its efficiency and effectiveness.

Automation:

Project automates each and every activity of the manual system and increases its throughput.

Accuracy:

Project provides user accurate result when the user required it.

User-Friendly:

Project has a very user-friendly interface. User can easily work on the project. The software provides accuracy result.

Maintenance Cost:

Reduce the cost of maintenance.

CHAPTER 2

SYSTEM ANALYSIS

- 2.1 Analysis Methodology /requirement gathering techniques**
- 2.2 Feasibility Study**
- 2.3 Choice of Platforms**
 - 2.3.1 Software used**
 - 2.3.2 Hardware used**

2.1 Analysis Methodology /requirement gathering techniques-

It is the task of creating a list of requirements (functional system, technical, etc.) from the various end user like (customers, users, vendors, IT staff, etc.) that will be used as the basis for the requirements definition. It is an important part of any project and project management.

There are many types of requirement gathering techniques. Some of them are:

1. One-on-one interviews

The most common technique for gathering requirements is to sit down with the client/customer and ask them what they need.

2. Group interviews

Group interviews are similar to one-on-one interviews, except that more than one person is being interviewed-usually two or four. These interviews work well when all person is at the same level or has the same role.

3. Facilitated sessions

In a facilitated session, you bring a larger group (five or more) together for a common reason or purpose. Here we are trying to gather a set of common requirements from the group in a faster manner than if you were to interview each of them separately.

4. Questionnaires

Questionnaires are used to capture the requirements from end users in remote locations or those who will have only major input into the complete requirements. Questionnaires can also be used when you have to collect data from dozens, hundreds, or thousands of people.

5. Use cases

Use cases are basically stories that describe how individual processes work. The stories include people and describe how the solution works from the user

perspective. Although the use cases may need to be distilled later into the more particular detailed requirements.

6. Brainstorming

On few projects, the requirements are not “uncovered” as much as they are “discovered”. In other words, the solution is brand new and needs to be developed as a set of ideas that people can agree to. After all the ideas are developed, the participants prioritize the ones they think are the best for this solution. The resulting consensus of best ideas is used for the basic requirements.

Note: In our Project we take 100 one to one interviews and Questionnaires

1. In one to one interviews we will ask some questions related to our Software project from different types of workers like the Hospital’s Doctors, Staff such as Receptionists, etc.
2. In Questionnaires we will also ask about 10 questions related to our software project.

We will do this task in our project so that efficiency and effectiveness of our project will increase.

2.2 FEASIBILITY STUDY

As we all know that each and every project needs to have a feasibility study for the complete understand ability of the entire system. Basically, it is an assessment of the practicality of a proposed plan or system or method.

A project feasibility study is a comprehensive report that examines in detail about the system. It is an evaluation of a proposed system or project to determine if it

- (1) is technically feasible
- (2) is feasible within the estimated cost
- (3) will be profitable.

A feasibility study is an initial study understands before the real work of a project starts to certain the likelihood of the project success.

2.2.1 ECONOMIC FEASIBILITY

The procedure is to determine the benefits and savings that are expected from a customer side and compare it with the costs. If your profits outweighs costs, then the decision is made to design and put into the system. We have designed a software through which we can maintain the people and item records so manual labour cost is reduced. Otherwise further changes are made in the proposed system.

There are some points which tell how our project is economic feasibility is possible:

- It will reduce Manpower cost
- It will increase our profits by making this software

2.2.2 TECHNICAL FEASIBILITY

Technical feasibility is one of the earliest studies that must be regulate after the project has been identified. In large engineering projects consulting agencies that have abuned staffs of engineers and technicians conduct technical studies dealing with the projects. Our project is technically feasible as the user can perform various search operations with different search criteria and this will ultimately reduce the time and increase the efficiency. In this intermediate labour cost will reduced and improves staff efficiency.

There are some points which tell how our project's Technical feasibility is possible:

1. In our software we can easily Add, Edit, Delete, Search or List the records of Patients, Doctors, Staff, Medicines and Bills.
2. In our software we can give our patient/customer a computerized bill.

2.2.3 OPERATIONAL FEASIBILITY

It is a measure of how accurately a system fix the problems, and takes merit of the opportunities find during scope definition and how it assures the requirements recognize in the requirements analysis phase of system development. In this, manual efforts will be reduced by providing this software.

There are some points which tell how our project's operational feasibility is possible:

1. In this software our manual efforts will reduce by directly searching the needed patient/doctor/staff records. It also helps to maintain the stock of items or medicines.
2. In this software our manpower cost of doing all works manually will reduce time or cost.

2.3 CHOICE OF PLATFORM

2.3.1 HARDWARE REQUIREMENTS

Processor	: Intel Core i5 4400@3.1 Ghz
Clock speed	: 1000 Mhz
System bus	: 64 bits
RAM	: 8 GB of RAM
HDD	: 1 TB
Monitor	: any
Keyboard	: 108 keys
Mouse	: 2 button mouse

2.3.2 SOFTWARE REQUIREMENTS

OS	:	MS WINDOWS 10
Front End	:	C Programming

CHAPTER 3

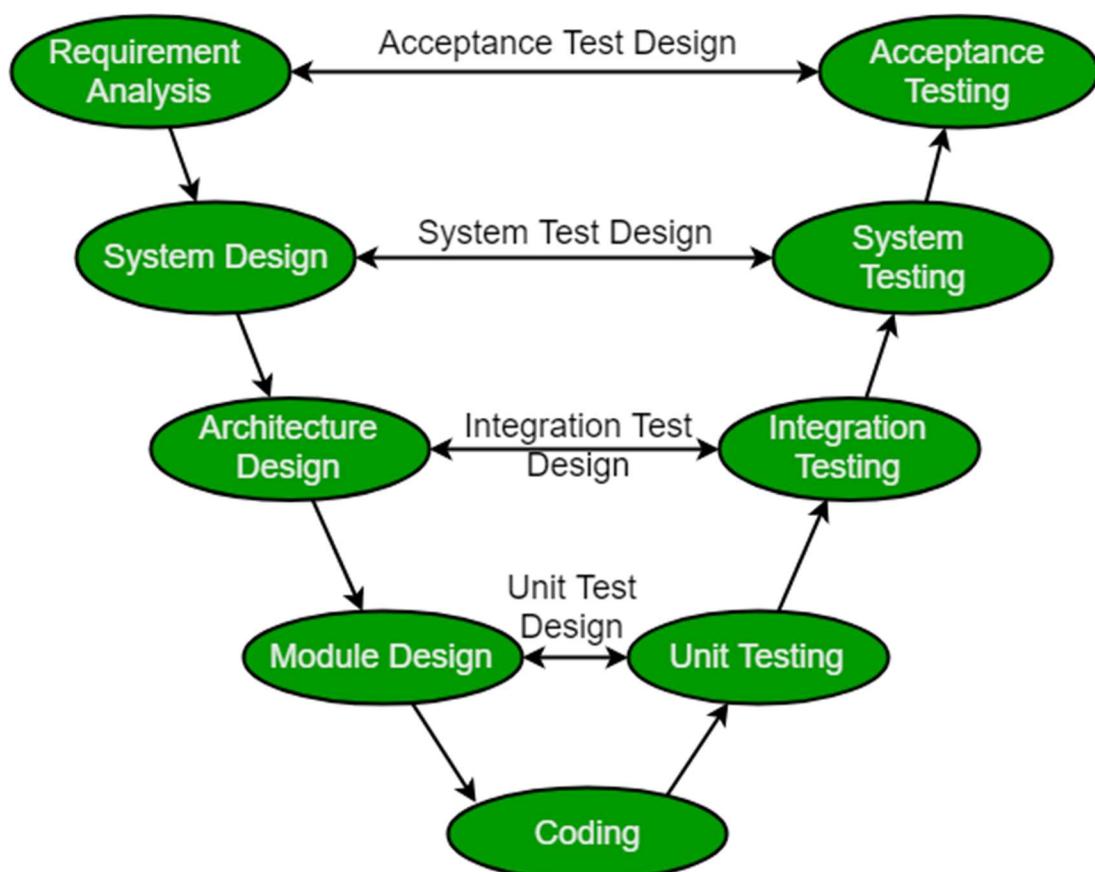
SYSTEM DESIGN

- 3.1 Process Model used**
- 3.2 Database Design**
- 3.3 Functional Design (DFD)**
- 3.4 Interface Design**
- 3.5 Output Design**

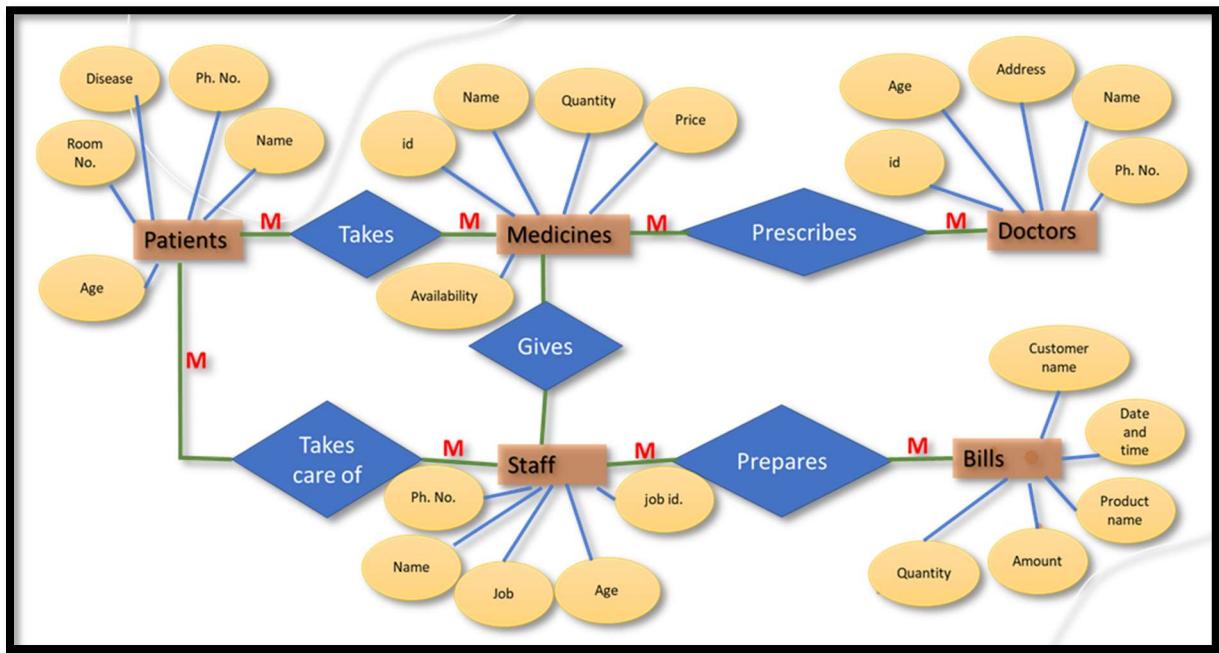
3.1 Process Model used

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

In our project, we use this model because requirements were clearly defined and fixed. Also we wanted to make an efficient system by checking for errors and defects at early stages.



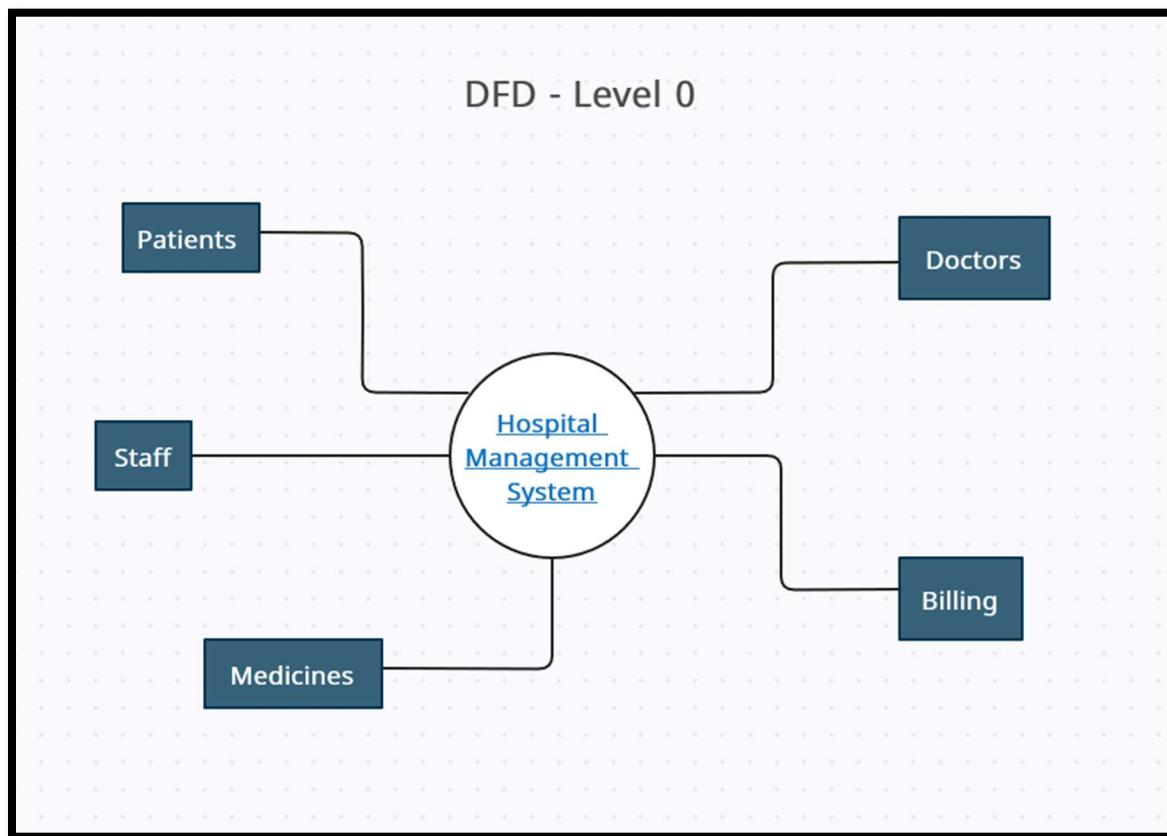
3.2 Database Design (ER Diagram)



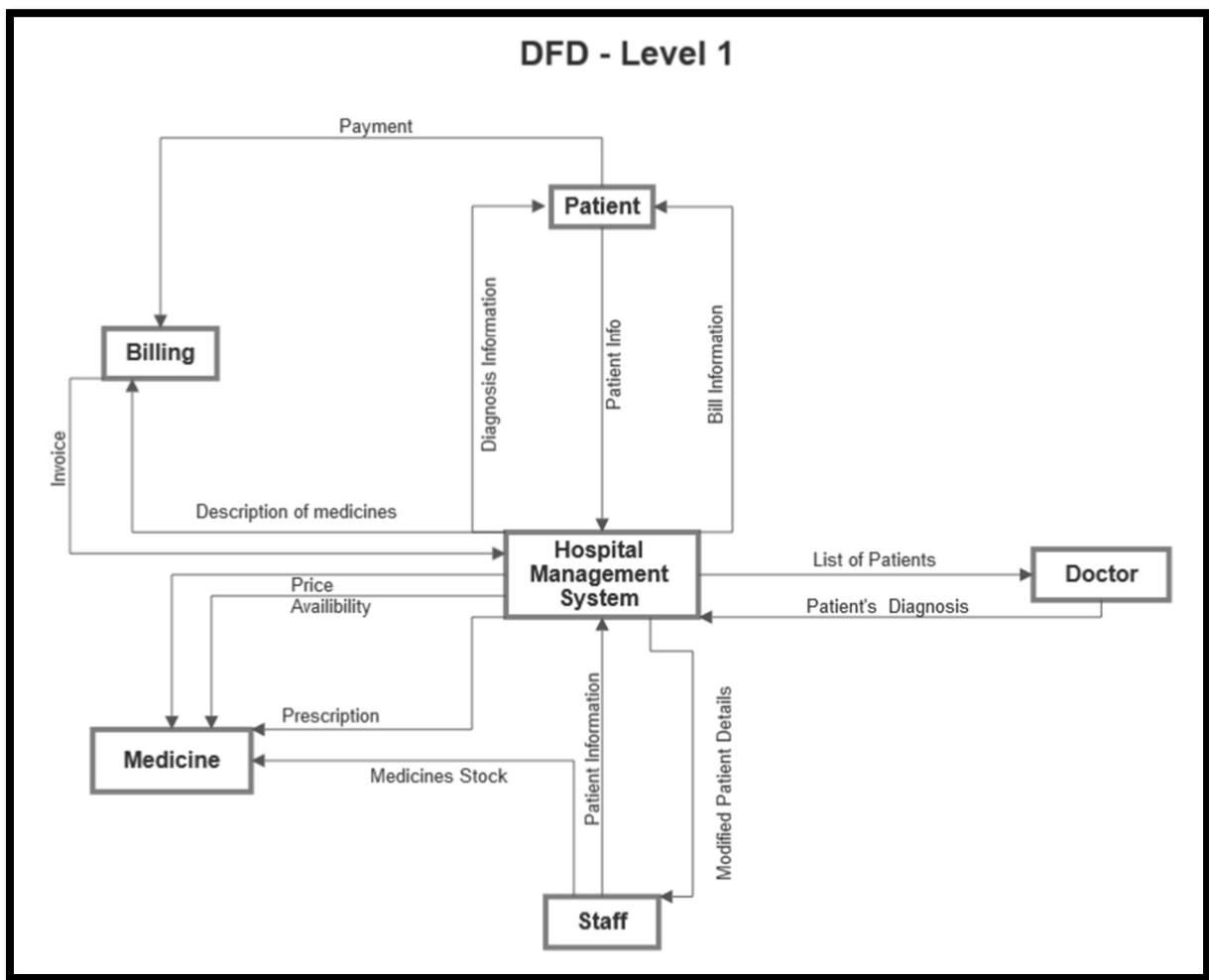
3.3 Functional Design (Data Flown Diagram)

A data flow diagram (DFD) is a graphic representation of the "flow" of data through an information system, modelling its process aspect. A DFD is often used as a initial step to create an overview of the system, which can later be elaborated.

0 Level DFD



1st Level DFD



3.4 Input Design

```
C minor_main.c minor_main.c...
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 > struct p...
11 } v[100];
12 void patient_read();
13 void patient_add();
14 void patient_view();
15 void patient_search();
16 void patient_edit();
17 void delete_patient();
18
19 > struct d...
25 } w[100];
26 void read_doctor();
27 void add_doctor();
28 void view_doctor();
29 void search_doctor();
30 void edit_doctor();
31 void delete_doctor();
32
33 > struct s...
39 } x[100];
40 void read_staff();
41 void add_staff();
42 void view_staff();
43 void search_staff();
44 void edit_staff();
45 void delete_staff();
46
47 > struct m...
52 } y[100];
53 void read_medicine();
54 void add_medicine();
55 void view_medicine();
56 void search_medicine();
57 void edit_medicine();
```

```
C minor_main.c minor_main.c\\.. ▶ ✎ ⚙ ☰ 🗃 ...  
94     printf("%.2f\\t\\t", qty * price);  
95     printf("\\n");  
96 }  
97 void generateBillFooter(float total)  
98 {  
99     printf("\\n");  
100    float dis = 0.1 * total;  
101    float netTotal = total - dis;  
102    float cgst = 0.09 * netTotal;  
103    float sgst = 0.09 * netTotal;  
104    float grandTotal = netTotal + cgst + sgst;  
105    printf("-----\\n");  
106    printf("Sub Total\\t\\t\\t%.2f", total);  
107    printf("\\nDiscount @10%\\t\\t\\t%.2f", "%", dis);  
108    printf("\\n\\t\\t\\t\\t-----");  
109    printf("\\nNet Total\\t\\t\\t%.2f", netTotal);  
110    printf("\\nCGST @9%\\t\\t\\t%.2f", "%", cgst);  
111    printf("\\nSGST @9%\\t\\t\\t%.2f", "%", sgst);  
112    printf("\\n-----");  
113    printf("\\nGrand Total\\t\\t\\t%.2f", grandTotal);  
114    printf("\\n-----\\n");  
115 }  
116  
117 int n, i, j = 0, a = 0, sum = 0, g, flag, num;  
118  
119 int main()  
120 {  
121     int module = 1;  
122     printf("*****\\n");  
123     printf("*****\\n");  
124     printf("**\\n");  
125     printf(" Hospital Management System      \\n");  
126     printf(" **\\n");  
127     printf("*****\\n");  
128     printf("*****\\n");  
129     int ch;  
130     do  
131     {
```

C minor_main.c minor_main.c...

```
131 {
132     printf("1. Patient\n2. Doctor\n3. Staff\n4. Medicine\n5. Billing\n6. Exit\nPlease enter the module you want to access = ");
133     scanf("%d", &ch);
134     if (ch == 1)
135     {
136         int n, i, j = 0, a = 0, sum = 0, g, flag, num;
137         patient_read();
138         int c, q;
139         int m;
140
141         while (c != 6)
142         {
143
144             printf("\n\nEnter your choice\n\n1. Add patient Information\n2. View patient Information\n3. Search patient\n4. Edit patient Information\n5.
Delete patient Information\n6. Exit Patient\n\nOption=");
145             scanf("%d", &c);
146             fflush(stdin);
147             if (c == 1)
148             {
149                 patient_add();
150             }
151             else if (c == 2)
152             {
153                 patient_view();
154             }
155             else if (c == 3)
156             {
157                 patient_search();
158             }
159             else if (c == 4)
160             {
161                 patient_edit();
162             }
163             else if (c == 5)
164             {
165                 delete_patient();
166             }
167             else if (c == 6)
```

```
C minor_main.c minor_main.c main()
167     else if (c == 6)
168     {
169         patient_write();
170     }
171     else
172     {
173         printf("\n\nInvalid input , try again by using valid inputs");
174     }
175     printf("\n\n");
176 }
177 }

178 }

179 else if (ch == 2)
180 {
181     int j = 0, a = 0, sum = 0, g, flag, num;
182     read_doctor();
183     int c, i, q;
184     int m, n;
185
186     while (c != 6)
187     {
188
189         printf("Enter your choice\n\n1. Add Doctor Information\n2. View Doctor Information\n3. Search Doctor Information\n4. Edit Doctor Information\n5.
190 Delete Doctor Information\n6. Exit Doctor\n\nOption=");
191         scanf("%d", &c);
192         fflush(stdin);
193         if (c == 1)
194         {
195             add_doctor();
196         }
197         else if (c == 2)
198         {
199             view_doctor();
200         }
201         else if (c == 3)
202         {
203             search_doctor();
204         }
205         else if (c == 4)
206         {
207             edit_doctor();
208         }
209         else if (c == 5)
210         {
211             delete_doctor();
212         }
213         else if (c == 6)
214         {
215             write_doctor();
216         }
217         else
218         {
219             printf("\n\nInvalid input, try again by using valid inputs.");
220         }
221         printf("\n\n");
222     }
223 }

224 }

225 else if (ch == 3)
226 {
227     int j = 0, a = 0, sum = 0, g, flag, num;
228     read_staff();
229     int c, i, q;
230     int m, n;
231
232     while (c != 6)
233     {
234
235         printf("Enter your choice\n\n1. Add Staff Information\n2. View Staff Information\n3. Search Staff\n4. Edit Staff Information\n5. Delete Staff
236 Information\n6. Exit Staff\n\nOption=");
237         scanf("%d", &c);
238         fflush(stdin);
239     }
240 }
```

```
C minor_main.c minor_main.c main()
202     {
203         search_doctor();
204     }
205     else if (c == 4)
206     {
207         edit_doctor();
208     }
209     else if (c == 5)
210     {
211         delete_doctor();
212     }
213     else if (c == 6)
214     {
215         write_doctor();
216     }
217     else
218     {
219         printf("\n\nInvalid input, try again by using valid inputs.");
220     }
221     printf("\n\n");
222 }

223 }

224 }

225 else if (ch == 3)
226 {
227     int j = 0, a = 0, sum = 0, g, flag, num;
228     read_staff();
229     int c, i, q;
230     int m, n;
231
232     while (c != 6)
233     {
234
235         printf("Enter your choice\n\n1. Add Staff Information\n2. View Staff Information\n3. Search Staff\n4. Edit Staff Information\n5. Delete Staff
236 Information\n6. Exit Staff\n\nOption=");
237         scanf("%d", &c);
238         fflush(stdin);
239     }
240 }
```

```
C minor_main.c minor_main.c main()
239     if (c == 1)
240     {
241         add_staff();
242     }
243     else if (c == 2)
244     {
245         view_staff();
246     }
247     else if (c == 3)
248     {
249         search_staff();
250     }
251     else if (c == 4)
252     {
253         edit_staff();
254     }
255     else if (c == 5)
256     {
257         delete_staff();
258     }
259     else if (c == 6)
260     {
261         write_staff();
262     }
263     else
264     {
265         printf("\n\nInvalid input, try again by using valid inputs.");
266     }
267     printf("\n\n");
268 }
269 }
270
271 else if (ch == 4)
272 {
273     int j = 0, a = 0, sum = 0, g, flag, num;
274     read_medicine();
275     j = 1;
276 }
```

```
C minor_main.c minor_main.c main()
276     int c, i, q;
277     int m, n;
278
279     while (c != 6)
280     {
281
282         printf("Enter your choice\n\n1. Add Medicine Information\n2. View Medicine Information\n3. Search Medicine Information\n4. Edit Medicine
283         Information\n5. Delete Medicine Information\n6. Exit Medicine\n\nOption=");
284         scanf("%d", &c);
285         fflush(stdin);
286         if (c == 1)
287         {
288             add_medicine();
289         }
290         else if (c == 2)
291         {
292             view_medicine();
293         }
294         else if (c == 3)
295         {
296             search_medicine();
297         }
298         else if (c == 4)
299         {
300             edit_medicine();
301         }
302         else if (c == 5)
303         {
304             delete_medicine();
305         }
306         else if (c == 6)
307         {
308             medicine_write();
309         }
310         else
311         {
312             printf("\n\nInvalid input, try again by using valid inputs.");
313         }
314     }
315 }
```

```
C minor_main.c minor_main.c...
312     |         printf("\n\nInvalid input, try again by using valid inputs.");
313     |     }
314     | }
315 }
316 }
317 }
318 else if (ch == 5)
319 {
320     int opt, n;
321     struct orders ord;
322     struct orders order;
323     char saveBill = 'y', contFlag = 'y';
324     char name[50];
325     FILE *fp;
326
327     while (contFlag == 'y')
328     {
329
330         float total = 0;
331         int invoiceFound = 0;
332         printf("\t===== Patient/Customer's Invoice =====");
333         printf("\n\nPlease select your preferred operation");
334         printf("\n1.Generate Invoice");
335         printf("\n2.Show all Invoices");
336         printf("\n3.Search Invoice");
337         printf("\n4.Exit");
338
339         printf("\n\nYour choice:\t");
340         scanf("%d", &opt);
341         fgetc(stdin);
342         switch (opt)
343         {
344             case 1:
345
346                 printf("\nPlease enter the name of patient/customer:\t");
347                 fgets(ord.customer, 50, stdin);
348                 ord.customer[strlen(ord.customer) - 1] = 0;
349                 strcpy(ord.date, __DATE__);
350                 printf("\nPlease enter the number of fields :\t");
351                 scanf("%d", &n);
352                 ord.numOfItems = n;
353                 for (int i = 0; i < n; i++)
354                 {
355                     fgetc(stdin);
356                     printf("\n");
357                     printf("Please enter the item %d:\t", i + 1);
358                     fgets(ord.item[i].item, 20, stdin);
359                     ord.item[i].item[strlen(ord.item[i].item) - 1] = 0;
360                     printf("Please enter the quantity:\t");
361                     scanf("%d", &ord.item[i].qty);
362                     printf("Please enter the unit price:\t");
363                     scanf("%f", &ord.item[i].price);
364                     total += ord.item[i].qty * ord.item[i].price;
365
366                 generateBillHeader(ord.customer, ord.date);
367                 for (int i = 0; i < ord.numOfItems; i++)
368                 {
369                     generateBillBody(ord.item[i].item, ord.item[i].qty, ord.item[i].price);
370                 }
371                 generateBillFooter(total);
372
373                 printf("\nDo you want to save the invoice [y/n]:\t");
374                 scanf("%s", &saveBill);
375
376                 if (saveBill == 'y')
377                 {
378                     fp = fopen("bill.txt", "a+");
379                     fwrite(&ord, sizeof(struct orders), 1, fp);
380                     if (fwrite != 0)
381                         printf("\nSuccessfully saved");
382                     else
383                         printf("\nError saving");
384                     fclose(fp);
385                 }
386             }
387         }
388     }
389 }
```

```
C minor_main.c minor_main.c...
348 ord.customer[strlen(ord.customer) - 1] = 0;
349 strcpy(ord.date, __DATE__);
350 printf("\nPlease enter the number of fields :\t");
351 scanf("%d", &n);
352 ord.numOfItems = n;
353 for (int i = 0; i < n; i++)
354 {
355     fgetc(stdin);
356     printf("\n");
357     printf("Please enter the item %d:\t", i + 1);
358     fgets(ord.item[i].item, 20, stdin);
359     ord.item[i].item[strlen(ord.item[i].item) - 1] = 0;
360     printf("Please enter the quantity:\t");
361     scanf("%d", &ord.item[i].qty);
362     printf("Please enter the unit price:\t");
363     scanf("%f", &ord.item[i].price);
364     total += ord.item[i].qty * ord.item[i].price;
365
366     generateBillHeader(ord.customer, ord.date);
367     for (int i = 0; i < ord.numOfItems; i++)
368     {
369         generateBillBody(ord.item[i].item, ord.item[i].qty, ord.item[i].price);
370     }
371     generateBillFooter(total);
372
373     printf("\nDo you want to save the invoice [y/n]:\t");
374     scanf("%s", &saveBill);
375
376     if (saveBill == 'y')
377     {
378         fp = fopen("bill.txt", "a+");
379         fwrite(&ord, sizeof(struct orders), 1, fp);
380         if (fwrite != 0)
381             printf("\nSuccessfully saved");
382         else
383             printf("\nError saving");
384         fclose(fp);
385     }
386 }
```

```
C minor_main.c minor_main.c\...
```

```
385     fclose(fp);
386 }
387 break;
388
389 case 2:
390
391     fp = fopen("bill.txt", "r");
392     printf("\n ****Your Previous Invoices****\n");
393     while (fread(&order, sizeof(struct orders), 1, fp))
394     {
395         float tot = 0;
396         generateBillHeader(order.customer, order.date);
397         for (int i = 0; i < order.numOfItems; i++)
398         {
399             generateBillBody(order.item[i].item, order.item[i].qty, order.item[i].price);
400             tot += order.item[i].qty * order.item[i].price;
401         }
402         generateBillFooter(tot);
403     }
404     fclose(fp);
405 break;
406
407 case 3:
408     printf("Enter the name of the patient/customer:\t");
409     fgets(name, 50, stdin);
410     name[strlen(name) - 1] = 0;
411
412     fp = fopen("bill.txt", "r");
413     printf("\n*****Invoice of %*****", name);
414     while (fread(&order, sizeof(struct orders), 1, fp))
415     {
416         float tot = 0;
417         if (!strcmp(order.customer, name))
418         {
419             generateBillHeader(order.customer, order.date);
420             for (int i = 0; i < order.numOfItems; i++)
421             {
422                 generateBillBody(order.item[i].item, order.item[i].qty, order.item[i].price);
423             }
424         }
425     }
426 }
```

```
C minor_main.c minor_main.c(main)
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457

    {
        generateBillBody(order.item[i].item, order.item[i].qty, order.item[i].price);
        tot += order.item[i].qty * order.item[i].price;
    }
    generateBillFooter(tot);
    invoiceFound = 1;
}
if (!invoiceFound)
{
    printf("Sorry the invoice for %s doesnot exists", name);
}
fclose(fp);
break;

case 4:
    printf("\n\t\t Bye Bye :)\n\n");
    break;

default:
    printf("Sorry invalid option");
    break;
}
printf("\nDo you want to perform another operation in billing?[y/n]:\t");
scanf("%s", &contFlag);
}

}

else if (ch == 6)
{
    exit(0);
}
```

```
C minor_main.c minor_main.c (main)
457     }
458
459     else
460         printf("\n\nInvalid Serial number\n");
461
462     printf("\nIf you want to continue module press 1 \n");
463     scanf("%d", &module);
464 } while (module == 1);
465
466
467 void patient_add()
468 {
469     printf("\n\n");
470     printf("Already data inputed on the database =%d\n\n", num);
471     printf("How many entry do you want to add=\n");
472     scanf("%d", &n);
473     sum = n + num;
474
475     for (i = num, j = 0; i < sum; i++)
476     {
477         printf("\n");
478         fflush(stdin);
479         printf("Enter patient's name = ");
480         gets(v[i].patient_name);
481         fflush(stdin);
482         printf("Enter disease = ");
483         gets(v[i].disease);
484         fflush(stdin);
485         printf("Enter age = ");
486         scanf("%d", &v[i].age);
487         fflush(stdin);
488         printf("Enter room number = ");
489         scanf("%d", &v[i].room_no);
490         fflush(stdin);
491         printf("Enter phone number = ");
492         gets(v[i].phone);
493         fflush(stdin);

```

```
C minor_main.c minor_main.c (main)
493     fflush(stdin);
494     printf("\n");
495     j++;
496     a++;
497     num++;
498 }
499
500 void patient_view()
501 {
502     for (i = 0; i < num; i++)
503     {
504         printf("\n");
505         printf("Serial number = %d\n", i);
506         printf("Patient_name = ");
507         puts(v[i].patient_name);
508         printf("Disease = ");
509         puts(v[i].disease);
510         printf("Phone number = ");
511         puts(v[i].phone);
512         printf("Room number = %d\nAge=%d", v[i].room_no, v[i].age);
513         printf("\n\n");
514     }
515 }
516 void patient_edit()
517 {
518     int q, p;
519     fflush(stdin);
520     printf("What do you want to edit ?\n");
521     printf("Enter your option\n");
522     printf("1.Patient name\n2.Disease\n3.Age\n4.Room number\n5.Phone no.\n");
523     printf("Option=");
524     scanf("%d", &q);
525     if (q <= 5)
526     {
527         printf("Enter the serial no of that patient = (0 - %d)=", num - 1);
528         scanf("%d", &p);
529         if (p < num)
530             {

```

```
C minor_main.c minor_main.c...
530     {
531         if (q == 1)
532         {
533             fflush(stdin);
534             printf("Enter the new patient name = ");
535             gets(v[p].patient_name);
536         }
537         else if (q == 2)
538         {
539             fflush(stdin);
540             printf("Enter the new disease = ");
541             gets(v[p].disease);
542         }
543         else if (q == 3)
544         {
545             fflush(stdin);
546             printf("Enter the new age = ");
547             scanf("%d", &v[p].age);
548         }
549
550         else if (q == 4)
551         {
552             fflush(stdin);
553             printf("Enter the new room number = ");
554             scanf("%d", &v[p].room_no);
555         }
556
557         else if (q == 5)
558         {
559             fflush(stdin);
560             printf("Enter the new phone no = ");
561             gets(v[p].phone);
562         }
563     }
564     else
565     {
566         printf("\n\nInvalid Serial \nTry Again !!\n\n");
567     }
568 }
```

```
C minor_main.c minor_main.c...
569     printf("\n\nInvalid Serial \nTry Again !!\n\n");
570 }
571 else
572 {
573     printf("\n\nInvalid option\nTry Again!!\n\n");
574 }
575 void patient_search()
576 {
577     int s, h, f;
578     char u[100];
579     printf("By what do you want to patient_search?\n");
580     printf("1.Serial no.\n2.Patient name\n3.Disease\n4.Room number.\n5.Phone no.\n6.Age\n\nOption = ");
581     scanf("%d", &h);
582     if (h == 1)
583     {
584         printf("Enter Serial number of the patient=");
585         scanf("%d", &s);
586         if (s < num)
587         {
588             printf("\n");
589             printf("Serial Number = %d\n", s);
590             printf("Patient name = ");
591             puts(v[s].patient_name);
592             printf("Disease = ");
593             puts(v[s].disease);
594             printf("Phone number = ");
595             puts(v[s].phone);
596             printf("Room number = %d\nAge = %d", v[s].room_no, v[s].age);
597             printf("\n\n");
598         }
599         else
600         {
601             printf("\n\nNot Found\n\n");
602         }
603     }
604     else if (h == 2)
605     {
606         int f = 1;
```

```
C minor_main.c minor_main.c...
603     int r = 1;
604     fflush(stdin);
605     printf("Enter Patient's Name=");
606     gets(u);
607     fflush(stdin);
608     for (g = 0; g < num; g++)
609     {
610         if (strcmp(u, v[g].patient_name) == 0)
611         {
612             printf("\n");
613             printf("Serial number = %d\n", g);
614             printf("Patient name = ");
615             puts(v[g].patient_name);
616             printf("Disease = ");
617             puts(v[g].disease);
618             printf("Phone number = ");
619             puts(v[g].phone);
620             printf("Room number = %d\nAge = %d", v[g].room_no, v[g].age);
621             printf("\n\n");
622             f = 0;
623         }
624     }
625     if (f == 1)
626     {
627         printf("\nNot Found\n");
628     }
629     else if (h == 3)
630     {
631         int f = 1;
632         fflush(stdin);
633         printf("Enter Disease = ");
634         gets(u);
635         fflush(stdin);
636         for (g = 0; g < num; g++)
637         {
638             if (strcmp(u, v[g].disease) == 0)
639             {
640                 printf("\n");
641                 printf("Serial Number = %d\n", g);
```

```
C minor_main.c minor_main.c...
641                 printf("Patient name = ");
642                 puts(v[g].patient_name);
643                 printf("Disease = ");
644                 puts(v[g].disease);
645                 printf("Phone number = ");
646                 puts(v[g].phone);
647                 printf("Room_no = %d\nAge = %d", v[g].room_no, v[g].age);
648                 printf("\n\n");
649                 f = 0;
650             }
651         }
652         if (f == 1)
653         {
654             printf("\nNot Found\n");
655         }
656     else if (h == 4)
657     {
658         int f = 1;
659         printf("Enter room_no = ");
660         scanf("%d", &f);
661         for (g = 0; g < num; g++)
662         {
663             if (f == v[g].room_no)
664             {
665                 printf("\n");
666                 printf("Serial Number = %d\n", g);
667                 printf("Patient Name = ");
668                 puts(v[g].patient_name);
669                 printf("Disease = ");
670                 puts(v[g].disease);
671                 printf("Phone number = ");
672                 puts(v[g].phone);
673                 printf("Room number = %d\nAge = %d", v[g].room_no, v[g].age);
674                 printf("\n\n");
675             }
676         }
677         if (f == 1)
678             printf("Not Found\n");
```

```
C minor_main.c minor_main.c...
678     |     printf("Not Found\n\n");
679 }
680 else if (h == 5)
{
681     int f = 1;
682     fflush(stdin);
683     printf("Enter Phone number = ");
684     gets(u);
685     fflush(stdin);
686     for (g = 0; g < num; g++)
    {
688         if (strcmp(u, v[g].phone) == 0)
    {
689             printf("\n");
690             printf("Serial Number = %d\n", g);
691             printf("Patient name = ");
692             puts(v[g].patient_name);
693             printf("Disease = ");
694             puts(v[g].disease);
695             printf("Phone number = ");
696             puts(v[g].phone);
697             printf("Room number = %d\nAge = %d", v[g].room_no, v[g].age);
698             printf("\n\n");
699             f = 0;
700         }
701     }
702 }
703 if (f == 1)
704     printf("Not Found");
705 }
706 else if (h == 6)
{
707     int f = 1;
708     printf("Enter Age = ");
709     scanf("%d", &f);
710     for (g = 0; g < num; g++)
    {
711         if (f == v[g].age)
712             f = 0;
713     }
714 }
```

```
C minor_main.c minor_main.c...
715     {
716         printf("\n");
717         printf("Serial number = %d\n", g);
718         printf("Patient name = ");
719         puts(v[g].patient_name);
720         printf("Disease = ");
721         puts(v[g].disease);
722         printf("Phone number = ");
723         puts(v[g].phone);
724         printf("Room number = %d\nAge = %d", v[g].room_no, v[g].age);
725         printf("\n\n");
726     }
727 }
728 if (f == 1)
729     printf("Not Found\n\n");
730 }
731 else
732     printf("\n\nInvalid input\n\n");
733 }
734 void delete_patient()
735 {
736     int f, h;
737     printf("Enter the serial number of the patient that you want to delete = ");
738     scanf("%d", &f);
739     if (f < num)
    {
740         printf("What do you want ?\n");
741         printf("1.Remove the whole record\n2.Remove Patient name\n3.Remove Disease\n4.Remove Age\n5.Remove Room number\n6.Remove Phone number\nOption = ");
742         scanf("%d", &h);
743         if (h == 1)
744         {
745             while (f < num)
    {
746                 strcpy(v[f].patient_name, v[f + 1].patient_name);
747                 strcpy(v[f].disease, v[f + 1].disease);
748                 strcpy(v[f].phone, v[f + 1].phone);
749                 v[f].age = v[f + 1].age;
750             }
751         }
752     }
753 }
```

```
C minor_main.c minor_main.c\\..  
751     v[1].age = v[0].age;  
752     v[f].room_no = v[f + 1].room_no;  
753     f++;  
754 }  
755 num--;  
756 }  
757 else if (h == 2)  
758 {  
    strcpy(v[f].patient_name, "Cleared");  
759 }  
else if (h == 3)  
760 {  
    strcpy(v[f].disease, "Cleared");  
761 }  
else if (h == 4)  
762 {  
    v[f].age = 0;  
763 }  
else if (h == 5)  
764 {  
    v[f].room_no = 0;  
765 }  
else if (h == 6)  
766 {  
    strcpy(v[f].phone, "cleared");  
767 }  
768 }  
769 else  
770 {  
    printf("\n\nInvalid Serial Number\n");  
771 }  
772 void patient_read()  
773 {  
    FILE *fp = fopen("patient.txt", "r");  
774     if (fp == NULL)  
775     {  
776         fp = fopen("patient.txt", "w");  
777         fclose(fp);  
778     }  
779 }
```

```
C minor_main.c minor_main.c\\..  
788     fclose(fp);  
789     printf("File does not exist, I JUST CREATED IT, exiting...\n\n");  
790     return 0;  
791 }  
792  
793 num = fread(v, sizeof(struct p), 100, fp);  
794     fclose(fp);  
795 }  
796 void patient_write()  
797 {  
    FILE *fp = fopen("patient.txt", "w");  
798     if (fp == NULL)  
799     {  
        printf("Error");  
        exit(1);  
800     }  
801     fwrite(v, sizeof(struct p), num, fp);  
802     fclose(fp);  
803 }  
804  
805 void add_doctor()  
806 {  
807     printf("\n\n");  
808     printf("Already data inputed on the database = %d\n\n", num);  
809     printf("How many entry do you want to add = \n");  
810     scanf("%d", &n);  
811     sum = n + num;  
812  
813     for (i = num, j = 0; i < sum; i++)  
814     {  
815         printf("\n");  
816         fflush(stdin);  
817         printf("Enter doctor's name = ");  
818         gets(w[i].doctor_name);  
819         fflush(stdin);  
820         printf("Enter address = ");  
821         gets(w[i].address);  
822     }  
823 }
```

```
C minor_main.c minor_main.c...
825     gets(w[i].address);
826     fflush(stdin);
827     printf("Enter the age = ");
828     scanf("%d", &w[i].age);
829     fflush(stdin);
830     printf("Enter doctor id = ");
831     scanf("%d", &w[i].doctor_id);
832     fflush(stdin);
833     printf("Enter phone number = ");
834     gets(w[i].phone);
835     fflush(stdin);
836     printf("\n");
837     j++;
838     a++;
839     num++;
840 }
841 }
842 void view_doctor()
843 {
844     for (i = 0; i < num; i++)
845     {
846         printf("\n");
847         printf("Serial number = %d\n", i);
848         printf("Doctor's name = ");
849         puts(w[i].doctor_name);
850         printf("Address = ");
851         puts(w[i].address);
852         printf("Phone number = ");
853         puts(w[i].phone);
854         printf("Doctor's id = %d\nAge = %d", w[i].doctor_id, w[i].age);
855         printf("\n\n");
856     }
857 }
858 void edit_doctor()
859 {
860     int q, p;
861     fflush(stdin);
862     printf("What do you want to edit ?\n");
```

```
C minor_main.c minor_main.c...
862     printf("What do you want to edit ?\n");
863     printf("Enter your option\n");
864     printf("1.Doctor's name\n2.Address\n3.Age\n4.Doctor id\n5.Phone number\n");
865     printf("Option=");
866     scanf("%d", &q);
867     if (q <= 5)
868     {
869         printf("Enter the serial no of that doctor = (0 - %d)=", num - 1);
870         scanf("%d", &p);
871         if (p < num)
872         {
873             if (q == 1)
874             {
875                 fflush(stdin);
876                 printf("Enter the new Doctor name = ");
877                 gets(w[p].doctor_name);
878             }
879             else if (q == 2)
880             {
881                 fflush(stdin);
882                 printf("Enter the new Address = ");
883                 gets(w[p].address);
884             }
885             else if (q == 3)
886             {
887                 fflush(stdin);
888                 printf("Enter the new Age = ");
889                 scanf("%d", &w[p].age);
890             }
891             else if (q == 4)
892             {
893                 fflush(stdin);
894                 printf("Enter the new Doctor id = ");
895                 scanf("%d", &w[p].doctor_id);
896             }
897         }
898         else if (q == 5)
```

```

C minor_main.c minor_main.c...
899     else if (q == 5)
900     {
901         fflush(stdin);
902         printf("Enter the new Phone number =");
903         gets(w[p].phone);
904     }
905     }
906     else
907     {
908         printf("\n\nInvalid Serial \nTry Again !!\n\n");
909     }
910 }
911 else
912 {
913     printf("\n\nInvalid Option\nTry Again!!\n\n");
914 }
915 }
916 void search_doctor()
917 {
918     int s, h, f;
919     char u[100];
920     printf("By what do you want to search_doctor ?\n");
921     printf("1.Serial number\n2.Doctor name\n3.Address\n4.Doctor id.\n5.Phone number\n6.Age\n\nOption = ");
922     scanf("%d", &h);
923     if (h == 1)
924     {
925         printf("Enter serial number of the doctor = ");
926         scanf("%d", &s);
927         if (s < num)
928         {
929             printf("\n");
930             printf("Serial number = %d\n", s);
931             printf("Doctor name = ");
932             puts(w[s].doctor_name);
933             printf("Address = ");
934             puts(w[s].address);
935             printf("Phone number = ");
936             puts(w[s].phone);
937             printf("Doctor id = %d\nAge = %d", w[s].doctor_id, w[s].age);
938             printf("\n\n");
939         }
940         else
941             printf("\n\nNot Found\n\n");
942     }
943     else if (h == 2)
944     {
945         int f = 1;
946         fflush(stdin);
947         printf("Enter Doctor's name = ");
948         gets(u);
949         fflush(stdin);
950         for (g = 0; g < num; g++)
951         {
952             if (strcmp(u, w[g].doctor_name) == 0)
953             {
954                 printf("\n");
955                 printf("Serial number=%d\n", g);
956                 printf("Doctor name = ");
957                 puts(w[g].doctor_name);
958                 printf("Address = ");
959                 puts(w[g].address);
960                 printf("Phone number = ");
961                 puts(w[g].phone);
962                 printf("Doctor id = %d\nAge = %d", w[g].doctor_id, w[g].age);
963                 printf("\n\n");
964                 f = 0;
965             }
966         }
967         if (f == 1)
968             printf("\nNot Found\n");
969     }
970     else if (h == 3)
971     {
972         int f = 1;

```

```

C minor_main.c minor_main.c...
936         puts(w[s].phone);
937         printf("Doctor id = %d\nAge = %d", w[s].doctor_id, w[s].age);
938         printf("\n\n");
939     }
940     else
941         printf("\n\nNot Found\n\n");
942 }
943 else if (h == 2)
944 {
945     int f = 1;
946     fflush(stdin);
947     printf("Enter Doctor's name = ");
948     gets(u);
949     fflush(stdin);
950     for (g = 0; g < num; g++)
951     {
952         if (strcmp(u, w[g].doctor_name) == 0)
953         {
954             printf("\n");
955             printf("Serial number=%d\n", g);
956             printf("Doctor name = ");
957             puts(w[g].doctor_name);
958             printf("Address = ");
959             puts(w[g].address);
960             printf("Phone number = ");
961             puts(w[g].phone);
962             printf("Doctor id = %d\nAge = %d", w[g].doctor_id, w[g].age);
963             printf("\n\n");
964             f = 0;
965         }
966     }
967     if (f == 1)
968         printf("\nNot Found\n");
969 }
970 else if (h == 3)
971 {
972     int f = 1;

```

```
C minor_main.c minor_main.c...
274     int c = -1;
973     fflush(stdin);
974     printf("Enter Address = ");
975     gets(u);
976     fflush(stdin);
977     for (g = 0; g < num; g++)
978     {
979         if (strcmp(u, w[g].address) == 0)
980         {
981             printf("\n");
982             printf("Serial number = %d\n", g);
983             printf("Doctor name = ");
984             puts(w[g].doctor_name);
985             printf("Address = ");
986             puts(w[g].address);
987             printf("Phone number = ");
988             puts(w[g].phone);
989             printf("Doctor id = %d\nAge = %d", w[g].doctor_id, w[g].age);
990             printf("\n\n");
991             f = 0;
992         }
993     }
994     if (f == 1)
995         printf("\nNot Found\n");
996 }
997 else if (h == 4)
998 {
999     int f = 1;
1000    printf("Enter Doctor's id = ");
1001    scanf("%d", &f);
1002    for (g = 0; g < num; g++)
1003    {
1004        if (f == w[g].doctor_id)
1005        {
1006            printf("\n");
1007            printf("Serial number = %d\n", g);
1008            printf("Doctor name = ");
1009            puts(w[g].doctor_name);
```

```
C minor_main.c minor_main.c...
1009     puts(w[g].doctor_name);
1010     printf("Address = ");
1011     puts(w[g].address);
1012     printf("Phone number = ");
1013     puts(w[g].phone);
1014     printf("Doctor id = %d\nAge = %d", w[g].doctor_id, w[g].age);
1015     printf("\n\n");
1016     f = 0;
1017 }
1018 }
1019 if (f == 1)
1020     printf("Not Found\n\n");
1021 }
1022 else if (h == 5)
1023 {
1024     int f = 1;
1025     fflush(stdin);
1026     printf("Enter phone number = ");
1027     gets(u);
1028     fflush(stdin);
1029     for (g = 0; g < num; g++)
1030     {
1031         if (strcmp(u, w[g].phone) == 0)
1032         {
1033             printf("\n");
1034             printf("Serial number = %d\n", g);
1035             printf("Doctor name = ");
1036             puts(w[g].doctor_name);
1037             printf("Address = ");
1038             puts(w[g].address);
1039             printf("Phone number = ");
1040             puts(w[g].phone);
1041             printf("Doctor id = %d\nAge = %d", w[g].doctor_id, w[g].age);
1042             printf("\n\n");
1043             f = 0;
1044         }
1045     }
1046     if (f == 1)
```

```
C minor_main.c minor_main.c...
1046     if (f == 1)
1047         printf("Not Found");
1048     }
1049     else if (h == 6)
1050     {
1051         int f = 1;
1052         printf("Enter Age = ");
1053         scanf("%d", &f);
1054         for (g = 0; g < num; g++)
1055         {
1056             if (f == w[g].age)
1057             {
1058                 printf("\n");
1059                 printf("Serial number = %d\n", g);
1060                 printf("Doctor name = ");
1061                 puts(w[g].doctor_name);
1062                 printf("Address = ");
1063                 puts(w[g].address);
1064                 printf("Phone number = ");
1065                 puts(w[g].phone);
1066                 printf("Doctor id = %d\nAge = %d", w[g].doctor_id, w[g].age);
1067                 printf("\n\n");
1068             }
1069         }
1070         if (f == 1)
1071             printf("Not Found\n\n");
1072     }
1073     else
1074         printf("\n\nInvalid input\n\n");
1075     }
1076 void delete_doctor()
1077 {
1078     int f, h;
1079     printf("Enter the serial number of the doctor that you want to delete = ");
1080     scanf("%d", &f);
1081     if (f < num)
1082     {
1083         printf("What do you want ?\n");
```

```
C minor_main.c minor_main.c...
1083     printf("What do you want ?\n");
1084     printf("1.Remove the whole record\n2.Remove Doctor's Name\n3.Remove Address\n4.Remove Age\n5.Remove Doctor Id.\n6.Remove Phone Number\nOption = ");
1085     scanf("%d", &h);
1086     if (h == 1)
1087     {
1088         while (f < num)
1089         {
1090             strcpy(w[f].doctor_name, w[f + 1].doctor_name);
1091             strcpy(w[f].address, w[f + 1].address);
1092             strcpy(w[f].phone, w[f + 1].phone);
1093             w[f].age = w[f + 1].age;
1094             w[f].doctor_id = w[f + 1].doctor_id;
1095             f++;
1096         }
1097         num--;
1098     }
1099     else if (h == 2)
1100     {
1101         strcpy(w[f].doctor_name, "Cleared");
1102     }
1103     else if (h == 3)
1104     {
1105         strcpy(w[f].address, "Cleared");
1106     }
1107     else if (h == 4)
1108     {
1109         w[f].age = 0;
1110     }
1111     else if (h == 5)
1112     {
1113         w[f].doctor_id = 0;
1114     }
1115     else if (h == 6)
1116     {
1117         strcpy(w[f].phone, "Cleared");
1118     }
1119 }
1120 else
```

```
C minor_main.c minor_main.c\\..  
1120     else  
1121         |     printf("\\n\\nInvalid Serial Number\\n");  
1122     }  
1123 void read_doctor()  
1124 {  
1125     FILE *fp = fopen("doctor.txt", "r");  
1126     if (fp == NULL)  
1127     {  
1128         fp = fopen("doctor.txt", "w");  
1129         fclose(fp);  
1130         printf("File does not exist, I JUST CREATED IT, exiting...\\n\\n\\n");  
1131         return 0;  
1132     }  
1133  
1134     num = fread(w, sizeof(struct d), 100, fp);  
1135     fclose(fp);  
1136 }  
1137 void write_doctor()  
1138 {  
1139     FILE *fp = fopen("doctor.txt", "w");  
1140     if (fp == NULL)  
1141     {  
1142         printf("Error");  
1143         exit(1);  
1144     }  
1145     fwrite(w, sizeof(struct d), num, fp);  
1146  
1147     fclose(fp);  
1148 }  
1149  
1150 void add_staff()  
1151 {  
1152     printf("\\n\\n");  
1153     printf("Already data inputed on the database = %d\\n\\n", num);  
1154     printf("How many entry do you want to add = \\n");  
1155     scanf("%d", &n);  
1156     sum = n + num;  
1157 }
```

```
C minor_main.c minor_main.c\\..  
1157  
1158     for (i = num, j = 0; i < sum; i++)  
1159     {  
1160         printf("\\n");  
1161         fflush(stdin);  
1162         printf("Enter Staff's name = ");  
1163         gets(x[i].staff_name);  
1164         fflush(stdin);  
1165         printf("Enter Job = ");  
1166         gets(x[i].job);  
1167         fflush(stdin);  
1168         printf("Enter the Age = ");  
1169         scanf("%d", &x[i].age);  
1170         fflush(stdin);  
1171         printf("Enter Job id = ");  
1172         scanf("%d", &x[i].job_id);  
1173         fflush(stdin);  
1174         printf("Enter Phone Number = ");  
1175         gets(x[i].phone);  
1176         fflush(stdin);  
1177         printf("\\n");  
1178         j++;  
1179         a++;  
1180         num++;  
1181     }  
1182 }  
1183 void view_staff()  
1184 {  
1185     for (i = 0; i < num; i++)  
1186     {  
1187         printf("\\n");  
1188         printf("Serial Number = %d\\n", i);  
1189         printf("Staff Name = ");  
1190         puts(x[i].staff_name);  
1191         printf("Job = ");  
1192         puts(x[i].job);  
1193         printf("Phone Number = ");  
1194         puts(x[i].phone);  
1195     }
```

```
C minor_main.c minor_main.c...
1194     puts(x[i].phone);
1195     printf("Job id = %d\nAge=%d", x[i].job_id, x[i].age);
1196     printf("\n\n");
1197 }
1198 }
1199 void edit_staff()
1200 {
1201     int q, p;
1202     fflush(stdin);
1203     printf("What do you want to edit ?\n");
1204     printf("Enter your option\n");
1205     printf("1.Staff name\n2.Job\n3.Age\n4.Job id\n5.Phone number\n");
1206     printf("Option = ");
1207     scanf("%d", &q);
1208     if (q <= 5)
1209     {
1210         printf("Enter the serial no of that staff = (0 - %d)=", num - 1);
1211         scanf("%d", &p);
1212         if (p < num)
1213         {
1214             if (q == 1)
1215             {
1216                 fflush(stdin);
1217                 printf("Enter the new Staff Name = ");
1218                 gets(x[p].staff_name);
1219             }
1220             else if (q == 2)
1221             {
1222                 fflush(stdin);
1223                 printf("Enter the new Job = ");
1224                 gets(x[p].job);
1225             }
1226             else if (q == 3)
1227             {
1228                 fflush(stdin);
1229                 printf("Enter the new Age = ");
1230                 scanf("%d", &x[p].age);
1231             }
1232         }
1233     }
1234 }
```

```
C minor_main.c minor_main.c...
1230     scanf(" %d", &x[p].age);
1231 }
1232
1233 else if (q == 4)
1234 {
1235     fflush(stdin);
1236     printf("Enter the new Job id = ");
1237     scanf("%d", &x[p].job_id);
1238 }
1239
1240 else if (q == 5)
1241 {
1242     fflush(stdin);
1243     printf("Enter the new Phone number = ");
1244     gets(x[p].phone);
1245 }
1246
1247 else
1248 {
1249     printf("\n\nInvalid Serial \nTry Again !!\n\n");
1250 }
1251
1252 else
1253 {
1254     printf("\n\nInvalid option\nTry Again!!\n\n");
1255 }
1256 }
1257 void search_staff()
1258 {
1259     int s, h, f;
1260     char u[100];
1261     printf("By what do you want to search staff ?\n");
1262     printf("1.Serial number \n2.Staff name\n3.Job\n4.Job id.\n5.Phone number\n6.Age\n\nOption = ");
1263     scanf("%d", &h);
1264     if (h == 1)
1265     {
1266         printf(" Enter Serial number of the staff = ");
1267         scanf("%d", &s);
1268     }
1269 }
```

```
C minor_main.c minor_main.c...
1267     scanf("%d", &s);
1268     if (s < num)
1269     {
1270         printf("\n");
1271         printf("Serial Number = %d\n", s);
1272         printf("Staff name = ");
1273         puts(x[s].staff_name);
1274         printf("Job = ");
1275         puts(x[s].job);
1276         printf("Phone number = ");
1277         puts(x[s].phone);
1278         printf("Job id = %d\nAge = %d", x[s].job_id, x[s].age);
1279         printf("\n\n");
1280     }
1281     else
1282     {
1283         printf("\n\nNot Found\n\n");
1284     }
1285     else if (h == 2)
1286     {
1287         int f = 1;
1288         fflush(stdin);
1289         printf("Enter staff name =");
1290         gets(u);
1291         fflush(stdin);
1292         for (g = 0; g < num; g++)
1293         {
1294             if (strcmp(u, x[g].staff_name) == 0)
1295             {
1296                 printf("\n");
1297                 printf("Serial Number = %d\n", g);
1298                 printf("Staff Name = ");
1299                 puts(x[g].staff_name);
1300                 printf("Job = ");
1301                 puts(x[g].job);
1302                 printf("Phone number = ");
1303                 puts(x[g].phone);
1304                 printf("Job id = %d\nAge = %d", x[g].job_id, x[g].age);
1305                 printf("\n\n");
1306             }
1307         }
1308         if (f == 1)
1309             printf("\nNot Found\n");
1310     }
1311     else if (h == 3)
1312     {
1313         int f = 1;
1314         fflush(stdin);
1315         printf("Enter job = ");
1316         gets(u);
1317         fflush(stdin);
1318         for (g = 0; g < num; g++)
1319         {
1320             if (strcmp(u, x[g].job) == 0)
1321             {
1322                 printf("\n");
1323                 printf("Serial Number = %d\n", g);
1324                 printf("Staff name = ");
1325                 puts(x[g].staff_name);
1326                 printf("Job = ");
1327                 puts(x[g].job);
1328                 printf("Phone number = ");
1329                 puts(x[g].phone);
1330                 printf("Job id = %d\nAge = %d", x[g].job_id, x[g].age);
1331                 printf("\n\n");
1332             }
1333         }
1334         if (f == 1)
1335             printf("\nNot Found\n");
1336     }
1337     else if (h == 4)
1338     {
1339         int f = 1;
1340         printf("Enter Job id = ");
```

```
C minor_main.c minor_main.c...
1304         printf("\n\n");
1305         f = 0;
1306     }
1307 }
1308 if (f == 1)
1309     printf("\nNot Found\n");
1310 }
1311 else if (h == 3)
1312 {
1313     int f = 1;
1314     fflush(stdin);
1315     printf("Enter job = ");
1316     gets(u);
1317     fflush(stdin);
1318     for (g = 0; g < num; g++)
1319     {
1320         if (strcmp(u, x[g].job) == 0)
1321         {
1322             printf("\n");
1323             printf("Serial Number = %d\n", g);
1324             printf("Staff name = ");
1325             puts(x[g].staff_name);
1326             printf("Job = ");
1327             puts(x[g].job);
1328             printf("Phone number = ");
1329             puts(x[g].phone);
1330             printf("Job id = %d\nAge = %d", x[g].job_id, x[g].age);
1331             printf("\n\n");
1332         }
1333     }
1334     if (f == 1)
1335         printf("\nNot Found\n");
1336 }
1337 else if (h == 4)
1338 {
1339     int f = 1;
1340     printf("Enter Job id = ");
```

```
C minor_main.c minor_main.c...
1341     printf("Enter Job id = ");
1342     scanf("%d", &f);
1343     for (g = 0; g < num; g++)
1344     {
1345         if (f == x[g].job_id)
1346         {
1347             printf("\n");
1348             printf("Serial Number = %d\n", g);
1349             printf("Staff name = ");
1350             puts(x[g].staff_name);
1351             printf("Job = ");
1352             puts(x[g].job);
1353             printf("Phone number = ");
1354             puts(x[g].phone);
1355             printf("Job id = %d\nAge = %d", x[g].job_id, x[g].age);
1356             printf("\n\n");
1357             f = 0;
1358         }
1359     }
1360     if (f == 1)
1361         printf("Not Found\n\n");
1362 }
1363 else if (h == 5)
1364 {
1365     int f = 1;
1366     fflush(stdin);
1367     printf("Enter Phone number = ");
1368     gets(u);
1369     fflush(stdin);
1370     for (g = 0; g < num; g++)
1371     {
1372         if (strcmp(u, x[g].phone) == 0)
1373         {
1374             printf("\n");
1375             printf("Serial Number = %d\n", g);
1376             printf("Staff name = ");
1377             puts(x[g].staff_name);
1378             printf("Job = ");
1379         }
1380     }
1381     if (f == 1)
1382         printf("Not Found");
1383 }
1384 else if (h == 5)
1385 {
1386     printf("Job = ");
1387     puts(x[g].job);
1388     printf("Phone number = ");
1389     puts(x[g].phone);
1390     printf("Job id = %d\nAge = %d", x[g].job_id, x[g].age);
1391     printf("\n\n");
1392     f = 0;
1393 }
1394 if (f == 1)
1395     printf("Not Found");
1396 }
```

```
C minor_main.c minor_main.c...
1378     printf("Job = ");
1379     puts(x[g].job);
1380     printf("Phone number = ");
1381     puts(x[g].phone);
1382     printf("Job id = %d\nAge = %d", x[g].job_id, x[g].age);
1383     printf("\n\n");
1384     f = 0;
1385 }
1386 }
1387 if (f == 1)
1388     printf("Not Found");
1389 }
1390 else if (h == 6)
1391 {
1392     int f = 1;
1393     printf("Enter Age = ");
1394     scanf("%d", &f);
1395     for (g = 0; g < num; g++)
1396     {
1397         if (f == x[g].age)
1398         {
1399             printf("\n");
1400             printf("Serial Number = %d\n", g);
1401             printf("Staff name = ");
1402             puts(x[g].staff_name);
1403             printf("Job = ");
1404             puts(x[g].job);
1405             printf("Phone number = ");
1406             puts(x[g].phone);
1407             printf("Job id = %d\nAge = %d", x[g].job_id, x[g].age);
1408             printf("\n\n");
1409         }
1410     }
1411     if (f == 1)
1412         printf("Not Found\n\n");
1413 }
1414 else
```

```
C minor_main.c minor_main.c...
1415     |     printf("\n\nInvalid input\n\n");
1416 }
1417 void delete_staff()
1418 {
1419     int f, h;
1420     printf("Enter the serial number of the staff that you want to delete=");
1421     scanf("%d", &f);
1422     if (f < num)
1423     {
1424         printf("What do you want ?\n");
1425         printf("1.Remove the whole record \n2.Remove Staff Name\n3.Remove Job\n4.Remove Age\n5.Remove Job id\n6.Remove Phone Number\nOption = ");
1426         scanf("%d", &h);
1427         if (h == 1)
1428         {
1429             while (f < num)
1430             {
1431                 strcpy(x[f].staff_name, x[f + 1].staff_name);
1432                 strcpy(x[f].job, x[f + 1].job);
1433                 strcpy(x[f].phone, x[f + 1].phone);
1434                 x[f].age = x[f + 1].age;
1435                 x[f].job_id = x[f + 1].job_id;
1436                 f++;
1437             }
1438             num--;
1439         }
1440         else if (h == 2)
1441         {
1442             strcpy(x[f].staff_name, "Cleared");
1443         }
1444         else if (h == 3)
1445         {
1446             strcpy(x[f].job, "Cleared");
1447         }
1448         else if (h == 4)
1449         {
1450             x[f].age = 0;
1451         }
1452     }
1453     else if (h == 5)
1454     {
1455         x[f].job_id = 0;
1456     }
1457     else if (h == 6)
1458     {
1459         strcpy(x[f].phone, "Cleared");
1460     }
1461     else
1462     {
1463         printf("\n\nInvalid serial number\n");
1464     }
1465 }
```

```
C minor_main.c minor_main.c...
1452     |
1453     {
1454         x[f].job_id = 0;
1455     }
1456     else if (h == 6)
1457     {
1458         strcpy(x[f].phone, "Cleared");
1459     }
1460     else
1461     {
1462         printf("\n\nInvalid serial number\n");
1463     }
1464 }
```

```
C minor_main.c minor_main.c...
1465 void read_staff()
1466 {
1467     FILE *fp = fopen("staff.txt", "r");
1468     if (fp == NULL)
1469     {
1470         fp = fopen("staff.txt", "w");
1471         fclose(fp);
1472         printf("File does not exist, I JUST CREATED IT, exiting...\n\n");
1473         return 0;
1474     }
1475     num = fread(x, sizeof(struct s), 100, fp);
1476     fclose(fp);
1477 }
1478 void write_staff()
1479 {
1480     FILE *fp = fopen("staff.txt", "w");
1481     if (fp == NULL)
1482     {
1483         printf("Error");
1484         exit(1);
1485     }
1486     fwrite(x, sizeof(struct s), num, fp);
1487 }
```

```
C minor_main.c minor_main.c...
1488     fclose(fp);
1489 }
1490
1491
1492 void add_medicine()
1493 {
1494     printf("\n\n");
1495     printf("Already data inputed on the database = %d\n\n", num);
1496     printf("How many entry do you want to add = \n");
1497     scanf("%d", &n);
1498     sum = n + num;
1499
1500     for (i = num, j = 0; i < sum; i++)
1501     {
1502         printf("\n");
1503         fflush(stdin);
1504         printf("Enter Medicine's Name = ");
1505         gets(y[i].medicine_name);
1506         fflush(stdin);
1507         printf("Enter Availability = ");
1508         gets(y[i].availability);
1509         fflush(stdin);
1510         printf("Enter The Quantity = ");
1511         scanf("%d", &y[i].quantity);
1512         fflush(stdin);
1513         printf("Enter Product id = ");
1514         scanf("%d", &y[i].product_id);
1515         fflush(stdin);
1516         printf("Enter Price = ");
1517         scanf("%d", &y[i].price);
1518         fflush(stdin);
1519         printf("\n");
1520         j++;
1521         a++;
1522         num++;
1523     }
1524 }
1525 void view_medicine()
```

```
C minor_main.c minor_main.c...
1525 void view_medicine()
1526 {
1527     for (i = 0; i < num; i++)
1528     {
1529         printf("\n");
1530         printf("Serial number = %d\n", i);
1531         printf("Name = ");
1532         puts(y[i].medicine_name);
1533         printf("Availability = ");
1534         puts(y[i].availability);
1535         printf("Product id = %d\nPrice = %d\nQuantity = %d", y[i].product_id, y[i].price, y[i].quantity);
1536         printf("\n\n");
1537     }
1538 }
1539 void edit_medicine()
1540 {
1541     int q, p;
1542     fflush(stdin);
1543     printf("What do you want to edit ?\n");
1544     printf("Enter your option\n");
1545     printf("1.Name\n2.Availability\n3.Quantity\n4.Product id\n5.Price\n");
1546     printf("Option=");
1547     scanf("%d", &q);
1548     if (q <= 5)
1549     {
1550         printf("Enter the serial no of that medicine = (0 - %d)=", num - 1);
1551         scanf("%d", &p);
1552         if (p < num)
1553         {
1554             if (q == 1)
1555             {
1556                 fflush(stdin);
1557                 printf("Enter the new medicine name = ");
1558                 gets(y[p].medicine_name);
1559             }
1560             else if (q == 2)
1561             {
1562                 fflush(stdin);
```



```
C minor_main.c minor_main.c...
1562     ffflush(stdin);
1563     printf("Enter the new availability = ");
1564     gets(y[p].availability);
1565 }
1566 else if (q == 3)
1567 {
1568     ffflush(stdin);
1569     printf("Enter the new quantity = ");
1570     scanf("%d", &y[p].quantity);
1571 }
1572
1573 else if (q == 4)
1574 {
1575     ffflush(stdin);
1576     printf("Enter the new product id = ");
1577     scanf("%d", &y[p].product_id);
1578 }
1579
1580 else if (q == 5)
1581 {
1582     ffflush(stdin);
1583     printf("Enter the new price = ");
1584     scanf("%d", &y[p].price);
1585 }
1586
1587 else
1588 {
1589     printf("\n\nInvalid Serial \nTry Again !!\n\n");
1590 }
1591
1592 else
1593 {
1594     printf("\n\nInvalid Option\nTry Again!!\n\n");
1595 }
1596 }
1597 void search_medicine()
1598 {
1599     int s, h, f;
```

```
C minor_main.c minor_main.c...
1599     int s, h, f;
1600     char u[100];
1601     printf("By what do you want to search medicine ?\n");
1602     printf("1.Serial no.\n2.Name\n3.Availability\n4.Product id.\n5.Price\n6.Quantity\n\nOption = ");
1603     scanf("%d", &h);
1604     if (h == 1)
1605     {
1606         printf("Enter Serial number of the medicine = ");
1607         scanf("%d", &s);
1608         if (s < num)
1609         {
1610             printf("\n");
1611             printf("Serial number = %d\n", s);
1612             printf("Name = ");
1613             puts(y[s].medicine_name);
1614             printf("Availability = ");
1615             puts(y[s].availability);
1616             printf("Product id = %d\nPrice = %d\nQuantity = %d", y[s].product_id, y[s].price, y[s].quantity);
1617             printf("\n\n");
1618         }
1619         else
1620             printf("\n\nNot Found\n\n");
1621     }
1622     else if (h == 2)
1623     {
1624         int f = 1;
1625         ffflush(stdin);
1626         printf("Enter Medicine's Name = ");
1627         gets(u);
1628         ffflush(stdin);
1629         for (g = 0; g < num; g++)
1630         {
1631             if (strcmp(u, y[g].medicine_name) == 0)
1632             {
1633                 printf("\n");
1634                 printf("Serial number = %d\n", g);
1635                 printf("Name = ");
1636                 puts(y[g].medicine_name);
```

```
C minor_main.c minor_main.c...
1636     puts(y[g].medicine_name);
1637     printf("Availability = ");
1638     puts(y[g].availability);
1639     printf("Product_id = %d\nPrice = %d\nQuantity = %d", y[g].product_id, y[g].price, y[g].quantity);
1640     printf("\n\n");
1641     f = 0;
1642 }
1643 if (f == 1)
1644     printf("\nNot Found\n");
1645 }
1646 else if (h == 3)
1647 {
1648     int f = 1;
1649     fflush(stdin);
1650     printf("Enter Availability = ");
1651     gets(u);
1652     fflush(stdin);
1653     for (g = 0; g < num; g++)
1654     {
1655         if (strcmp(u, y[g].availability) == 0)
1656         {
1657             printf("\n");
1658             printf("Serial number = %d\n", g);
1659             printf("Name = ");
1660             puts(y[g].medicine_name);
1661             printf("Availability = ");
1662             puts(y[g].availability);
1663             printf("Product id = %d\nPrice = %d\nQuantity = %d", y[g].product_id, y[g].price, y[g].quantity);
1664             printf("\n\n");
1665             f = 0;
1666         }
1667     }
1668     if (f == 1)
1669         printf("\nNot Found\n");
1670 }
1671 else if (h == 4)
1672 }
```

```
C minor_main.c minor_main.c...
1673 {
1674     int f = 1;
1675     printf("Enter Product id = ");
1676     scanf("%d", &f);
1677     for (g = 0; g < num; g++)
1678     {
1679         if (f == y[g].product_id)
1680         {
1681             printf("\n");
1682             printf("Serial number = %d\n", g);
1683             printf("Name = ");
1684             puts(y[g].medicine_name);
1685             printf("Availability = ");
1686             puts(y[g].availability);
1687             printf("Product_id = %d\nPrice = %d\nQuantity = %d", y[g].product_id, y[g].price, y[g].quantity);
1688             printf("\n\n");
1689             f = 0;
1690         }
1691     }
1692     if (f == 1)
1693         printf("Not Found\n\n");
1694 }
1695 else if (h == 5)
1696 {
1697     int f = 1;
1698     printf("Enter Price = ");
1699     scanf("%d", &f);
1700     for (g = 0; g < num; g++)
1701     {
1702         if (f == y[g].price)
1703         {
1704             printf("\n");
1705             printf("Serial Number = %d\n", g);
1706             printf("Name = ");
1707             puts(y[g].medicine_name);
1708             printf("Availability = ");
1709             puts(y[g].availability);
1710         }
1711     }
1712 }
```

```

C minor_main.c minor_main.c...
1709     puts(y[g].availability);
1710     printf("Product id = %d\nPrice = %d\nQuantity = %d", y[g].product_id, y[g].price, y[g].quantity);
1711     printf("\n\n");
1712 }
1713 }
1714 if (f == 1)
1715 {
1716     printf("Not Found");
1717 }
1718 else if (h == 6)
1719 {
1720     int f = 1;
1721     printf("Enter Quantity = ");
1722     scanf("%d", &f);
1723     for (g = 0; g < num; g++)
1724     {
1725         if (f == y[g].quantity)
1726         {
1727             printf("\n");
1728             printf("Serial number = %d\n", g);
1729             printf("Name = ");
1730             puts(y[g].medicine_name);
1731             printf("Availability = ");
1732             puts(y[g].availability);
1733             printf("Product id = %d\nPrice = %d\nQuantity = %d", y[g].product_id, y[g].price, y[g].quantity);
1734             printf("\n\n");
1735         }
1736         if (f == 1)
1737             printf("Not Found\n\n");
1738     }
1739     else
1740     {
1741         printf("\n\nInvalid input\n\n");
1742     }
1743 void delete_medicine()
1744 {
1745     int f, h;
1746     printf("Enter the serial number of the medicine that you want to delete=");
1747     scanf("%d", &f);

```

```

C minor_main.c minor_main.c...
1746     scanf("%d", &f);
1747     if (f < num)
1748     {
1749         printf("What do you want ?\n");
1750         printf("1.Remove the whole record\n2.Remove Name\n3.Remove Availability\n4.Remove Quantity\n5.Remove Product id\n6.Remove Price\n0Option = ");
1751         scanf("%d", &h);
1752         if (h == 1)
1753         {
1754             while (f < num)
1755             {
1756                 strcpy(y[f].medicine_name, y[f + 1].medicine_name);
1757                 strcpy(y[f].availability, y[f + 1].availability);
1758                 y[f].quantity = y[f + 1].quantity;
1759                 y[f].product_id = y[f + 1].product_id;
1760                 y[f].price = y[f + 1].price;
1761                 f++;
1762             }
1763             num--;
1764         }
1765         else if (h == 2)
1766         {
1767             strcpy(y[f].medicine_name, "      ");
1768         }
1769         else if (h == 3)
1770         {
1771             strcpy(y[f].availability, "      ");
1772         }
1773         else if (h == 4)
1774         {
1775             y[f].quantity = 0;
1776         }
1777         else if (h == 5)
1778         {
1779             y[f].product_id = NULL;
1780         }
1781         else if (h == 6)
1782         {
1783             y[f].price = NULL;

```

```
C minor_main.c minor_main.c...
1783     y[f].price = NULL;
1784 }
1785 }
1786 else
1787     printf("\n\nInvalid Serial Number\n");
1788 }
1789 void read_medicine()
1790 {
1791     FILE *fp = fopen("medicine.txt", "r");
1792     if (fp == NULL)
1793     {
1794         fp = fopen("medicine.txt", "w");
1795         fclose(fp);
1796         printf("File does not exist, I JUST CREATED IT, exiting...\n\n");
1797         return 0;
1798     }
1799
1800     num = fread(y, sizeof(struct m), 100, fp);
1801     fclose(fp);
1802 }
1803 void medicine_write()
1804 {
1805     FILE *fp = fopen("medicine.txt", "w");
1806     if (fp == NULL)
1807     {
1808         printf("Error");
1809         exit(1);
1810     }
1811     fwrite(y, sizeof(struct m), num, fp);
1812     fclose(fp);
1813 }
1814
1815 }
1816
```

3.5 Output Design

```
*****
*****
**
Hospital Management System
**
*****
*****
1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit
Please enter the module you want to access = 1

Enter your choice

1. Add patient Information
2. View patient Information
3. Search patient
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

Option=1

Already data inputed on the database =4

How many entry do you want to add=
1

Enter patient's name = Jackey
Enter disease = Dengue
Enter age = 18
Enter room number = 046
```

```
Enter phone number = 9837492745
```

```
Enter your choice
```

1. Add patient Information
2. View patient Information
3. Search patient
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

```
Option=2
```

```
Serial number = 0
Patient_name = vansh sharma
Disease = oversmartness
Phone number = 9289890909
Room number = 8
Age=20
```

```
Serial number = 1
Patient_name = amandeep
Disease = helpness
Phone number = 8287918611
Room number = 1
Age=20
```

```
Serial number = 2
Patient_name = Jojo
Disease = Fever
Phone number = 8726927387
```

```
Room number = 7
```

```
Age=21
```

```
Serial number = 3
```

```
Patient_name = Muskan
```

```
Disease = Malaria
```

```
Phone number = 9278435671
```

```
Room number = 56
```

```
Age=19
```

```
Serial number = 4
```

```
Patient_name = Jackey
```

```
Disease = Dengue
```

```
Phone number = 9837492745
```

```
Room number = 46
```

```
Age=18
```

```
Enter your choice
```

1. Add patient Information
2. View patient Information
3. Search patient
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

```
Option=3
```

```
By what do you want to patient_search?
```

- 1.Serial no.
- 2.Patient name
- 3.Disease

4.Room number.

5.Phone no.

6.Age

Option = 1

Enter Serial number of the patient=0

Serial Number = 0

Patient name = vansh sharma

Disease = oversmartness

Phone number = 9289890909

Room number = 8

Age = 20

Enter your choice

1. Add patient Information

2. View patient Information

3. Search patient

4. Edit patient Information

5. Delete patient Information

6. Exit Patient

Option=4

What do you want to edit ?

Enter your option

1.Patient name

2.Disease

3.Age

4.Room number

5.Phone no.

Option=3

Enter the serial no of that patient = (0 - 4)=2

```
Enter the new age = 45
```

```
Enter your choice
```

1. Add patient Information
2. View patient Information
3. Search patient
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

```
Option=5
```

```
Enter the serial number of the patient that you want to delete = 0
```

```
What do you want ?
```

- 1.Remove the whole record
- 2.Remove Patient name
- 3.Remove Disease
- 4.Remove Age
- 5.Remove Room number
- 6.Remove Phone number

```
Option = 1
```

```
Enter your choice
```

1. Add patient Information
2. View patient Information
3. Search patient
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

Option=6

If you want to continue module press 1

- 1
- 1. Patient
- 2. Doctor
- 3. Staff
- 4. Medicine
- 5. Billing
- 6. Exit

Please enter the module you want to access = 2

Enter your choice

- 1. Add Doctor Information
- 2. View Doctor Information
- 3. Search Doctor Information
- 4. Edit Doctor Information
- 5. Delete Doctor Information
- 6. Exit Doctor

Option=1

Already data inputed on the database = 2

How many entry do you want to add =

1

Enter doctor's name = MR Raj

Enter address = BH 493 Shalimar Bagh

Enter the age = 42

Enter doctor id = 096

Enter phone number = 9434395739

Enter your choice

1. Add Doctor Information
2. View Doctor Information
3. Search Doctor Information
4. Edit Doctor Information
5. Delete Doctor Information
6. Exit Doctor

Option=2

Serial number = 0
Doctor's name = Sarthak
Address = Mukherjee Nagar
Phone number = 9913803746
Doctor's id = 92
Age = 35

Serial number = 1
Doctor's name = Sarthak
Address = Old Ramesh Nagar
Phone number = 7918273378
Doctor's id = 11
Age = 48

Serial number = 2
Doctor's name = MR Raj
Address = BH 493 Shalimar Bagh
Phone number = 9434395739
Doctor's id = 96
Age = 42

Enter your choice

1. Add Doctor Information
2. View Doctor Information
3. Search Doctor Information
4. Edit Doctor Information
5. Delete Doctor Information
6. Exit Doctor

Option=3

By what do you want to search_doctor ?

- 1.Serial number
- 2.Doctor name
- 3.Address
- 4.Doctor id.
- 5.Phone number
- 6.Age

Option = 4

Enter Doctor's id = 11

Serial number = 1
Doctor name = Sarthak
Address = Old Ramesh Nagar
Phone number = 7918273378
Doctor id = 11
Age = 48

Enter your choice

1. Add Doctor Information
2. View Doctor Information
3. Search Doctor Information
4. Edit Doctor Information

```
5. Delete Doctor Information  
6. Exit Doctor
```

Option=4

What do you want to edit ?

Enter your option

1.Doctor's name

2.Address

3.Age

4.Doctor id

5.Phone number

Option=5

Enter the serial no of that doctor = (0 - 2)=0

Enter the new Phone number =7839489568

Enter your choice

```
1. Add Doctor Information  
2. View Doctor Information  
3. Search Doctor Information  
4. Edit Doctor Information  
5. Delete Doctor Information  
6. Exit Doctor
```

Option=5

Enter the serial number of the doctor that you want to
delete = 1

What do you want ?

```
1.Remove the whole record  
2.Remove Doctor's Name  
3.Remove Address  
4.Remove Age  
5.Remove Doctor Id.  
6.Remove Phone Number
```

Option = 1

Enter your choice

1. Add Doctor Information
2. View Doctor Information
3. Search Doctor Information
4. Edit Doctor Information
5. Delete Doctor Information
6. Exit Doctor

Option=6

If you want to continue module press 1

1

1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit

Please enter the module you want to access = 3

Enter your choice

1. Add Staff Information
2. View Staff Information
3. Search Staff
4. Edit Staff Information
5. Delete Staff Information
6. Exit Staff

Option=1

Already data inputed on the database = 0

How many entry do you want to add =

```
How many entry do you want to add =  
2
```

```
Enter Staff's name = Raju  
Enter Job = Sweeper  
Enter the Age = 19  
Enter Job id = 012  
Enter Phone Number = 93573964868
```

```
Enter Staff's name = Mohan  
Enter Job = Security Guard  
Enter the Age = 26  
Enter Job id = 04  
Enter Phone Number = 9576936839
```

```
Enter your choice
```

1. Add Staff Information
2. View Staff Information
3. Search Staff
4. Edit Staff Information
5. Delete Staff Information
6. Exit Staff

```
Option=2
```

```
Serial Number = 0  
Staff Name = Raju  
Job = Sweeper  
Phone Number = 93573964868  
Job id = 12  
Age=19
```

```
Serial Number = 1
Staff Name = Mohan
Job = Security Guard
Phone Number = 9576936839
Job id = 4
Age=26
```

Enter your choice

1. Add Staff Information
2. View Staff Information
3. Search Staff
4. Edit Staff Information
5. Delete Staff Information
6. Exit Staff

Option=3

By what do you want to search staff ?

- 1.Serial number
- 2.Staff name
- 3.Job
- 4.Job id.
- 5.Phone number
- 6.Age

Option = 6

Enter Age = 26

```
Serial Number = 1
Staff name = Mohan
Job = Security Guard
Phone number =9576936839
Job id = 4
Age = 26
```

Enter your choice

1. Add Staff Information
2. View Staff Information
3. Search Staff
4. Edit Staff Information
5. Delete Staff Information
6. Exit Staff

Option=4

What do you want to edit ?

Enter your option

- 1.Staff name
- 2.Job
- 3.Age
- 4.Job id
- 5.Phone number

Option = 2

Enter the serial no of that staff = (0 - 1)=1

Enter the new Job = Peon

Enter your choice

1. Add Staff Information
2. View Staff Information
3. Search Staff
4. Edit Staff Information
5. Delete Staff Information
6. Exit Staff

Option=5

Enter the serial number of the staff that you want to
delete=0

What do you want ?

- 1.Remove the whole record
- 2.Remove Staff Name

```
3.Remove Job  
4.Remove Age  
5.Remove Job id  
6.Remove Phone Number  
Option = 3
```

Enter your choice

1. Add Staff Information
2. View Staff Information
3. Search Staff
4. Edit Staff Information
5. Delete Staff Information
6. Exit Staff

Option=6

If you want to continue module press 1

- 1
1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit

Please enter the module you want to access = 4

Enter your choice

1. Add Medicine Information
2. View Medicine Information
3. Search Medicine Information
4. Edit Medicine Information
5. Delete Medicine Information
6. Exit Medicine

Option=1

Already data inputed on the database = 0

How many entry do you want to add =
1

Enter Medicine's Name = Cetirizine 10 Mg
Enter Availability = In Stock
Enter The Quantity = 200
Enter Product id = 093
Enter Price = 20

Enter your choice

1. Add Medicine Information
2. View Medicine Information
3. Search Medicine Information
4. Edit Medicine Information
5. Delete Medicine Information
6. Exit Medicine

Option=2

Serial number = 0
Name = Cetirizine 10 Mg
Availability = In Stock
Product id = 93
Price = 20
Quantity = 200

Enter your choice

1. Add Medicine Information
2. View Medicine Information
3. Search Medicine Information
4. Edit Medicine Information
5. Delete Medicine Information
6. Exit Medicine

Option=3

By what do you want to search medicine ?

- 1.Serial no.
- 2.Name
- 3.Availability
- 4.Product id.
- 5.Price
- 6.Quantity

Option = 2

Enter Medicine's Name = Cetirizine 10 Mg

Serial number = 0
Name = Cetirizine 10 Mg
Availability = In Stock
Product_id = 93
Price = 20
Quantity = 200

Enter your choice

1. Add Medicine Information
2. View Medicine Information
3. Search Medicine Information
4. Edit Medicine Information
5. Delete Medicine Information

```
6. Exit Medicine
```

```
Option=4
```

```
What do you want to edit ?
```

```
Enter your option
```

- 1.Name
 - 2.Availability
 - 3.Quantity
 - 4.Product id
 - 5.Price
- ```
Option=5
```
- ```
Enter the serial no of that medicine = (0 - 0)=0
```
- ```
Enter the new price = 17
```

```
Enter your choice
```

- 1. Add Medicine Information
- 2. View Medicine Information
- 3. Search Medicine Information
- 4. Edit Medicine Information
- 5. Delete Medicine Information
- 6. Exit Medicine

```
Option=5
```

```
Enter the serial number of the medicine that you want
to delete=0
```

```
What do you want ?
```

- 1.Remove the whole record
  - 2.Remove Name
  - 3.Remove Availability
  - 4.Remove Quantity
  - 5.Remove Product id
  - 6.Remove Price
- ```
Option = 1
```

Enter your choice

1. Add Medicine Information
2. View Medicine Information
3. Search Medicine Information
4. Edit Medicine Information
5. Delete Medicine Information
6. Exit Medicine

Option=6

If you want to continue module press 1

1

1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit

Please enter the module you want to access = 5

===== Patient/Customer's Invoice =====

=====

Please select your preferred operation

1. Generate Invoice
2. Show all Invoices
3. Search Invoice
4. Exit

Your choice: 1

Please enter the name of patient/customer: Aman

Please enter the number of fields : 3

Please enter the item 1: Admit Days
Please enter the quantity: 2
Please enter the unit price: 3500

Please enter the item 2: Cotton
Please enter the quantity: 1
Please enter the unit price: 200

Please enter the item 3: Paracetamol
Please enter the quantity: 2
Please enter the unit price: 40

Invoice

Date: Apr 18 2023

Invoice To: Aman

Items	Qty	Total
Admit Days	2	7000.00
Cotton	1	200.00
Paracetamol	2	80.00
Sub Total		7280.00
Discount @10%		728.00
Net Total		6552.00
CGST @9%		589.68
SGST @9%		589.68
Grand Total		7731.36

```
Do you want to save the invoice [y/n]: y
Successfully saved
Do you want to perform another operation in billing?[y/n]: y
===== Patient/Customer's Invoice =====
=====
Please select your prefered operation
1.Generate Invoice
2.Show all Invoices
3.Search Invoice
4.Exit

Your choice: 2
*****Your Previous Invoices*****

                Invoice
-----
Date:Apr 18 2023
Invoice To: Rahul
-----
Items          Qty        Total
-----
Days of Admission           2        7000.0
0
Dolo 650                  1        30.00
-----
Sub Total                 7030.00
Discount @10%              703.00
-----
Net Total                 6327.00
```

Net Total	6327.00
CGST @9%	569.43
SGST @9%	569.43

Grand Total	7465.86

Invoice

Date: Apr 18 2023

Invoice To: Aman

Items	Qty	Total
Admit Days	2	7000.00
Cotton	1	200.00
Paracetamol	2	80.00

Sub Total 7280.00

Discount @10% 728.00

Net Total 6552.00
CGST @9% 589.68
SGST @9% 589.68

Grand Total 7731.36

Do you want to perform another operation in billing? [y/n]: y

===== Patient/Customer's Invoice =====
=====

Please select your preferred operation

Please select your preferred operation

1. Generate Invoice
2. Show all Invoices
3. Search Invoice
4. Exit

Your choice: 3

Enter the name of the patient/customer: Rahul

*****Invoice of Rahul*****

Invoice

Date: Apr 18 2023

Invoice To: Rahul

Items	Qty	Total
Days of Admission	2	7000.0
0		
Dolo 650	1	30.00

Sub Total		7030.00
Discount @10%		703.00

Net Total		6327.00
CGST @9%		569.43
SGST @9%		569.43

Grand Total		7465.86

Do you want to perform another operation in billing? [y/n]: y

===== Patient/Customer's Invoice =====

Grand Total 7465.86

Do you want to perform another operation in billing?[y/n]: y

===== Patient/Customer's Invoice =====
=====

Please select your preferred operation

1. Generate Invoice
2. Show all Invoices
3. Search Invoice
4. Exit

Your choice: 4

Bye Bye :)

Do you want to perform another operation in billing?[y/n]: n

If you want to continue module press 1

- 1
2. Patient
3. Doctor
4. Staff
5. Medicine
6. Billing
7. Exit

Please enter the module you want to access = 6

PS D:\C Prog Shekhar Sir> █

CHAPTER 4

TESTING AND IMPLEMENTATION

- 4.1 Testing Methodology**
- 4.2 Testing Methodology applied**
- 4.3 Test Cases**
- 4.4 Test Data**
- 4.5 Gap Analysis (Planned Vs Achieved)**
- 4.6 Rework/ Retest**
- 4.7 Implementation**
- 4.8 Post Implementation**
- 4.9 Hardware & Software Requirement**
 - 4.9.1 Hardware Requirement**
 - 4.9.2 Software Requirement**

4.1 Testing Methodology (Types)

Testing is a process of executing a program with the significance of finding an error. A good test is one that has high probability of finding the yet unfound error. Testing should regularly uncover different classes of errors in a minimum amount of time with a minimum amount of efforts. A software composition that includes a software requirement specification, a design specification and source code.

1. A software composition that includes a test plan and procedure, any testing tool and test cases and their expected results.

Testing is divided into several distinct operations:

1. Unit Testing

Unit test comprises of a set tests performed by an individual program earlier to the integration of the unit into large system. A program unit is usually the minor free functioning part of the whole system. Module unit testing should be as complete as possible to ensure that each representation handled by each module has been tested. All the units that makeup the system must be tested independently to assure that they work as required. During unit testing some errors were lifted and all of them were rectified and handled well. The result was so simple satisfactory and it worked well.

2. Integration Testing

It is the phase in software testing in which specific software modules are combined and tested as a group. It happen after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in big modules, keep tests defined in an integration test plan to those aggregates, and delivers as its output the coordinate system ready for system testing.

4.2 Testing Methodology Applied (Why)

The purpose of integration testing is to confirm functional, performance, and reliability requirements placed on major design items. These design items i.e., assemblies are exercised through their interfaces using testing, success and error cases being pretend via relevant guideline and data inputs. Test cases are built to test whether all the elements within assemblies interact correctly or not, for example across procedure calls or process activities, and this is done after testing individual modules, i.e., unit testing. The idea of a building block approach, in which confirm assemblies are added to a verified base which is then used to support the testing of further assemblies .Software Integration Testing is executes according to the Software Development Life Cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatic complexity of the software and software architecture, reusability of modules and life-cycle/ versioning management

4.3 Test cases.

4.3.1 Characteristics of a good test case -

***ACCURATE : EXTRACT THE PURPOSE.**

***ECONOMICAL : NO UNECESSARY STEPS OR WORDS.**

***TRACEABLE : CAPABLE TO TRACE THE REQUIREMENTS.**

***REPEATABLE : CAN BE USED TO EXECUTE THE TEST OVER AND OVER.**

***REUSABLE : CAN BE REUSED**

Test case 1: result on entering correct module number

```
*****
*****
**
Hospital Management System
**
*****
*****
1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit
Please enter the module you want to access = 1

Enter your choice

1. Add patient Information
2. View patient Information
3. Search patient
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

Option=
```

If the user enters correct module number then he can access that module and its functionalities.

Test case 2: result on entering incorrect module number

```
*****
*****
**
Hospital Management System
**
*****
*****
1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit
Please enter the module you want to access = 7

Invalid Serial number

If you want to continue press 1
1
1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit
Please enter the module you want to access = █
```

If the user enters wrong module number then he gets a chance to correct his option by continuing.

Test case 3: Multiple records for patient, doctor, staff and medicine can be added at a time

```
Enter your choice

1. Add patient Information
2. View patient Information
3. Search patient .
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

Option=1

Already data inputed on the database =2

How many entry do you want to add=
2

Enter patient's name = Jojo
Enter disease = Fever
Enter age = 21
Enter room number = 007
Enter phone number = 8726927387

Enter patient's name = Muskan
Enter disease = Malaria
Enter age = 19
Enter room number = 056
Enter phone number = 9278435671
```

Test case 4: Multiple records for patient, doctor, staff, medicine and bills can be search at a time.

```
Enter your choice

1. Add Doctor Information
2. View Doctor Information
3. Search Doctor Information
4. Edit Doctor Information
5. Delete Doctor Information
6. Exit Doctor

Option=3
By what do you want to search_doctor ?
1.Serial number
2.Doctor name
3.Address
4.Doctor id.
5.Phone number
6.Age

Option = 2
Enter Doctor's name = Sarthak

Serial number=0
Doctor name = Sarthak
Address = Mukherjee Nagar
Phone number = 9913803746
Doctor id = 92
Age = 35

Serial number=1
Doctor name = Sarthak
Address = Old Ramesh Nagar
Phone number = 7918273378
Doctor id = 11
Age = 48
```

Test case 5: All records for patient, doctor, staff, medicine and bills can be listed at a time.

```
Enter your choice
```

1. Add Staff Information
2. View Staff Information
3. Search Staff
4. Edit Staff Information
5. Delete Staff Information
6. Exit Staff

```
Option=2
```

```
Serial Number = 0
Staff Name = Jimmy
Job = Sweeper
Phone Number = 9247582083
Job id = 6
Age=26
```

```
Serial Number = 1
Staff Name = Micheal
Job = Security Guard
Phone Number = 8739237463
Job id = 4
Age=35
```

```
Serial Number = 2
Staff Name = Jasmine
Job = Receptionist
Phone Number = 9999928289
Job id = 2
Age=23
```

Test case 6: Bills can be generated in proper computerized format.

```
===== Patient/Customer's Invoice =====

Please select your prefered operation

1.Generate Invoice
2.Show all Invoices
3.Search Invoice
4.Exit

Your choice: 1

Please enter the name of patient/customer: Rahul

Please enter the number of fields : 2

Please enter the item 1: Days of Admission
Please enter the quantity: 2
Please enter the unit price: 3500

Please enter the item 2: Dolo 650
Please enter the quantity: 1
Please enter the unit price: 30
```

```
Invoice
-----
Date:Apr 18 2023
Invoice To: Rahul
-----
Items          Qty        Total
-----
Days of Admission           2          7000.00
Dolo 650                  1          30.00
-----
Sub Total                7030.00
Discount @10%              703.00
-----
Net Total                6327.00
CGST @9%                  569.43
SGST @9%                  569.43
-----
Grand Total               7465.86
-----
Do you want to save the invoice [y/n]: y
Successfully saved
Do you want to perform another operation?[y/n]: █
```

4.4 Test Data

Test data is exactly the input given to a software program. It represents data that affects or is affected by the implementation of the specific module. Some data may be used for positive testing, typically to check that a given set of input to a given function produces a predicted result. Other data may be used for negative testing to test the capacity of the program to handle unusual, extreme, exceptional, or unexpected input. Poorly designed testing data may not test all possible test scenarios which will prevent the quality of the software.

4.5 Gap Analysis (Planned Vs Archived)

Planning measures means the time your resources spend working that can be billed to a client. Actual utilization, then, is a view into how much work an exact or a specific resource has scheduled, both in the past and future. What's important about this specific report is that it highlights the relationship between an individual's actuals and scheduled hours measured to predefined organization and personal targets that are necessary to achieve to improve or meet complete company profitability goals.

This metric is useful for you to know where you land month-over-month — are resources hitting scheduled targets, and if not, you can dig into what happened. You can also use this data in corporate to see where improvements need to be made, for example you are running at 62% utilization, but target is at 65%, so you are 3% under.

4.6 REWORK

If there is any error, it is important to rectify it.

Also, it should be rechecked and retested on different parameters after a specific period of time.

Checking and making compatibility is also necessary.

4.7 Implementation

After testing of the system, proposed system is installed at the Hospital. Implementation is a process of switching the manual system with a newly developed system and making it useful without disturbing the functionality of the organization.

There are 3 types of implementation

- 1) Fresh Implementation
- 2) Replacement Implementation
- 3) Modified Implementation

4.8 Post implementation

A post implementation review is an evaluation of a system in terms of the extent to which the system accomplishes stated objective system accomplishes stated objective and actual project cost exceeds initial estimates.

It is normally carried out by people who have an independent view point and are not responsible for the development and maintenance of the system while evaluating a system, the following points are normally considered.

- 1) Performance Evaluation
- 2) Cost Analysis
- 3) Time Analysis
- 4) User Satisfaction
- 5) Documentation Review Plan
- 6) Hardware Review Plan
- 7) Ease for modification

8) Failure Rate

4.9Hardware & Software Requirement

4.9.1 HARDWARE REQUIREMENTS

Processor	:	Intel Core i5@3.1Ghz
Clock speed	:	1000 Mhz
RAM	:	8 GB of RAM
HDD	:	1 TB
Monitor	:	any
Keyboard	:	108 keys
Mouse	:	2 button mouse

4.9.2 SOFTWARE REQUIREMENTS

OS	:	MS WINDOWS 10
Front End	:	C Programming

CHAPTER 5

CONCLUSION AND REFERENCE

- 5.1 Conclusion**
- 5.2 Limitation of System**
- 5.3 Future Scope for Modification**
- 5.4 References/ Bibliography**

5.1 CONCLUSION

The project “Hospital Management System” is completed, satisfying the need design for specifications. The system provides a user-friendly interface. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work outstanding. Thus the system has fulfilled all the target or goals identified and is able to replace the existing system. The constraints are met and overcome successfully. The system is designed as like it was chooses in the design phase. This software has a user-friendly screen that allow the user to use without any inconvenience.

The application has been tested with live data and has provided an outstanding result. Hence the software has proved to work efficiently.

5.2 LIMITATIONS OF SYSTEM

Existing system refers to the system that is being observe till now. The existing system requires more computational time, more manual work, and the complexity involved in Selection of features is high. After inserting the data to database, staff need not to worry about the orders received through the system and hence decrease the manual labor. One of the best features of the system is to refund the amount on cancellation of order by customer. But in early years it is not easy for the staff to modify the whole records manually and it also increases labour of the employees. Existing system requires large paper work and even a small transaction require many papers fill. Loss of even a single paper led to difficult situation because all the papers are connected with each other. In existing system it is not easy to share data. Also the two departments in an organization cannot interact with each other without the real movement of data.

5.3 Future Scope for Modification

To avoid all these limitations and make the working more perfectly the system needs to be computerized.

5.4 References/ Bibliography

ITEMS:

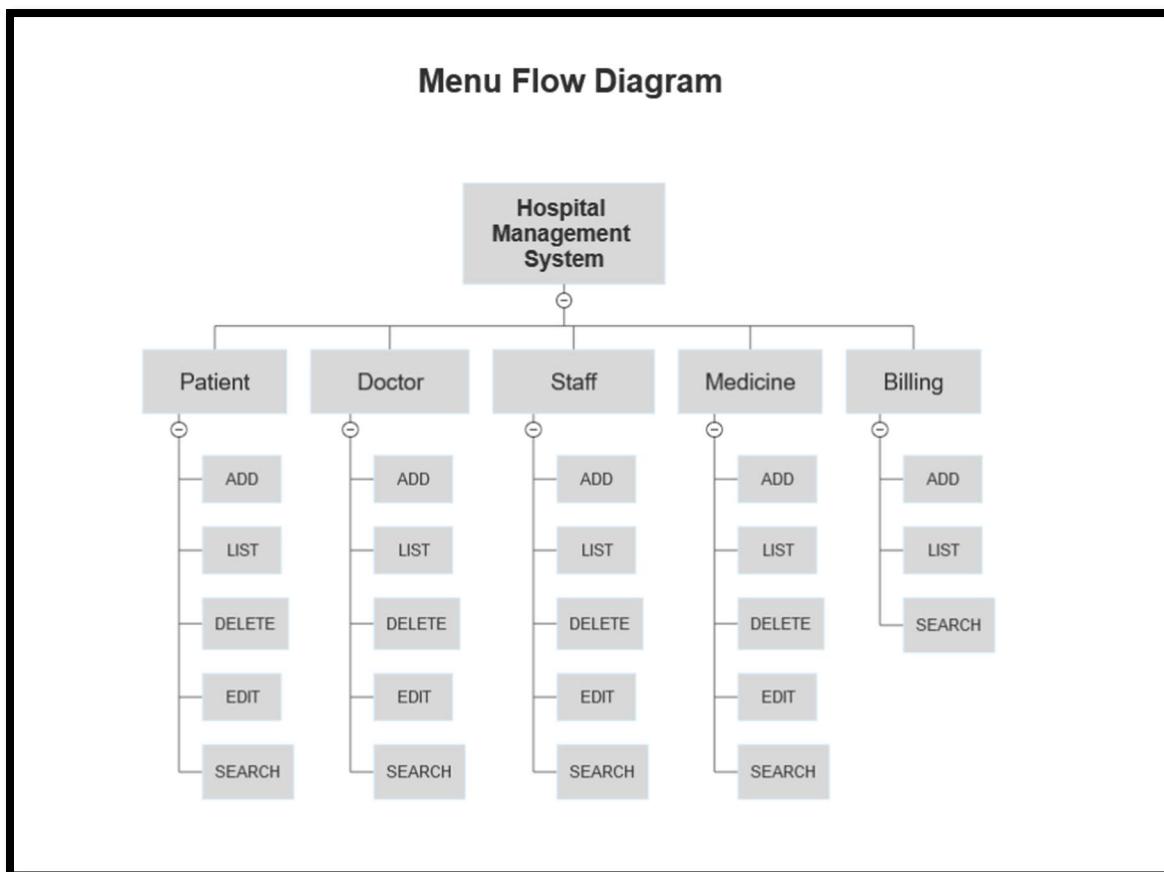
- 1.** Let us C by Yashwant Kanetkar
- 2.** Programming in Ansi C
- 3.** Expert C Programming

CHAPTER 6

ANNEXURES

- 6.1 Menu Flow Diagram**
- 6.2 Sample Input**
- 6.3 Sample Output**
- 6.4 Plagiarism certificate**

6.1 Menu Flow Diagram



6.2 SAMPLE INPUT

```
Enter your choice
```

- 1. Add patient Information
- 2. View patient Information
- 3. Search patient
- 4. Edit patient Information
- 5. Delete patient Information
- 6. Exit Patient

```
Option=1
```

```
Already data inputed on the database =4
```

```
How many entry do you want to add=
```

```
1
```

```
Enter patient's name = Samarth
```

```
Enter disease = Cicken Pox
```

```
Enter age = 17
```

```
Enter room number = 045
```

```
Enter phone number = 9434843734
```

Enter your choice

1. Add Doctor Information
2. View Doctor Information
3. Search Doctor Information
4. Edit Doctor Information
5. Delete Doctor Information
6. Exit Doctor

Option=1

Already data inputed on the database = 2

How many entry do you want to add =
1

Enter doctor's name = Shivam Sharma
Enter address = 42 Burari
Enter the age = 43
Enter doctor id = 067
Enter phone number = 9476854048

6.3 SAMPLE OUTPUT

```
1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit
Please enter the module you want to access = 1

Enter your choice

1. Add patient Information
2. View patient Information
3. Search patient
4. Edit patient Information
5. Delete patient Information
6. Exit Patient

Option=5
Enter the serial number of the patient that you want to delete = 0
What do you want ?
1.Remove the whole record
2.Remove Patient name
3.Remove Disease
4.Remove Age
5.Remove Room number
6.Remove Phone number
Option = 1
```

```
1. Patient
2. Doctor
3. Staff
4. Medicine
5. Billing
6. Exit
Please enter the module you want to access = 2
Enter your choice

1. Add Doctor Information
2. View Doctor Information
3. Search Doctor Information
4. Edit Doctor Information
5. Delete Doctor Information
6. Exit Doctor

Option=5
Enter the serial number of the doctor that you want to delete = 1
What do you want ?
1.Remove the whole record
2.Remove Doctor's Name
3.Remove Address
4.Remove Age
5.Remove Doctor Id.
6.Remove Phone Number
Option = 1
```

