Table of Contents

[Project Title 2](#_Toc75504964)

[Introduction 3](#_Toc75504965)

[Entities 4](#_Toc75504966)

[Modules 5](#_Toc75504967)

[ER-Diagrams 6](#_Toc75504968)

[Use Case Diagrams 8](#_Toc75504969)

[Screen Designs 11](#_Toc75504970)

[Actual Screens 20](#_Toc75504971)

[Validation Rules 30](#_Toc75504972)

[Program Code 35](#_Toc75504973)

[Conclusion 85](#_Toc75504974)

# Project Title

Hospital Data Management System (H.D.M.S)

By

Shivanki Srivastava ECE C 2019B311040

Team Name

CYAPPA

Semester

4th SEM

Under the guidance of

Dr B P Sharma

# Introduction

Hospital management system is a computer system that helps manage the information related to health care and aids in the job completion of health care providers effectively. They manage the data related to all departments of healthcare such as

* Patient details
* Doctor details
* Staff details
* Resources available

Patient management includes patient details like name, patient ID, age, gender, contact details, issue We can also update the details. We can add new patient details in the saved data. Doctor management includes doctor details like name, doctor ID, age, gender, availability, qualification, specialization, experience, contact details. We can also update the details. We can add new details in the saved data. Staff management includes staff details like staff name, staff ID, contact details, role. We can also add new details. This tells us about the data of equipments like oxygen cylinder, beds, ambulances, ventilators, etc.

This system monitors the essential data and details of the hospital.

# Entities

**Patient**

* This includes patient details like name , patient ID , age , gender, contact details , issue .
* We can view the list of all the patients.
* We can delete the save record.

**Doctor**

* This includes doctor details like name, doctor ID, age, gender, availability, qualification, specialization, experience, contact details.
* We can view the list of all the doctors.
* We can delete the save record.

**Staff**

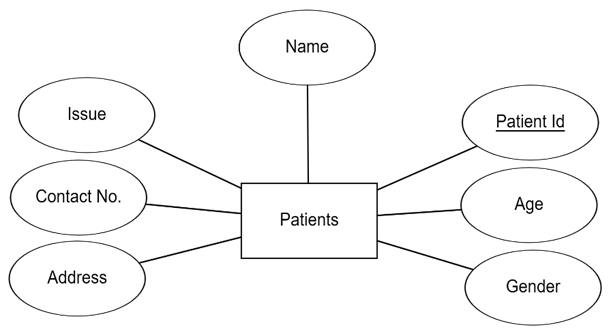
* This includes staff details like staff name, staff ID, contact details, role.
* We can view the list of all the staff.
* We can delete the save record.

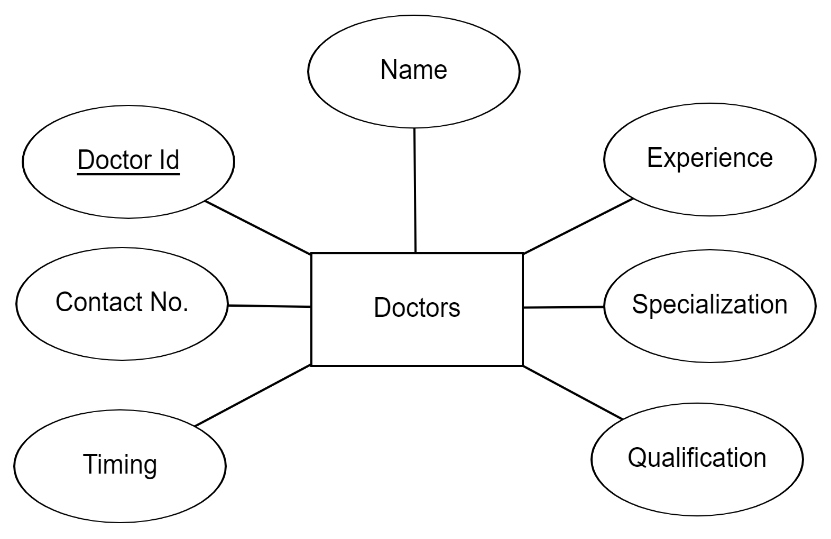
# Modules

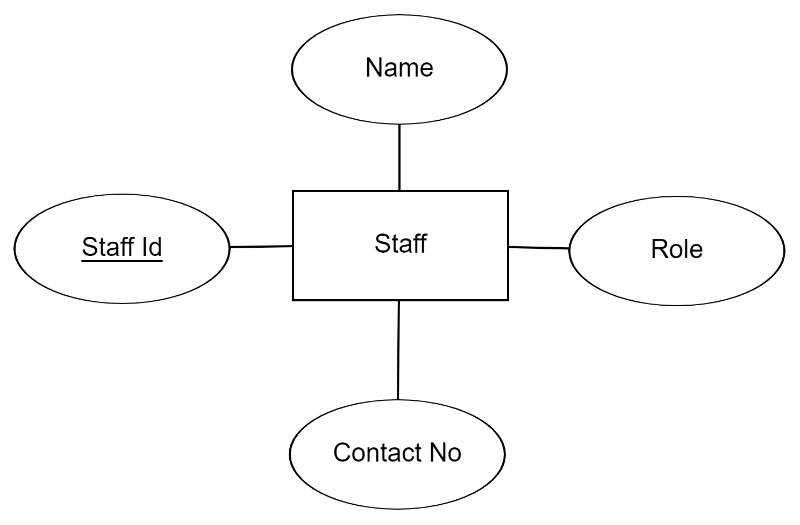
This system is designed to perform the certain tasks. These tasks are as follows:-

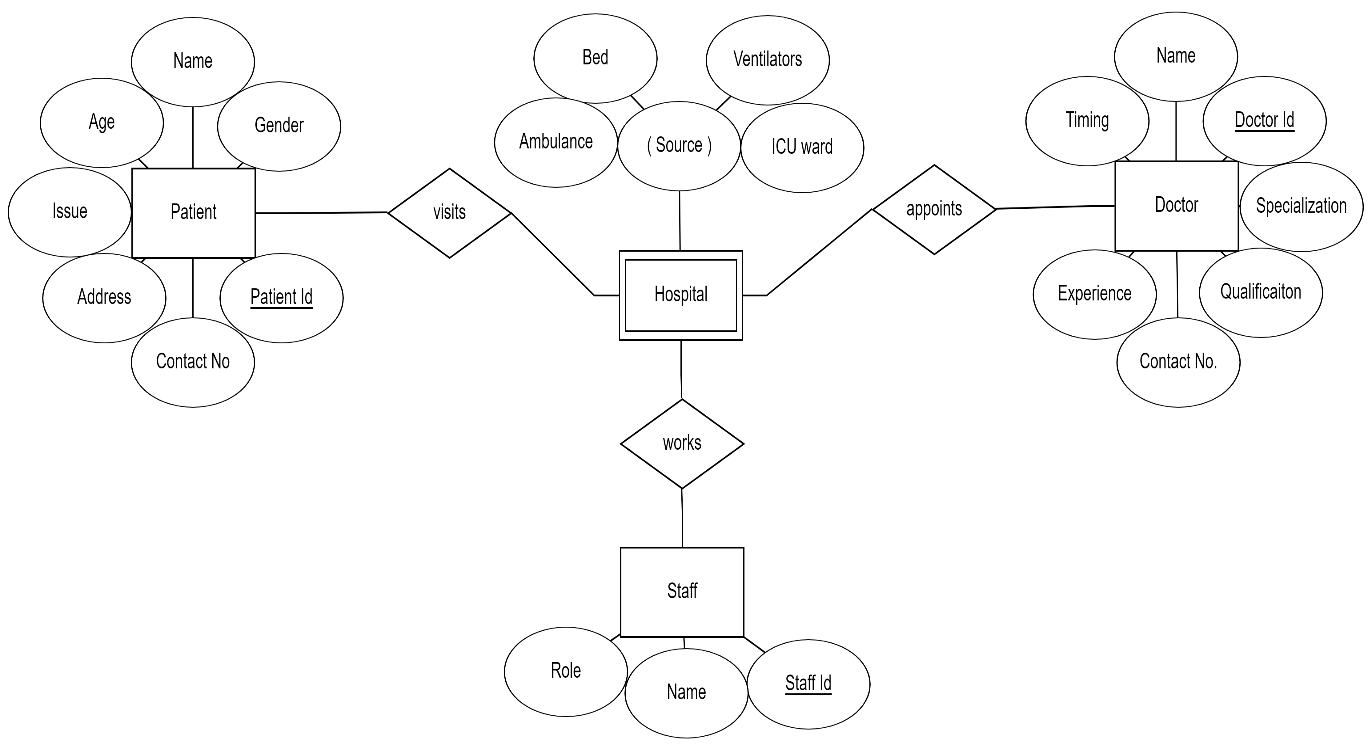
* It can add new patient, doctor, staff, resources.
* It can save as well as delete the record.
* It can show the list of doctors, patient and staff.
* It can show the data of the resources available in the hospital.
* It tells us about the availability(timing) of the doctor.

# ER-Diagrams



****

****



# Use Case Diagrams



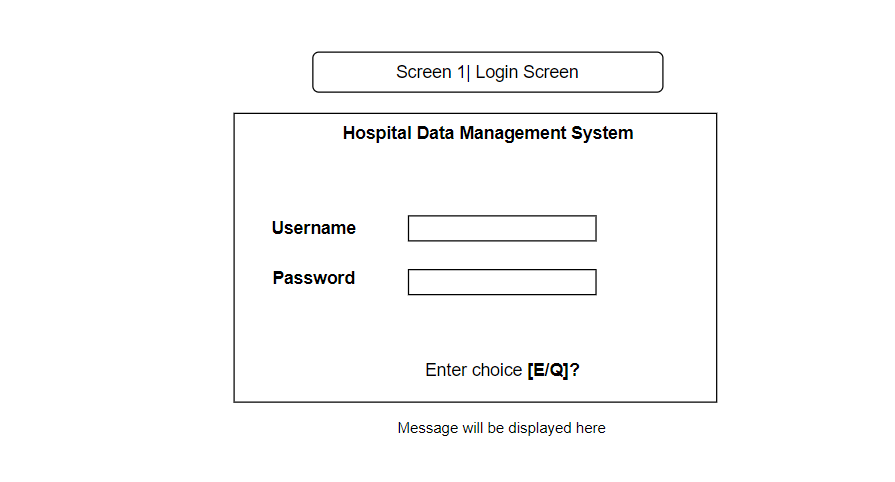


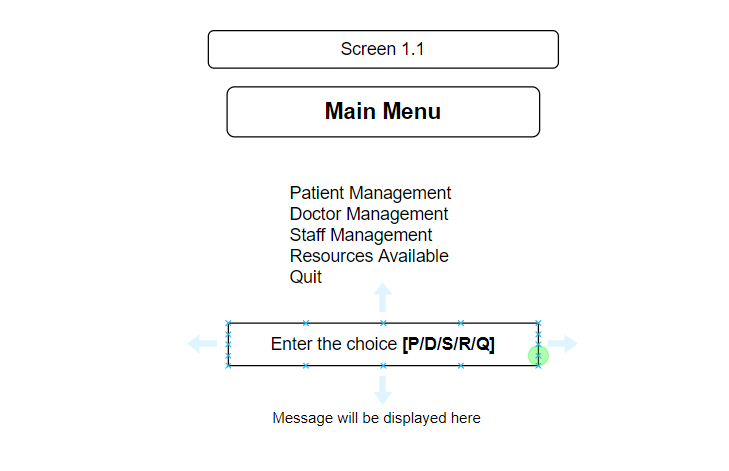


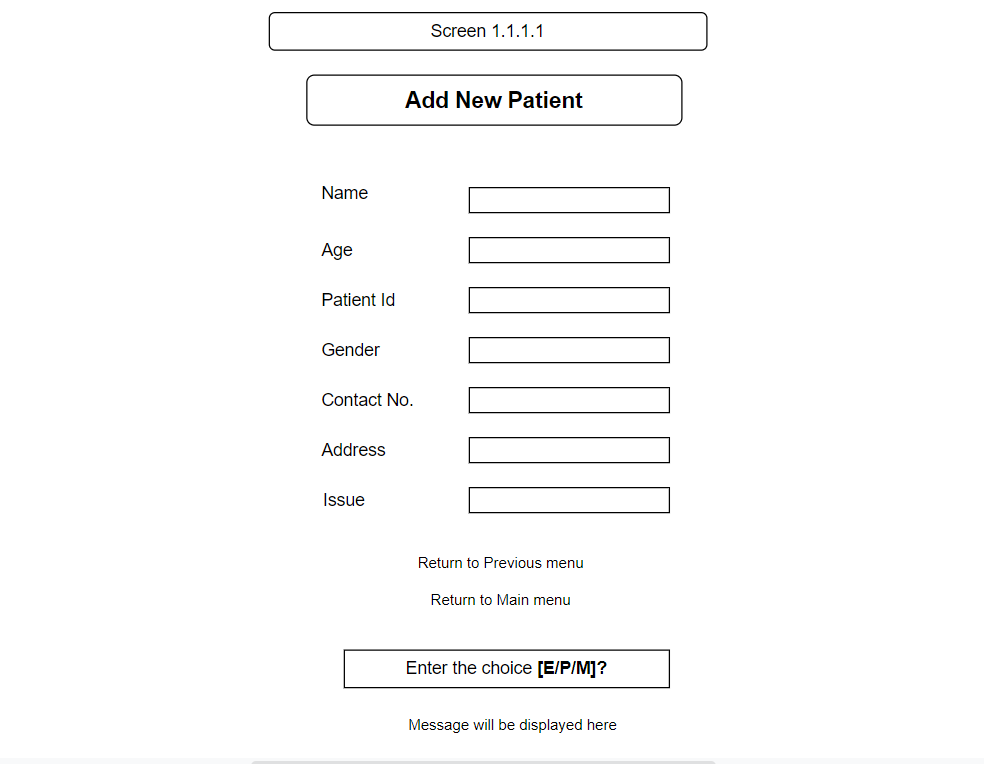


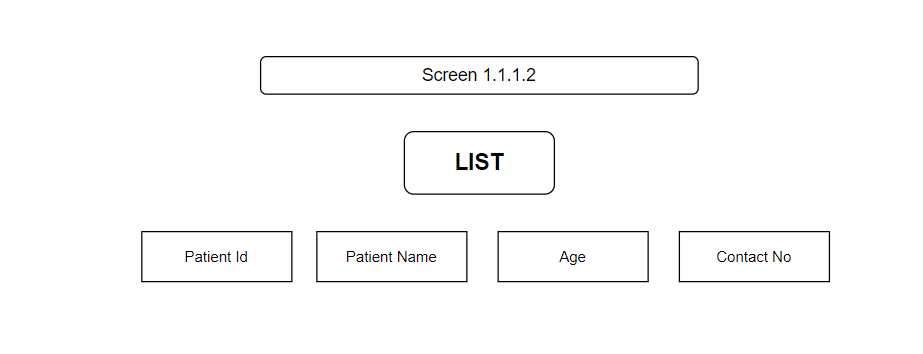


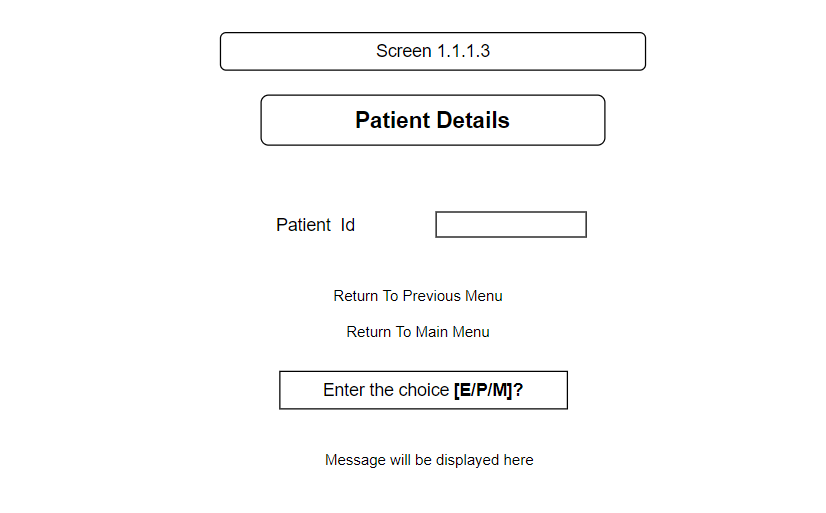
# Screen Designs

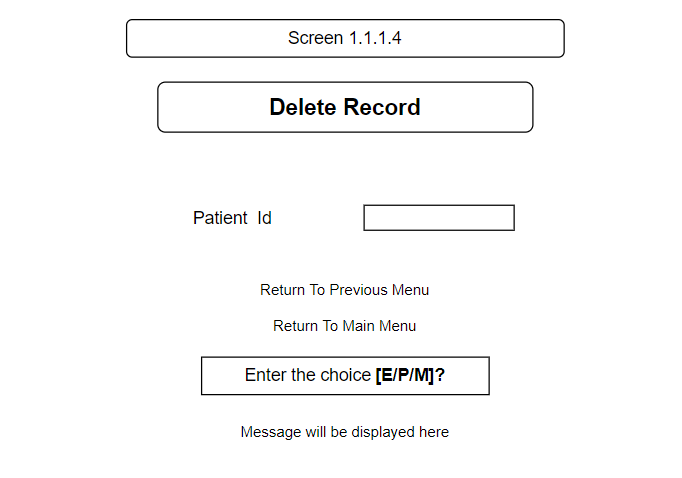


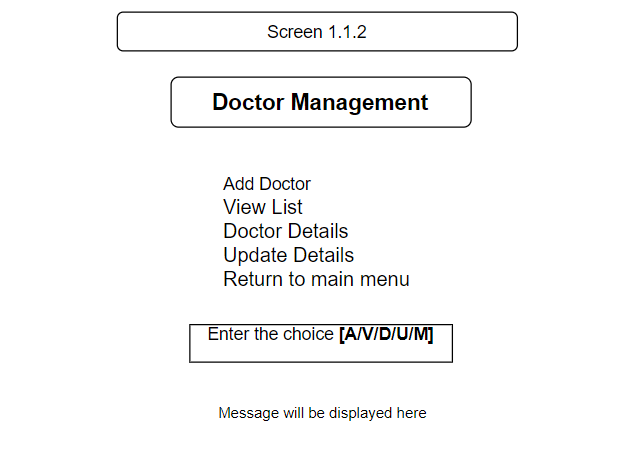


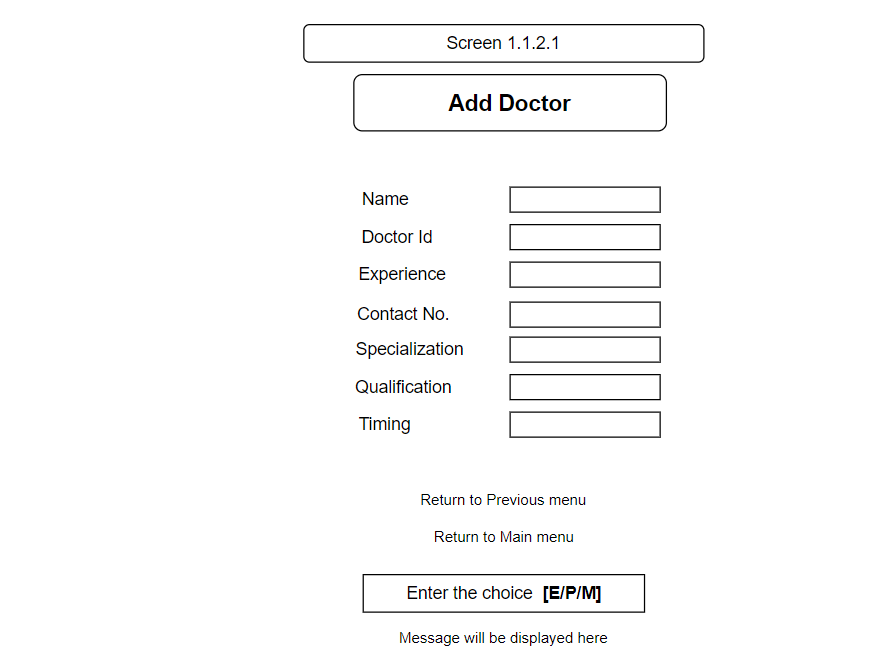


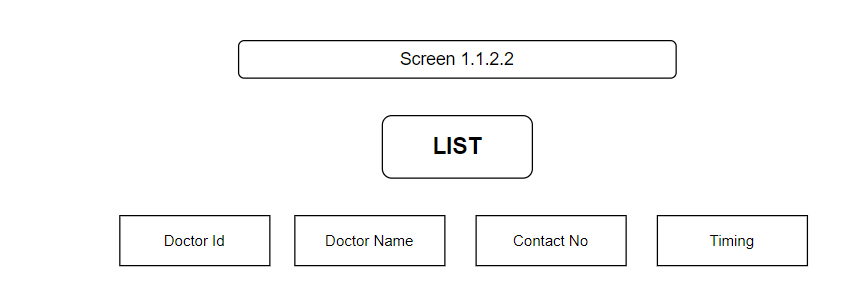


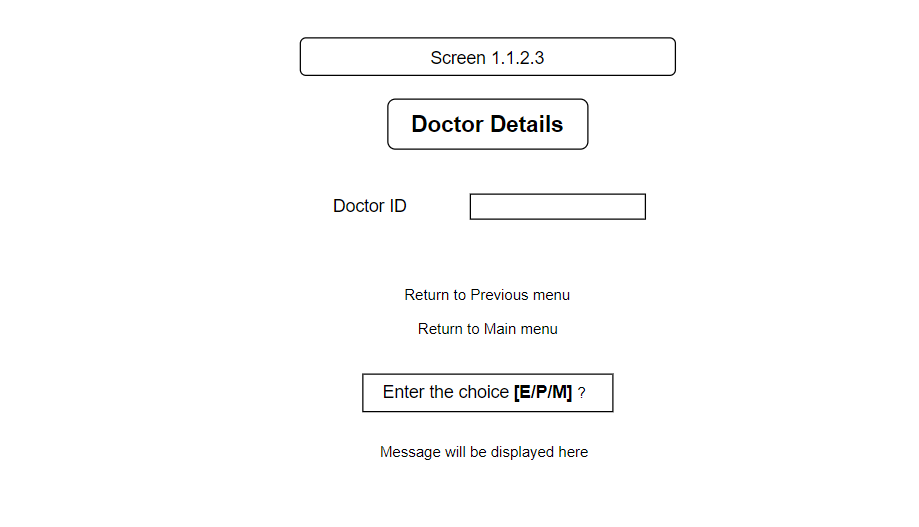


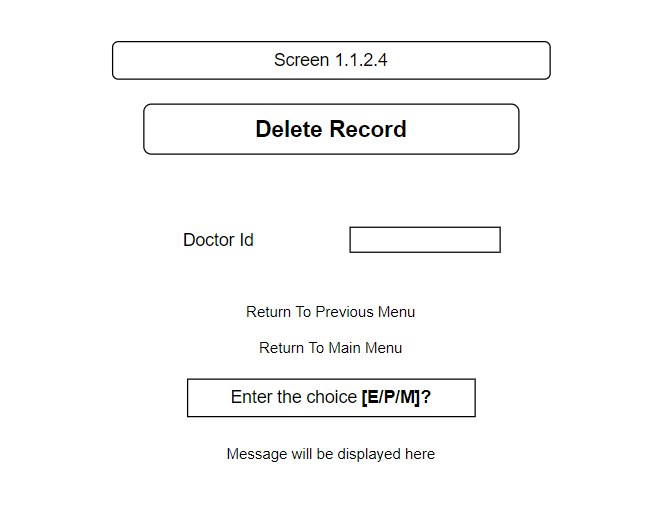


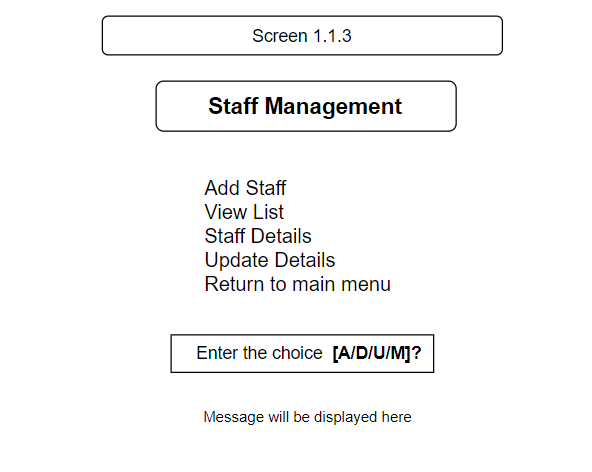


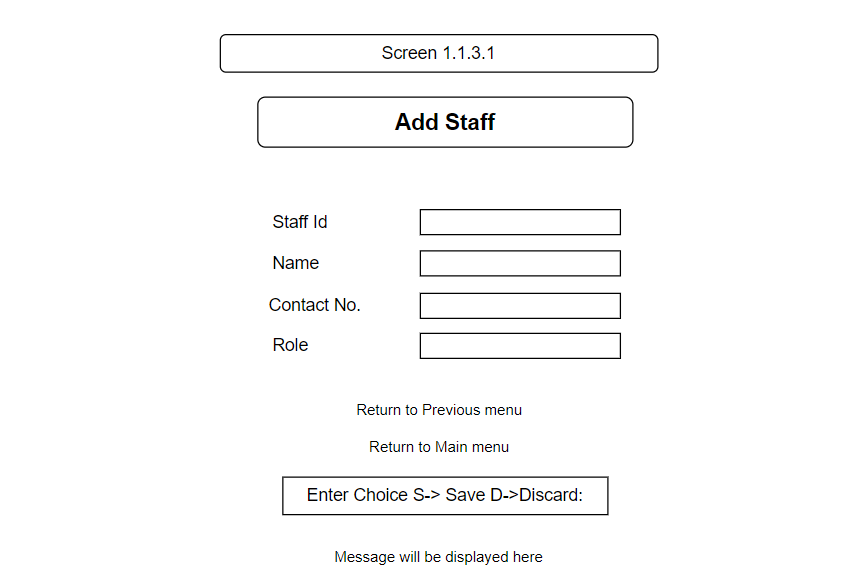


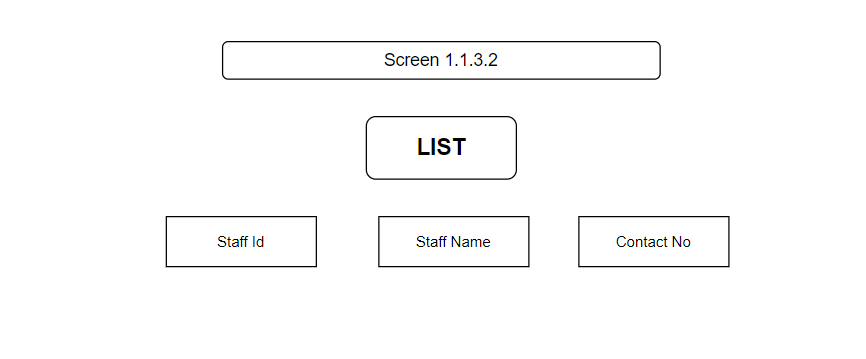


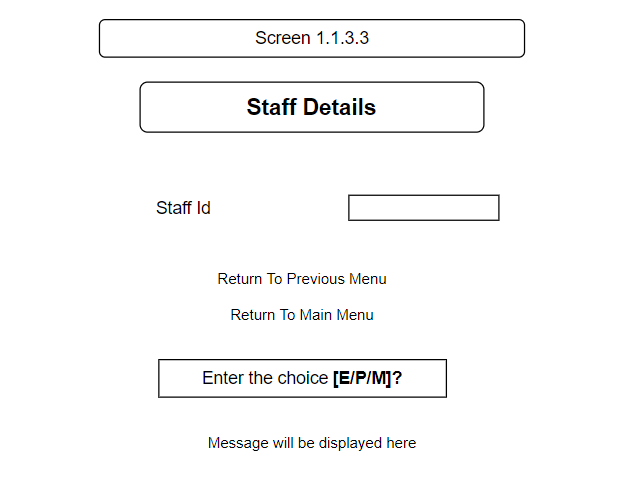


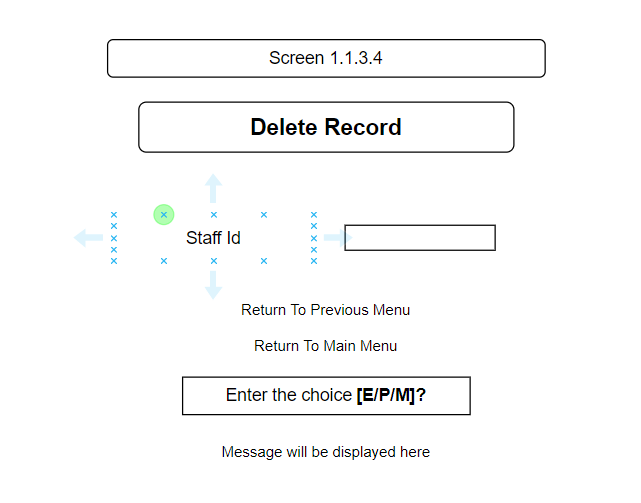


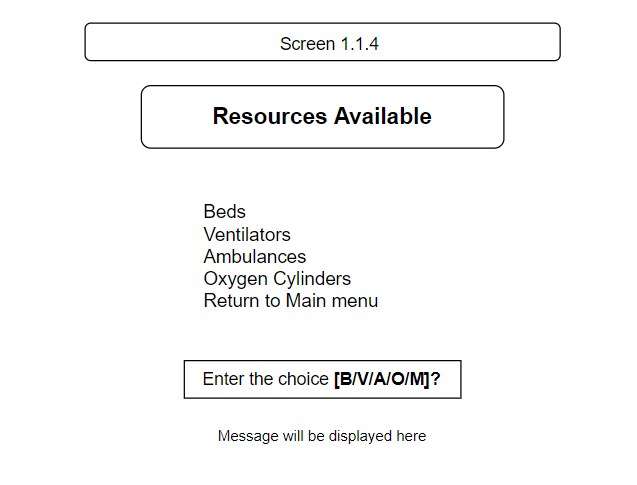


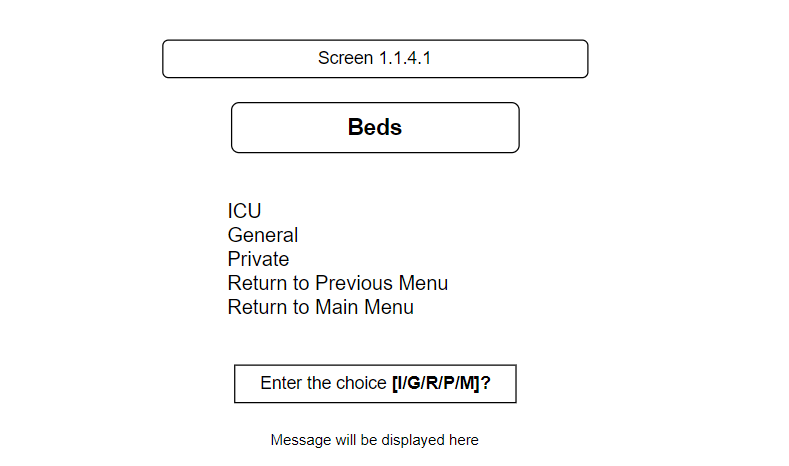






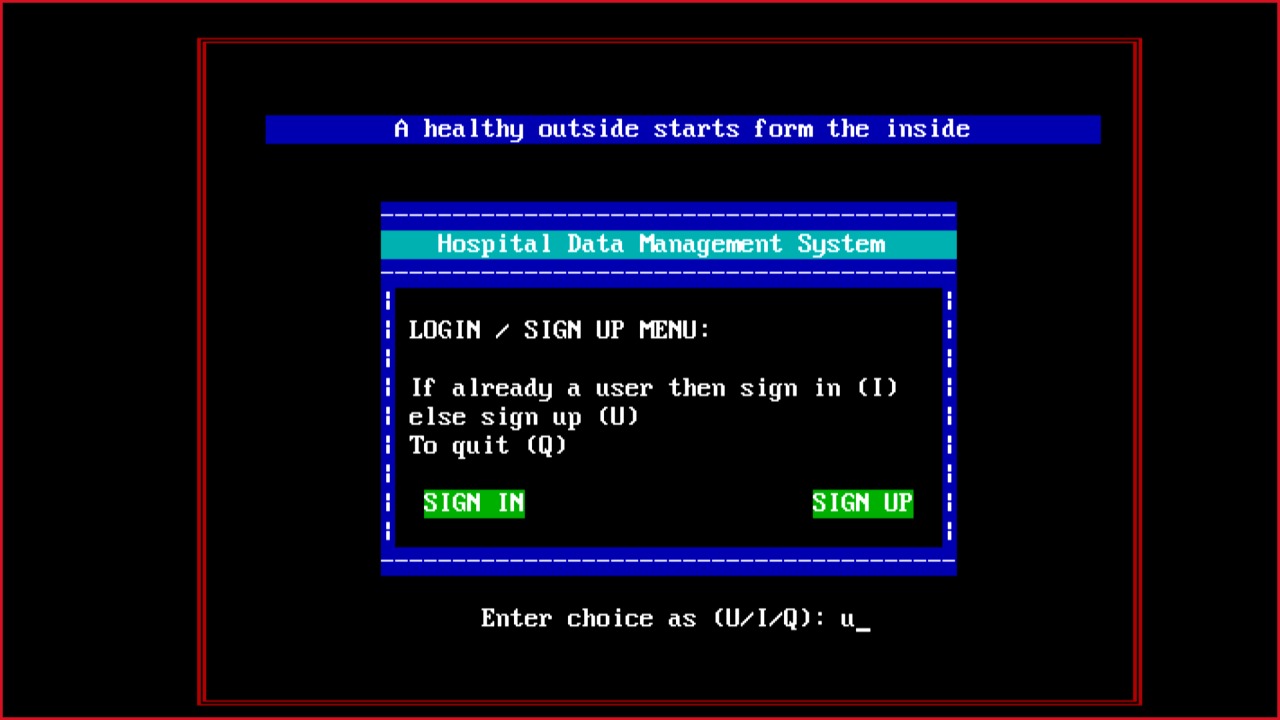




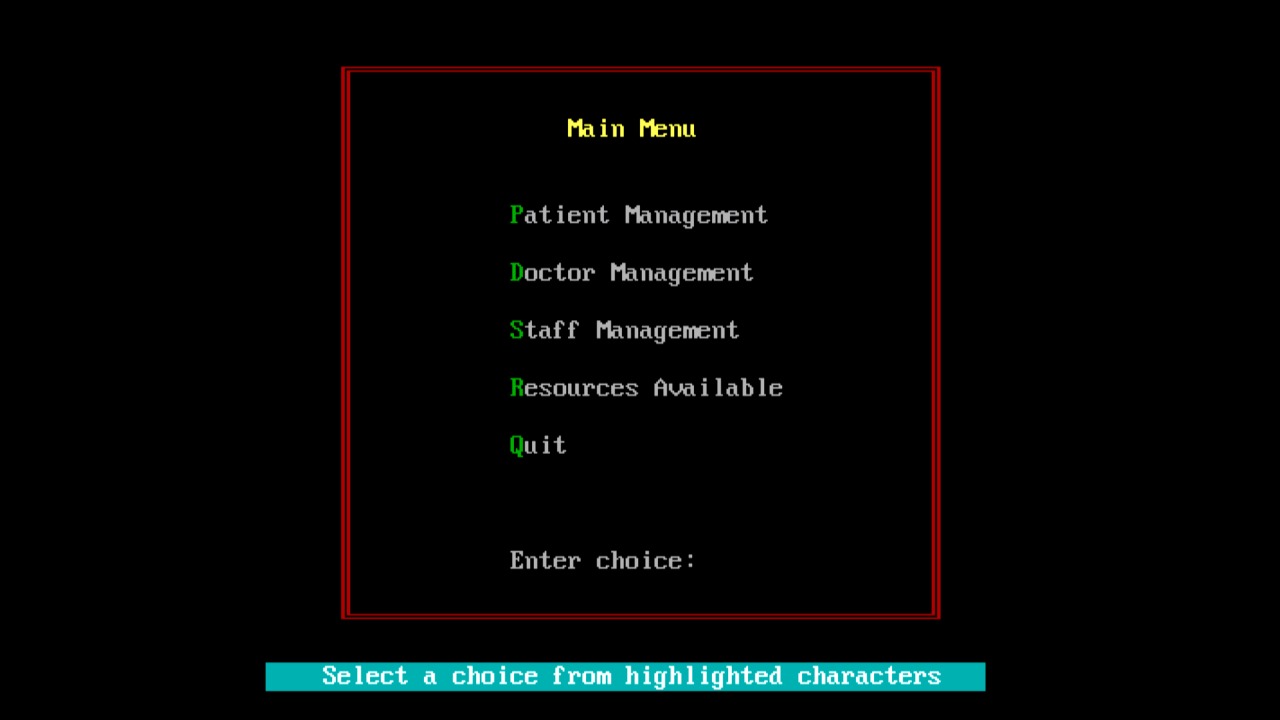


# Actual Screens

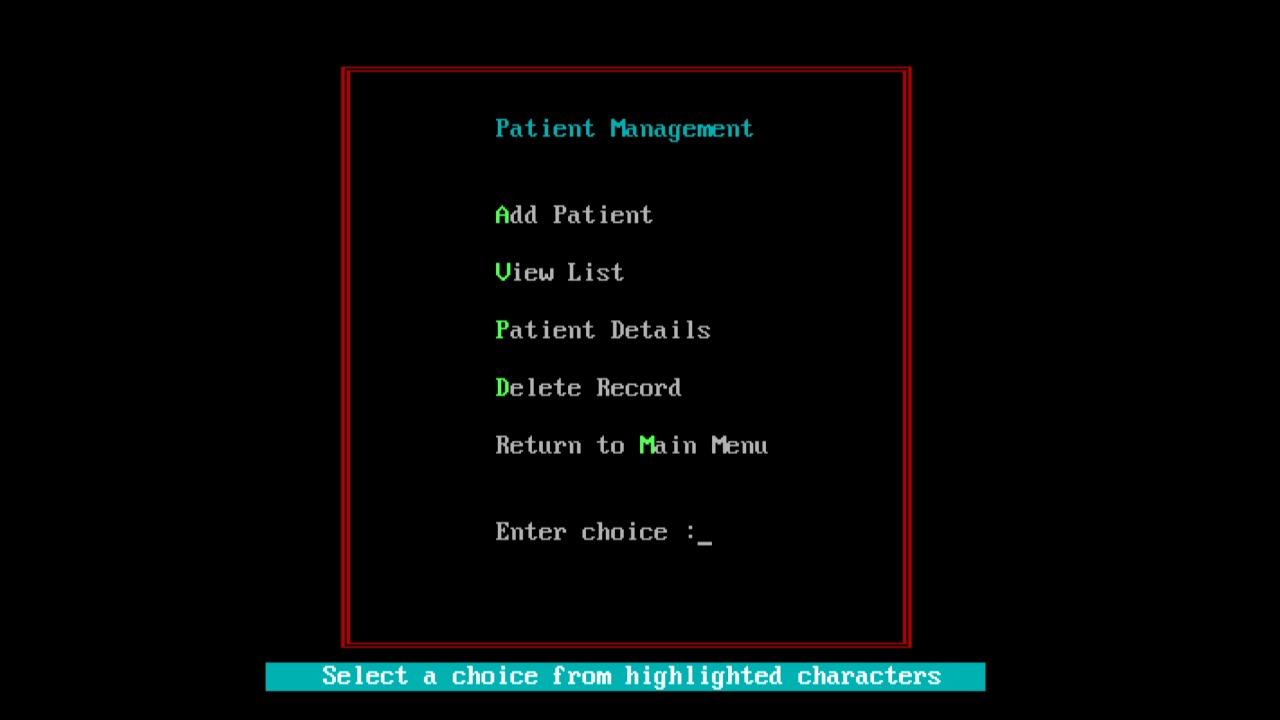
**Screen 1**



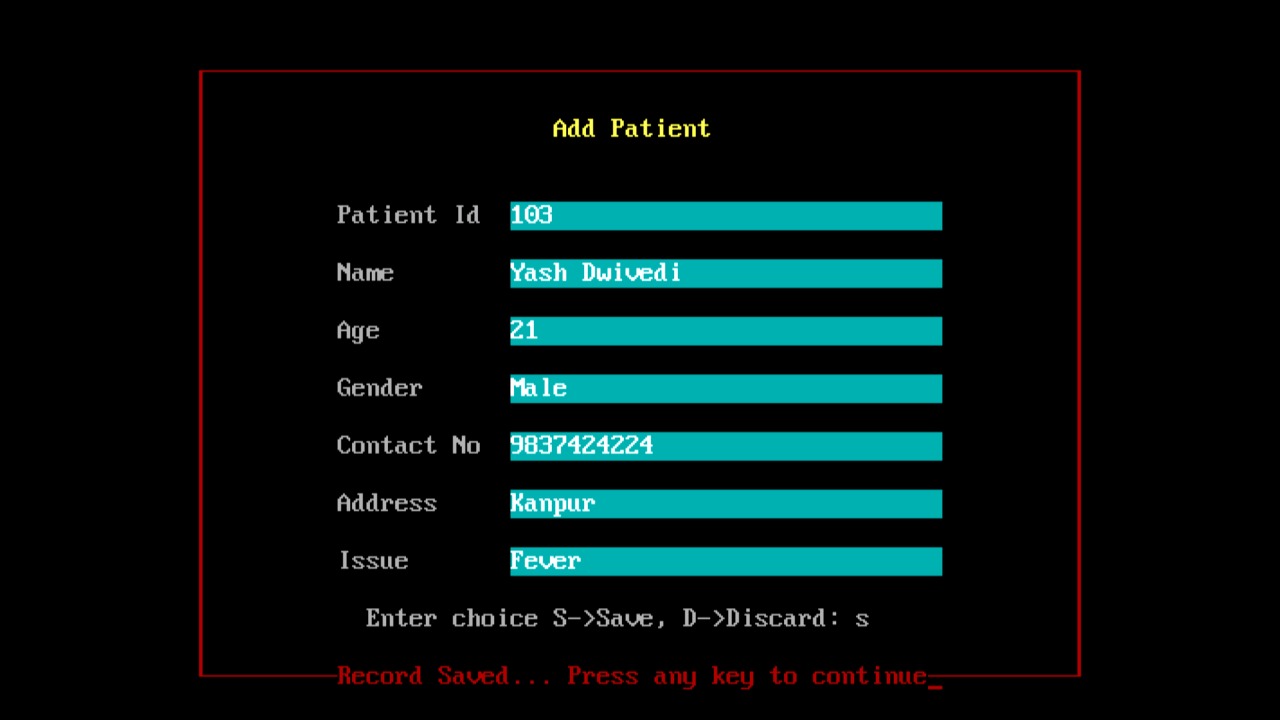
**Screen 11**



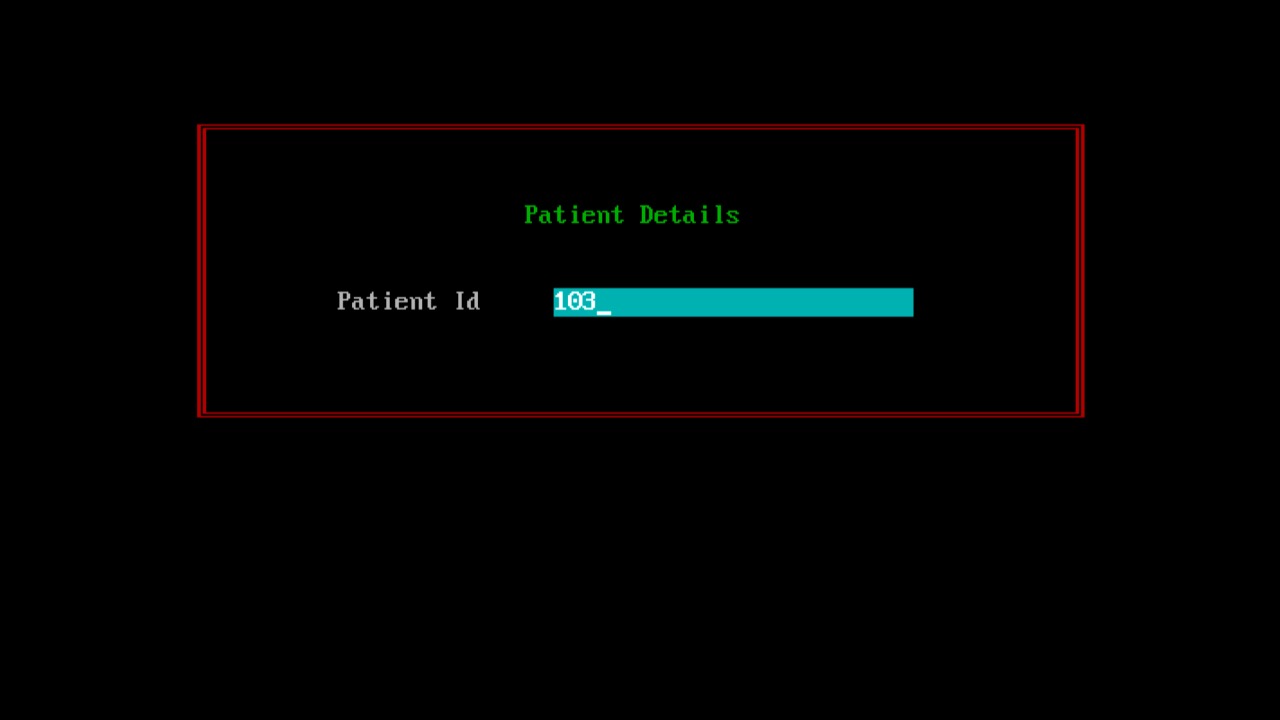
**Screen 111**



**Screen 1111**



**Screen 1113**

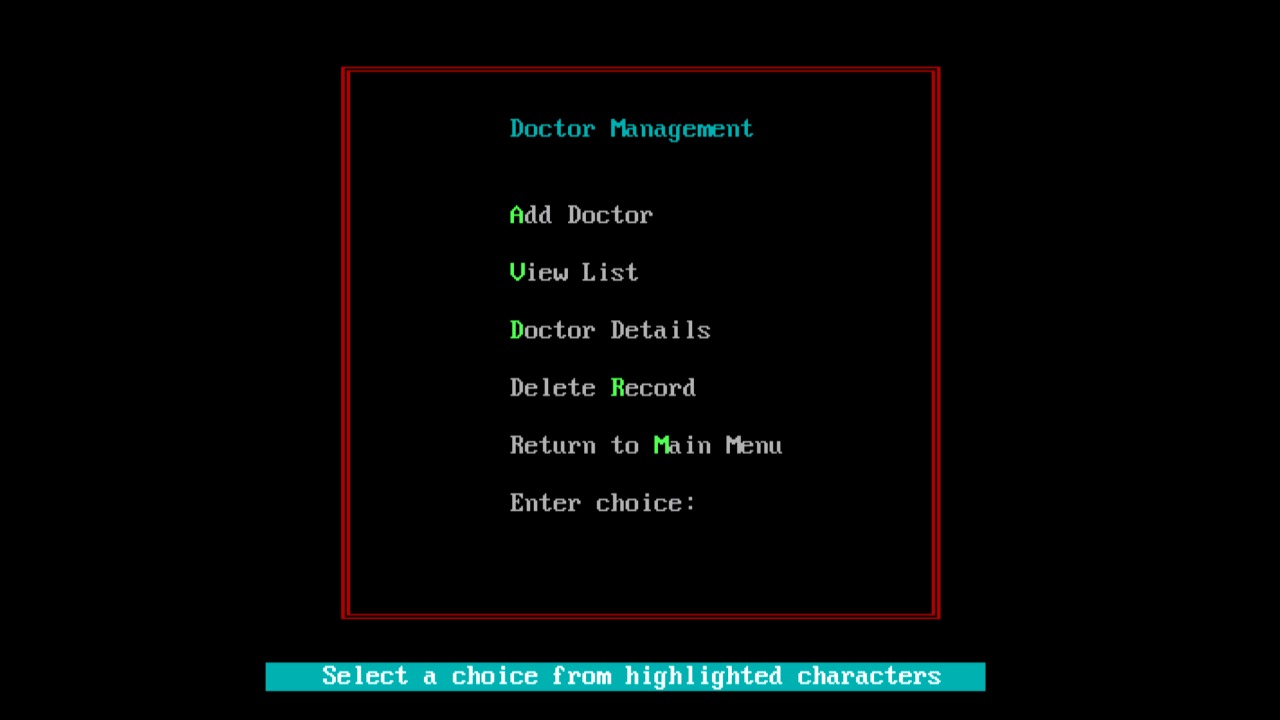




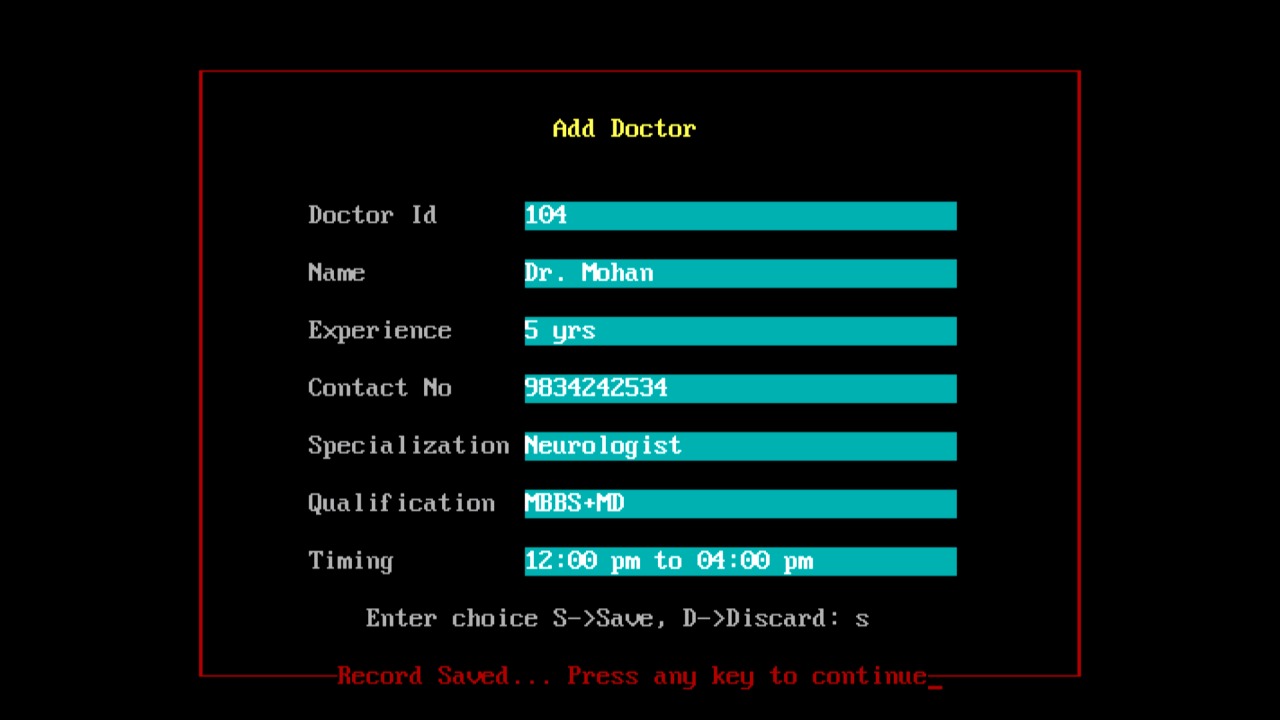
**Screen 1114**



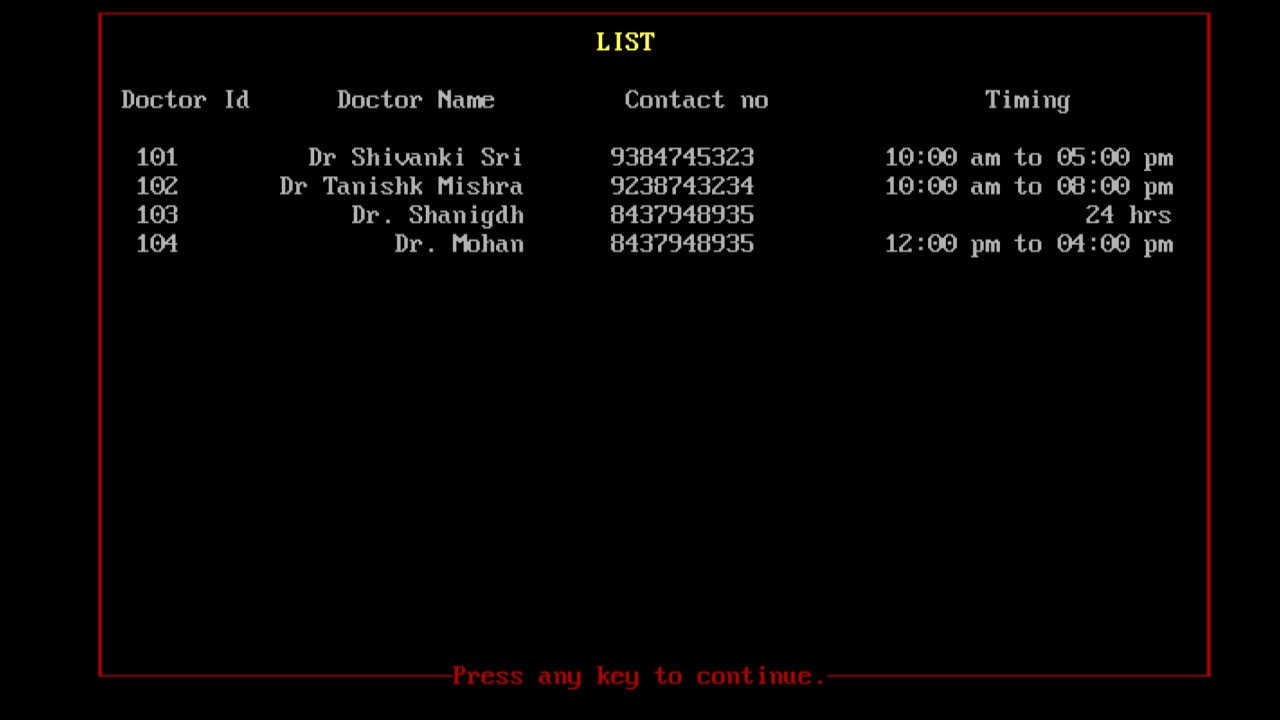
**Screen 112**



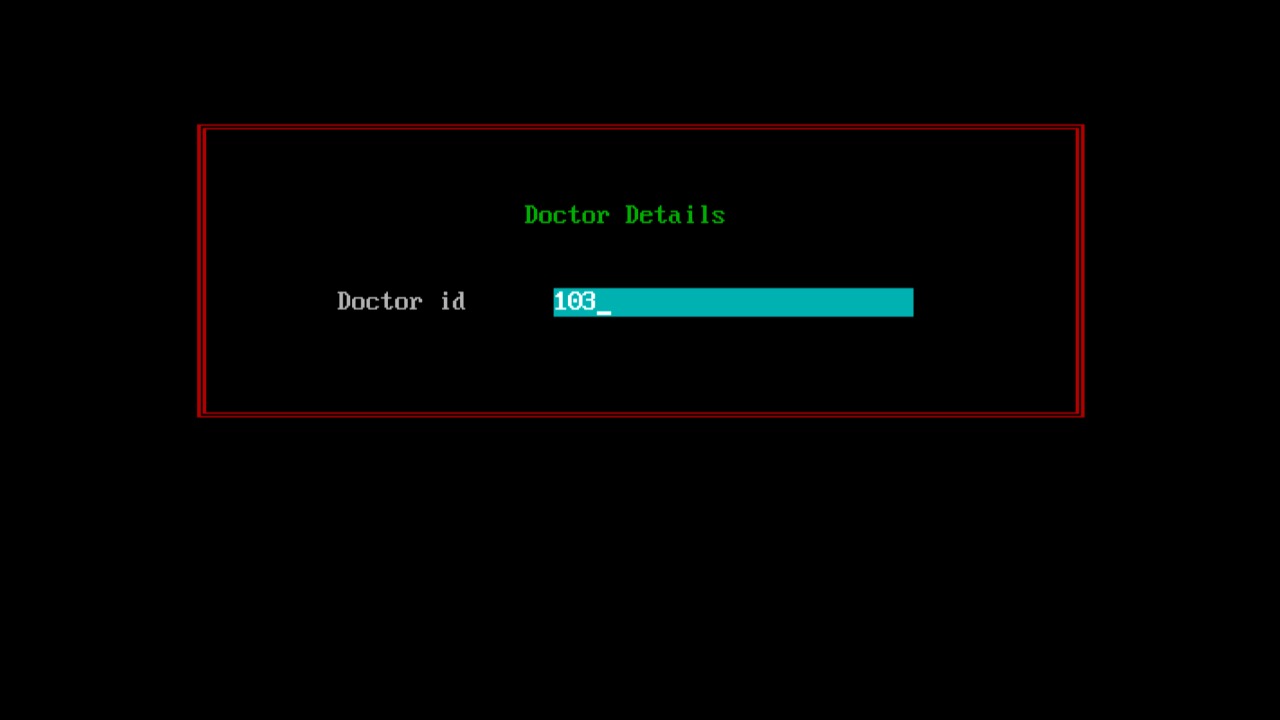
**Screen 1121**



**Screen 1122**



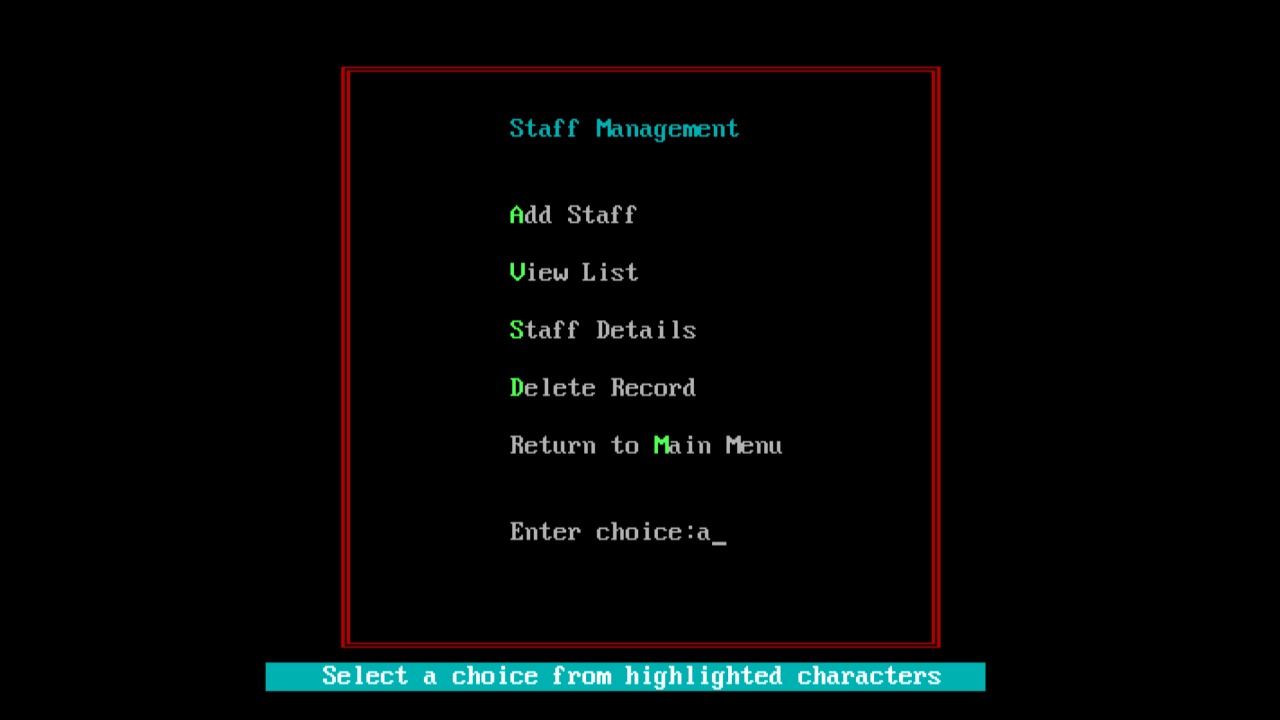
**Screen 1123**



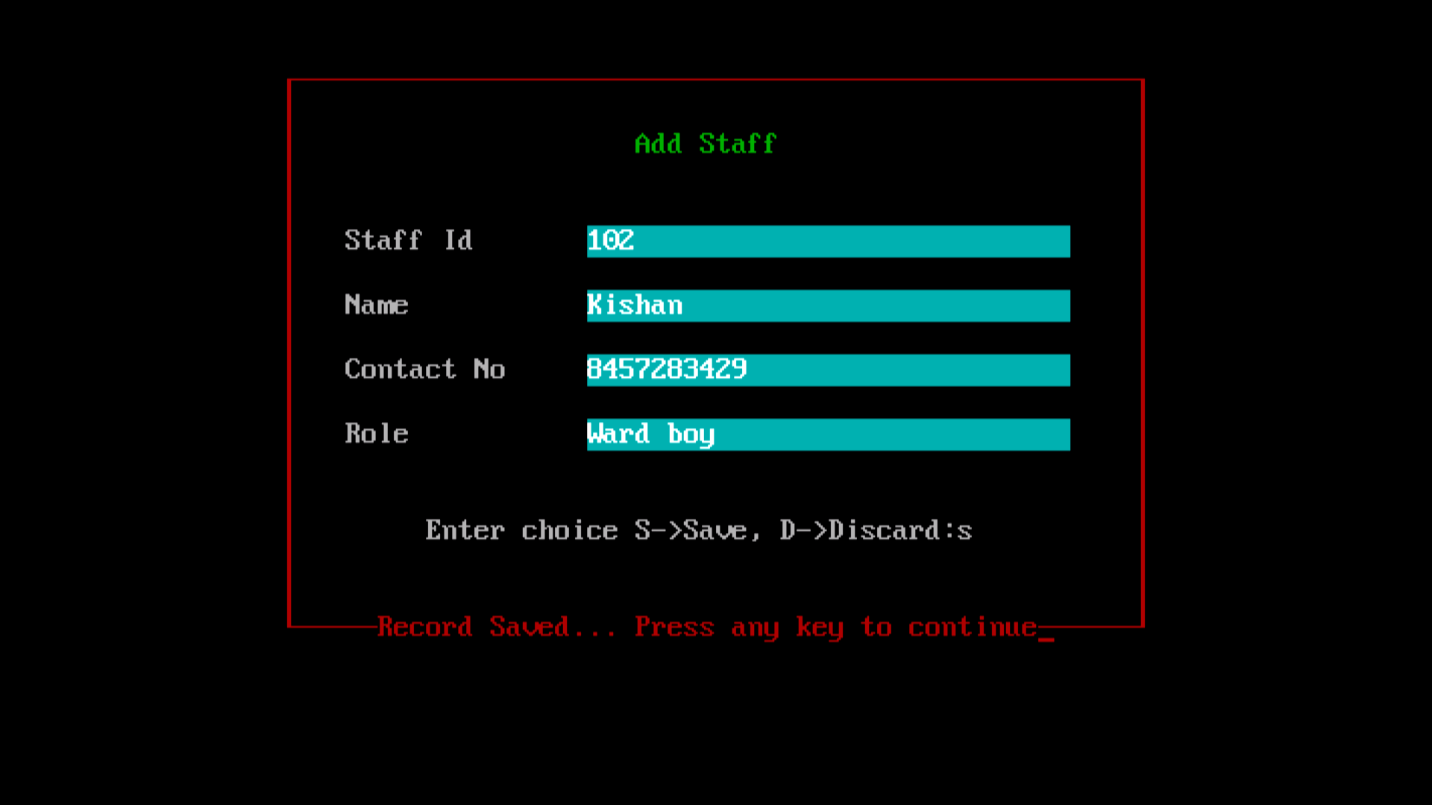
**Screen 1124**



**Screen 113**



**Screen 1131**



**Screen 1132**



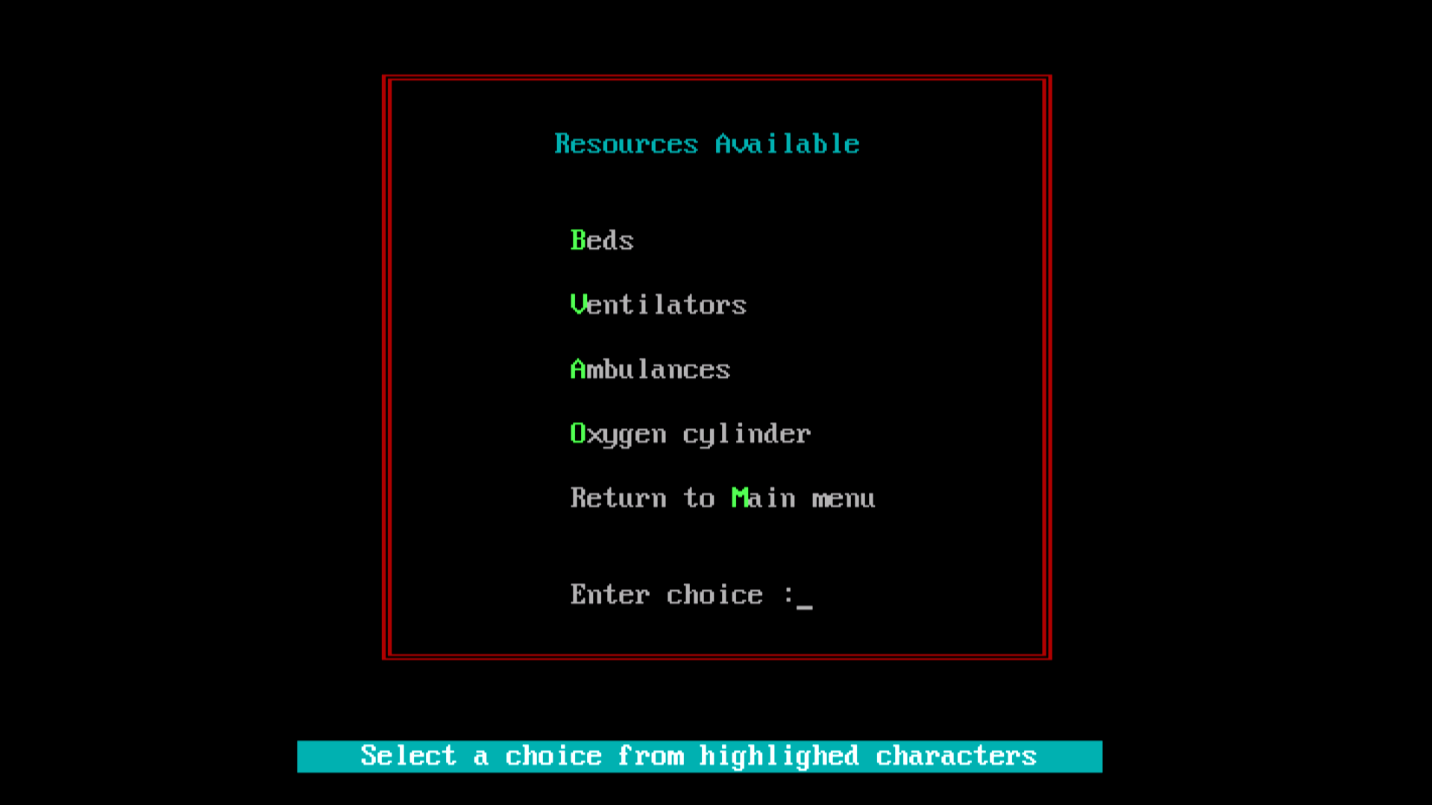
**Screen 1133**



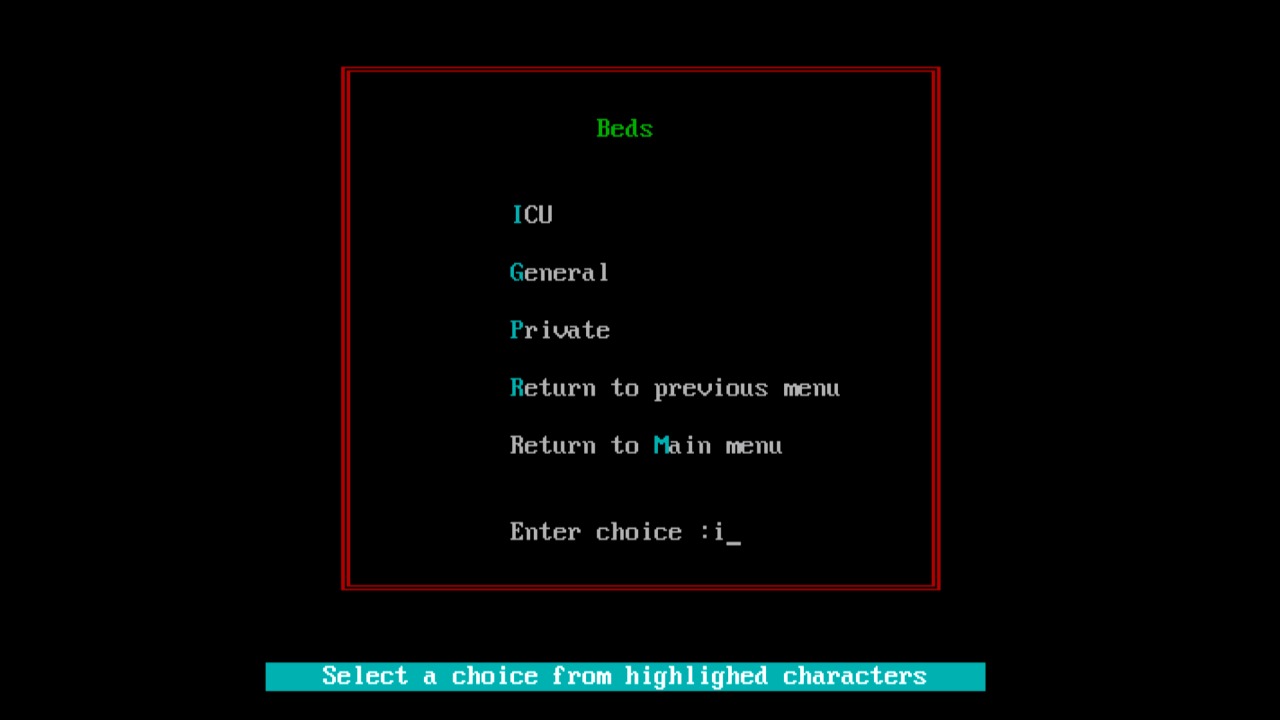
**Screen 1134**



**Screen 114**

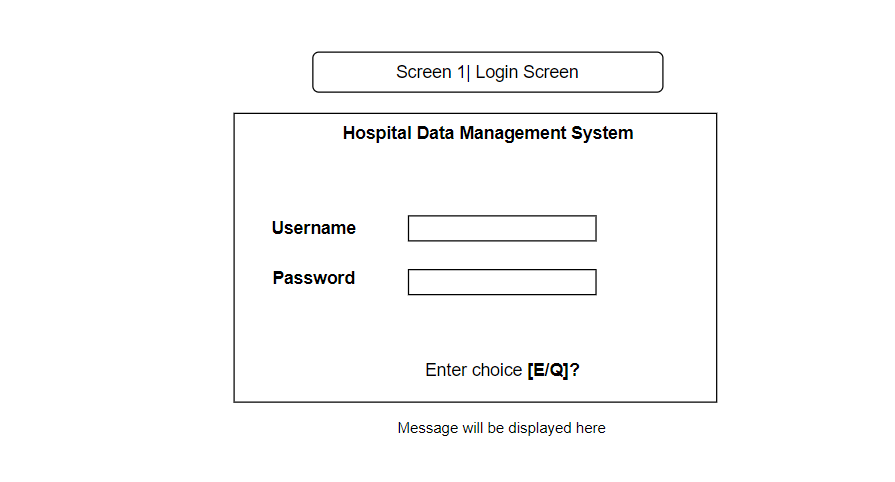


**Screen 1141**



# Validation Rules

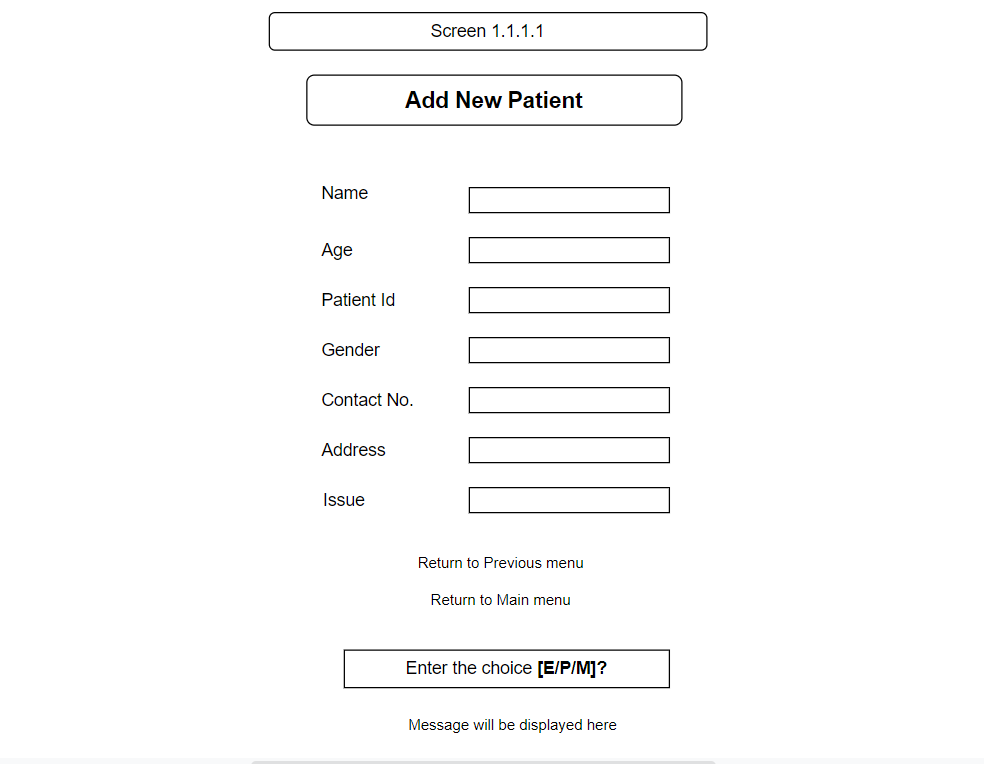
**Screen 1**



**Constraints on Username**

* Cannot be bank
* Min. 4 char long
* Max. 50 char long
* Allowed characters – Alphabets(a-z,A-Z), Space , Dot(.)

**Screen1111**



**Constraints on Patient Name**

* Cannot be bank
* Min. 4 char long
* Max. 50 char long
* Allowed characters – Alphabets(a-z,A-Z), Space , Dot(.)

**Constraints on Age**

* Cannot be blank
* Must be a positive integer

**Constraints on Patient Id**

* Auto generated and incremented

**Constraints on Gender**

* Cannot be blank
* Must be alphabets
* Can be any of the three choices
  + Male
  + Female
  + Others

**Constraints on Contact no.**

* Cannot be blank
* Fixed length of 10
* Must be unique.
* All characters Must be digit only

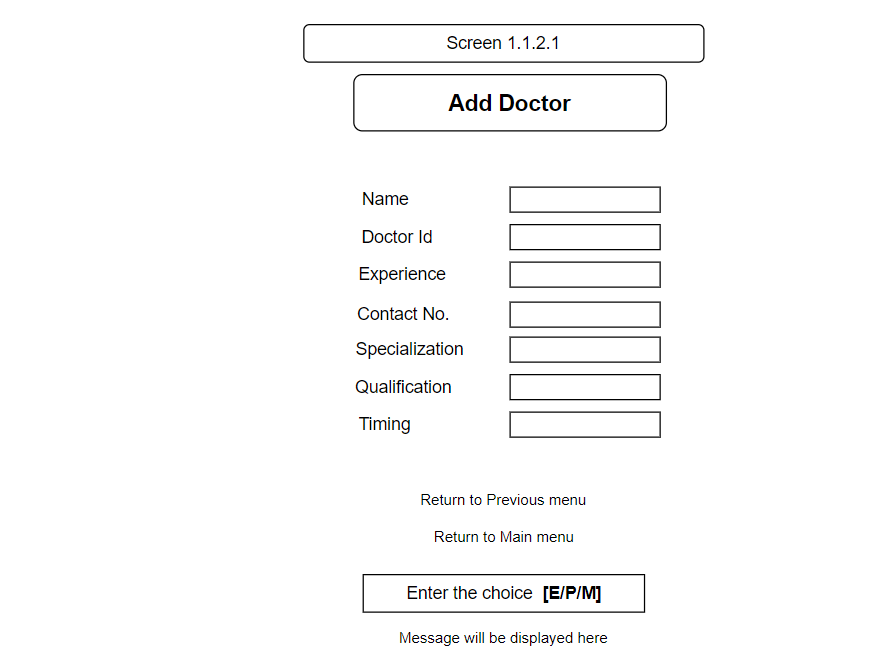
**Constraints on Address**

* Cannot be blank
* Min. 10 char long
* Max. 70 char long
* Can contain special characters, digits and alphabets

**Constraints on Issue**

* Cannot be blank
* Min. 4 char long
* Max. 50 char long
* Allowed alphabets only

**Screen1121**



**Constraints on Doctor Name**

* Cannot be bank
* Min. 4 char long
* Max. 50 char long
* Allowed characters – Alphabets(a-z,A-Z), Space , Dot(.)

**Constraints on Contact no.**

* Cannot be blank
* Fixed length of 10
* Must be unique.
* All characters Must be digit only

**Constraints on Doctor Id**

* Auto generated and incremented

**Constraints on Experience**

* Min 4 char long
* Max 10 char long
* Can be digits and alphabets

**Constraints on Specialization**

* Cannot be blank
* Min. 10 char long
* Max. 70 char long

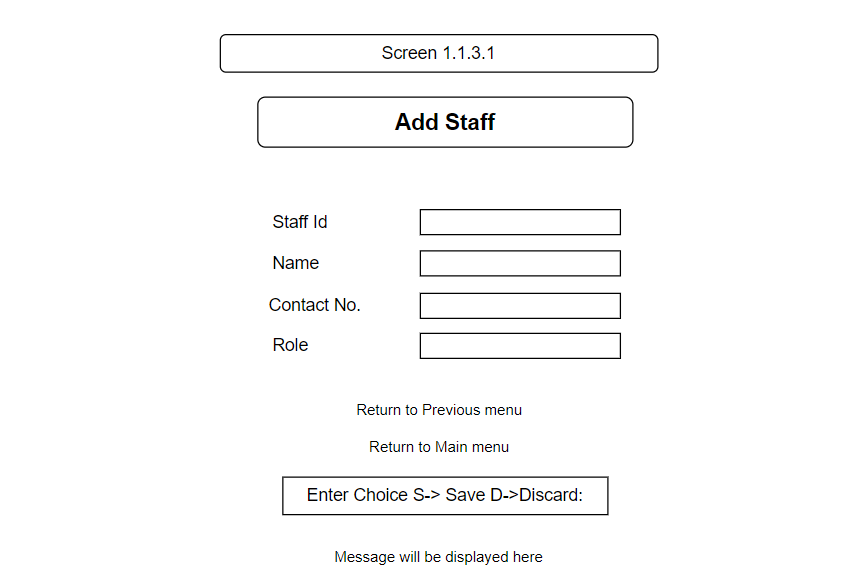
**Constraints on Qualification**

* Cannot be blank
* Min. 4 char long
* Max. 50 char long

**Constraints on Timing**

* Must be string
* Fixed length of 13 ( eg: 10:00 – 21:00)
* Can contain special characters(-,: ) and numeric values

**Screen1131**



**Constraints on Staff Name**

* Cannot be bank
* Min. 4 char long
* Max. 50 char long
* Allowed characters – Alphabets(a-z,A-Z), Space , Dot(.)

**Constraints on Contact no.**

* Cannot be blank
* Fixed length of 10
* Must be unique.
* All characters Must be digit only

**Constraints on Staff Role**

* Cannot be blank
* Min. 5 char long
* Max. 30 char long

**Constraints on Staff Id**

* Auto generated and incremented

# Program Code

// combine.c

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#include<process.h>

struct Userdata{

int total\_users;

char username[31];

char password[11];

};

FILE \*fp;

struct Userdata u;

struct Patient{

char name[51],mobile[11],gender[51],address[101],issue[101];

int age, patientid;

};

struct Patient p;

FILE \*fp;

struct Doctor{

char name[51],mobile[11],specialization[51],qualification[101],timing[51],experience[11];

int doctorid;

};

struct Doctor d;

FILE \*fp;

struct Staff{

char name[51],role[51];mobile[11];

int staffid;

};

struct Staff s;

FILE \*fp;

void SignIn();

/\* login screen \*/

void SignUp\_SignIn();

void SignUp();

void screen11();/\*Main Menu\*/

void screen111();//Patient Management

void screen112();//Doctor Management

void screen113();//Staff Management

void screen114();//Resources Available

void screen1111();//Add Patient

void screen1112();//List

void screen1113();//Patient Details

void screen1114();//Delete Record

void screen1121();//Add Doctor

void screen1122();//list

void screen1123();//Doctor Details

void screen1124();//Delete Record

void screen1131();//Add Staff

void screen1132();//List

void screen1133();//Staff Details

void screen1134();//Delete Record

void screen1141();

void title(char \*title, int row, int color);

void message(char \*message);

void error(char \*message);

void box(int size);

void outline(int c1, int r1, int c2, int r2);

void singleline(int c1, int r1, int c2, int r2);

int isValidName(char \*name){

int valid=1;

int i=0;

int length=strlen(name);

if(!(length>=4 && length<=50)){

valid=0;

return valid;

}

while(name[i]){

if(!(isalpha(name[i]) || name[i]=='.' || name[i]==' '))

{

valid=0;

return valid;

}

i++;

}

return valid;

}

int isValidAge(char \*age){

int i=0;

if(atoi(age)==0)

{

return 0;

}

while(age[i]){

if(!isdigit(age[i])){

return 0;

}

i++;

}

return 1;

}

int isValidMobile(char \*number){

int i=0;

while(number[i]){

if((!isdigit(number[i]))){

return 0;

}

i++ ;

}

return 1;

}

int isValidRole(char \*role){

int valid=1;

int i=0;

int length=strlen(role);

if(!(length>=4 && length<=50)){

valid=0;

return valid;

}

while(role[i]){

if(!(isalpha(role[i]) || role[i]==' '))

{

valid=0;

return valid;

}

i++;

}

return valid;

}

int nextUserID(){

int counter=0;

rewind(fp);

while(fread(&u,sizeof(u),1,fp))

counter++;

return (counter+1);

}

int nextPatientID(){

int counter=100;

rewind(fp);

while(fread(&p,sizeof(p),1,fp)){

counter++;

}

return (counter+1);

}

int nextDoctorID(){

int counter=100;

rewind(fp);

while(fread(&d,sizeof(d),1,fp)){

counter++;

}

return (counter+1);

}

int nextStaffID(){

int counter=100;

rewind(fp);

while(fread(&s,sizeof(s),1,fp)){

counter++;

}

return (counter+1);

}

void main(){

SignUp\_SignIn();

//SignUp();

//SignIn();

//getch();

}

void SignUp(){

char username[31],password[11],re\_password[11],ch,re\_ch;

int i=0,already\_exist=0;

clrscr();

outline(13,2,54,17);

//fp=fopen("demo.txt","a+");

fp=fopen("demo.dat","rb+");

if(fp==NULL){

fp=fopen("demo.dat","wb+");

}

title("Hospital Data Management System",5,YELLOW);

textcolor(YELLOW);

gotoxy(32,7);cprintf("User Authentication");

textcolor(WHITE);

gotoxy(25,10);printf("Username");gotoxy(44,10);box(15);

gotoxy(25,12);printf("Password");gotoxy(44,12);box(15);

gotoxy(25,14);printf("Re-Enter Password");gotoxy(44,14);box(15);

outline(15,5,53,14);

//do while{

gotoxy(44,10);box(15);

gotoxy(44,10);gets(username);

rewind(fp);

while(fread(&u,sizeof(u),1,fp)){

if(strcmp(u.username,username)==0){

already\_exist=1;

break;

}

else

already\_exist=0;

}

if(already\_exist==1){

gotoxy(30,20);printf("Username Already Exists...");

sleep(.5);

}

while(already\_exist==1);{

fflush(stdin);

gotoxy(44,12);

while(i<10){

ch=getch();

if(ch==13) break;

else if(ch==8){

if(i>0)

{

int n;

i--;

gotoxy(44,12);clreol();

for(n=0;n<i;n++) putchar('\*');

}

}

else{

password[i]=ch;

putchar('\*');

i++;

}

}

password[i]='\0';

i=0;

gotoxy(44,14);

fflush(stdin);

while(i<10){

re\_ch=getch();

if(re\_ch==13) break;

else if(re\_ch==8){

if(i>0)

{

int n;

i--;

gotoxy(44,14);clreol();

for(n=0;n<i;n++) putchar('\*');

}

}

else{

re\_password[i]=re\_ch;

putchar('\*');

i++;

}

}

re\_password[i]='\0';

gotoxy(44,12);clreol();box(15);

gotoxy(44,14);clreol();box(15);

}

while(strcmp(password,re\_password)!=0);

u.total\_users=nextUserID();

fflush(stdin);

strcpy(u.username,username);

fflush(stdin);

strcpy(u.password,password);

fseek(fp,0,SEEK\_END);

fwrite(&u,sizeof(struct Userdata),1,fp);

fclose(fp);

SignIn();

}

void SignIn(){

char username[31],password[11],ch;

int i=0,key=0;

clrscr();

title("Hospital Data Management System",5,YELLOW);

textcolor(WHITE);

gotoxy(32,7);printf("User Authentication");

gotoxy(25,10);printf("Username");gotoxy(37,10);box(15);

gotoxy(25,12);printf("Password");gotoxy(37,12);box(15);

outline(13,2,54,15);

gotoxy(37,10);gets(username);

fflush(stdin);

gotoxy(37,12);

while(i<10){

ch=getch();

if(ch==13) break;

else if(ch==8){

if(i>0)

{

int n;

i--;

gotoxy(37,12);clreol();

for(n=0;n<i;n++) putchar('\*');

}

}

else{

password[i]=ch;

putchar('\*');

i++;

}

}

password[i]='\0';

fp=fopen("demo.dat","rb+");

//rewind(fp);

while(fread(&u,sizeof(u),1,fp)){

if(strcmp(u.username,username)==0 && strcmp(u.password,password)==0){

textcolor(GREEN);gotoxy(37,16);cprintf("Success");textcolor(WHITE);

key=1;

screen11();

break;

}

}

fclose(fp);

if(key==0)

{

gotoxy(30,20);message("Invalid Username or Password!!");

sleep(1);

SignIn();

}

}

void SignUp\_SignIn(){

int i;char key[12]="SecurityKey",key1[12];

char choice[5];

clrscr();

outline(10,2,64,22);

gotoxy(15,5);

textbackground(LIGHTBLUE);

textcolor(WHITE);

cprintf(" A healthy outside starts form the inside ");

textbackground(9);

gotoxy(23,8);

cprintf("----------------------------------------");

gotoxy(23,20);

cprintf("----------------------------------------");

for(i=9;i<20;i++)

{

gotoxy(23,i);

cprintf("|");

gotoxy(62,i);

cprintf("|");

}

gotoxy(27,9);

textbackground(3);

cprintf("Hospital Data Management System");gotoxy(0,9);box(5);gotoxy(23,9);box(4);

fflush(stdin);

textbackground(9);

gotoxy(23,10);

cprintf("----------------------------------------");

gotoxy(24,12);

printf(" LOGIN / SIGN UP MENU:");

gotoxy(24,14);printf(" If already a user then sign in (I) ");

gotoxy(24,15);printf(" else sign up (U)");

gotoxy(24,16);printf(" To quit (Q)");

textbackground(10);

gotoxy(26,18);cprintf("SIGN IN");

gotoxy(53,18);cprintf("SIGN UP");

textbackground(WHITE);

do

{

gotoxy(30,22);printf("Enter choice as (U/I/Q): ");gets(choice);}

while(strcmp(choice,"u")!=0&&strcmp(choice,"i")!=0&&strcmp(choice,"q")!=0&&strcmp(choice,"U")!=0&&strcmp(choice,"I")!=0&&strcmp(choice,"Q")!=0);

fflush(stdin);

gotoxy(23,24);box(35);

textbackground(BLACK);

if(strcmp(choice,"i")==0||strcmp(choice,"I")==0)

SignIn();

else if(strcmp(choice,"u")==0||strcmp(choice,"U")==0){

textcolor(15);

gotoxy(23,24);printf("Enter Security Key : ");gotoxy(43,24);gets(key1);

if(strcmp(key1,key)==0)

SignUp();

else{

gotoxy(23,25);printf("Invalid Security Key...Please Try Again");

}

}

else if(strcmp(choice,"q")==0||strcmp(choice,"Q")==0)

exit(0);

}

void screen11(){

char choice;

clrscr();

for(;;){

title("Main Menu",5,YELLOW);

textcolor(GREEN);

gotoxy(32,8);cprintf("P");printf("atient Management");

gotoxy(32,10);cprintf("D");printf("octor Management");

gotoxy(32,12);cprintf("S");printf("taff Management");

gotoxy(32,14);cprintf("R");printf("esources Available");

gotoxy(32,16);cprintf("Q");printf("uit");

textcolor(WHITE);

gotoxy(32,20);printf("Enter choice: ");

outline(20,3,40,18);

message("Select a choice from highlighted characters");

gotoxy(46,20);fflush(stdin);

choice=getchar();

switch(choice){

case 'p': case 'P':

screen111();break;

case 'd': case 'D':

screen112();break;

case 's': case 'S':

screen113();break;

case 'r': case 'R':

screen114();break;

case 'q': case 'Q':

exit(0);

default:

textcolor(RED+BLINK);

error("Invalid Choice...");

textcolor(WHITE);

}

}

}

void screen111(){

char choice;

clrscr();

fp=fopen("patient.dat","rb+");

if(fp==NULL){

fp=fopen("patient.dat","wb+");

}

for(;;){

title ("Patient Management", 5,CYAN);

textcolor(LIGHTGREEN);

gotoxy(31,8);cprintf("A");printf("dd Patient");

gotoxy(31,10);cprintf("V");printf("iew List");

gotoxy(31,12);cprintf("P");printf("atient Details");

gotoxy(31,14);cprintf("D");printf("elete Record");

gotoxy(31,16);printf("Return to ");cprintf("M");printf("ain Menu");

textcolor(WHITE);

gotoxy(31,19);printf("Enter choice : ");

outline(20,3,38,19);

message("Select a choice from highlighted characters");

gotoxy(45,19);

choice=getchar();

switch(choice){

case 'a': case 'A':

screen1111();break;

case'v': case 'V':

screen1112();break;

case 'p': case 'P':

screen1113();break;

case 'd': case 'D':

screen1114();break;

case 'm': case 'M':

screen11();break;

default:

textcolor(RED+BLINK);

error("Invalid Choice...");

textcolor(WHITE);

fclose(fp);

}

}

}

void screen1111(){

char choice;

system("cls");

p.patientid=nextPatientID();

title("Add Patient",5,YELLOW);

textcolor(WHITE);

gotoxy(20,8);printf("Patient Id");gotoxy(32,8);box(30);

gotoxy(20,10);printf("Name");gotoxy(32,10);box(30);

gotoxy(20,12);printf("Age");gotoxy(32,12);box(30);

gotoxy(20,14);printf("Gender");gotoxy(32,14);box(30);

gotoxy(20,16);printf("Contact No");gotoxy(32,16);box(30);

gotoxy(20,18);printf("Address");gotoxy(32,18);box(30);

gotoxy(20,20);printf("Issue");gotoxy(32,20);box(30);

singleline(10,3,60,20);

gotoxy(32,8);fflush(stdin);printf("%d", p.patientid);

gotoxy(32,10);fflush(stdin);

for(;;){

gotoxy(32,10);

fflush(stdin);gets(p.name);

if(isValidName(p.name))

break;

else

{

gotoxy(30,24);

printf("Sorry! Invalid Name...");

}

}

gotoxy(32,12);fflush(stdin);

for(;;){

char age[4];

//gotoxy(32,12);box(30);

gotoxy(32,12);

fflush(stdin);scanf("%s",age);

if(isValidAge(age)){

p.age=atoi(age);

break;

}

else {

gotoxy(30,24);

printf("Sorry! Invalid Age...");

}

}

gotoxy(32,14);

for(;;){

gotoxy(32,14);box(30);

gotoxy(32,14);

fflush(stdin);scanf("%s",p.gender);

if(strcmpi(p.gender,"male")==0 || strcmpi(p.gender,"female")==0 || strcmpi(p.gender,"others")==0){

break;

}

else{

gotoxy(30,24);

printf("Sorry! Invalid Gender..."); }

}

gotoxy(32,16);fflush(stdin);

for(;;){

char number[51];

gotoxy(32,16);box(30);

gotoxy(32,16);

fflush(stdin);

scanf("%s",&number);

if((isValidMobile(number))&&(strlen(number)==10)){

strcpy(p.mobile,number);

break;

}

else{

gotoxy(28,24);

printf("Only 10 digits allowed!!");

}

}

gotoxy(32,18);fflush(stdin);gets(p.address);

gotoxy(32,20);fflush(stdin);gets(p.issue);

gotoxy(22,22);printf("Enter choice S->Save, D->Discard: ");

textcolor(WHITE);

gotoxy(56,22);

fflush(stdin);

choice=getchar();

switch(choice){

case 's': case 'S':

fseek(fp,0,SEEK\_END);

fwrite(&p,sizeof(p),1,fp);

gotoxy(20,24);printf("Record Saved... Press any key to continue");

getch();

case 'd': case 'D':

clrscr();

return;

}

}

void screen1112(){

system("cls");

title("LIST",3,YELLOW);

textcolor(WHITE);

rewind(fp);

gotoxy(5,5);printf("%-10s %20s %15s %20s\n\n","Patient Id","Patient Name","Age","Contact No");

while(fread(&p,sizeof(p),1,fp)){

printf("%8d %25s %15d %22s\n",p.patientid,p.name,p.age,p.mobile);

}

singleline(3,2,76,22);

gotoxy(30,24);system("pause");

}

void screen1113(){

int p1, found=0;

clrscr();

title("Patient Details", 8, GREEN);

textcolor(WHITE);

gotoxy(20,11);printf("Patient Id");gotoxy(35,11);box(25);

outline(10,5,60,9);

gotoxy(35,11);fflush(stdin);scanf("%d", &p1);

rewind(fp);

while(fread(&p,sizeof(p),1,fp)){

if(p.patientid==p1){

found=1;

break;

}

}

if(found){

system("cls");

title("Patient Info",5,YELLOW);

textcolor(WHITE);

gotoxy(28,8);printf("%-30s","Patient Id:");

gotoxy(28,10);printf("%-10s","Patient Name:");

gotoxy(28,12);printf("%s","Age:");

gotoxy(28,14);printf("%-20s","Gender:");

gotoxy(28,16);printf("%-10s","Mobile:");

gotoxy(28,18);printf("%-30s","Address:");

gotoxy(28,20);printf("%-50s","Issue:");

singleline(10,3,60,20);

gotoxy(45,8);printf("%d",p.patientid);

gotoxy(45,10);printf("%s",p.name);

gotoxy(45,12);printf("%d", p.age);

gotoxy(45,14);printf("%s",p.gender);

gotoxy(45,16);printf("%s",p.mobile);

gotoxy(45,18);printf("%s",p.address);

gotoxy(45,20);printf("%s",p.issue);

gotoxy(28,24);system("pause");

}

else{

clrscr();

textcolor(RED+BLINK);

error("Record not found..");

textcolor(WHITE);

}

}

void screen1114(){

int p1, found=0;

system("cls");

title("Delete Record",8, GREEN);

textcolor(WHITE);

gotoxy(20,11);printf("Patient Id");gotoxy(35,11);box(25);

outline(10,5,60,9);

gotoxy(35,11);fflush(stdin);scanf("%d",&p1);

rewind(fp);

while(fread(&p,sizeof(p),1,fp)){

if(p.patientid==p1){

found=1;

break;

}

}

if(found){

FILE \*temp=fopen("temp.dat","wb");

rewind(fp);

while(fread(&p,sizeof(p),1,fp)){

if(p.patientid!=p1){

fwrite(&p,sizeof(p),1,temp);

}

}

fclose(fp);

fclose(temp);

remove("patient.dat");

rename("temp.dat","patient.dat");

fp=fopen("patient.dat","rb+");

gotoxy(22,15);printf("Record Deleted..");system("pause");

}

else{

clrscr();

textcolor(RED+BLINK);

error("Record not found..");

textcolor(WHITE);

}

}

void screen112(){

char choice;

system("cls");

fp=fopen("doctor.dat","rb+");

if(fp==NULL){

fp=fopen("doctor.dat","wb+");

}

for(;;){

title ("Doctor Management", 5,CYAN);

textcolor(LIGHTGREEN);

gotoxy(32,8);cprintf("A");printf("dd Doctor");

gotoxy(32,10);cprintf("V");printf("iew List");

gotoxy(32,12);cprintf("D");printf("octor Details");

gotoxy(32,14);printf("Delete ");cprintf("R");printf("ecord");

gotoxy(32,16);printf("Return to ");cprintf("M");printf("ain Menu");

textcolor(WHITE);

gotoxy(32,18);printf("Enter choice: ");

outline(20,3,40,18);

message("Select a choice from highlighted characters");

gotoxy(45,18);

choice=getchar();

switch(choice){

case 'a': case 'A':

screen1121();break;

case 'v': case 'V':

screen1122();break;

case 'd': case 'D':

screen1123();break;

case 'r': case 'R':

screen1124();break;

case 'm': case 'M':

screen11();break;

default:

textcolor(RED+BLINK);

error("Invalid Choice...");

textcolor(WHITE);

fclose(fp);

}

}

}

void screen1121(){

char choice;

system("cls");

d.doctorid=nextDoctorID();

title("Add Doctor",5,YELLOW);

textcolor(WHITE);

gotoxy(18,8);printf("Doctor Id");gotoxy(33,8);box(30);

gotoxy(18,10);printf("Name");gotoxy(33,10);box(30);

gotoxy(18,12);printf("Experience");gotoxy(33,12);box(30);

gotoxy(18,14);printf("Contact No");gotoxy(33,14);box(30);

gotoxy(18,16);printf("Specialization");gotoxy(33,16);box(30);

gotoxy(18,18);printf("Qualification");gotoxy(33,18);box(30);

gotoxy(18,20);printf("Timing");gotoxy(33,20);box(30);

singleline(10,3,60,20);

gotoxy(33,8);fflush(stdin);printf("%d",d.doctorid);

gotoxy(33,10);fflush(stdin);

for(;;){

gotoxy(33,10);box(30);

gotoxy(33,10);

fflush(stdin);gets(d.name);

if(isValidName(d.name))

break;

else

{

gotoxy(30,24);

printf("Sorry! Invalid Name...");

}

}

gotoxy(33,12);fflush(stdin);gets(d.experience);

gotoxy(33,14);

for(;;){

char number[51];

gotoxy(33,14);box(30);

gotoxy(33,14);

fflush(stdin);

scanf("%s",&number);

if((isValidMobile(number))&&(strlen(number)==10)){

strcpy(d.mobile,number);

break;

}

else{gotoxy(28,24);

printf("Only 10 digits allowed!!");

}

}

gotoxy(33,16);fflush(stdin);gets(d.specialization);

gotoxy(33,18);fflush(stdin);gets(d.qualification);

gotoxy(33,20);fflush(stdin);gets(d.timing);

gotoxy(22,22);printf("Enter choice S->Save, D->Discard: ");

textcolor(WHITE);

gotoxy(56,22);

choice=getchar();

switch(choice){

case 's': case 'S':

fseek(fp,0,SEEK\_END);

fwrite(&d,sizeof(d),1,fp);

gotoxy(20,24);printf("Record Saved... Press any key to continue");

getch();

case 'd': case 'D':

clrscr();

return;

}

}

void screen1122(){

system("cls");

title("LIST",3,YELLOW);

textcolor(WHITE);

// fseek(fp,0,SEEK\_SET);

rewind(fp);

gotoxy(5,5);

printf("%-10s %15s %18s %20s\n\n","Doctor Id","Doctor Name","Contact no","Timing");

while(fread(&d,sizeof(d),1,fp)){

//gotoxy(5,7);

printf("%8d %23s %15s %28s\n",d.doctorid,d.name,d.mobile,d.timing);

}

singleline(3,2,76,22);

gotoxy(28,24);system("pause");

}

void screen1123(){

int d1,found=0;

system("cls");

title("Doctor Details", 8,GREEN);

textcolor(WHITE);

gotoxy(20,11);printf("Doctor id");gotoxy(35,11);box(25);gotoxy(35,11);

outline(10,5,60,9);

gotoxy(35,11);scanf("%d",&d1);

rewind(fp);

while(fread(&d,sizeof(d),1,fp)){

if(d.doctorid==d1){

found=1;

break;

}

}

if(found){

system("cls");

title("Doctor Info",5,YELLOW);

textcolor(WHITE);

gotoxy(28,8);printf("%-30s ","Doctor Id:");

gotoxy(28,10);printf("%-10s","Doctor name:");

gotoxy(28,12);printf("%10s","Experience:");

gotoxy(28,14);printf("%-20s","Contact No:");

gotoxy(28,16);printf("%-10s","Specialization:");

gotoxy(28,18);printf("%-30s","Qualification:");

gotoxy(28,20);printf("%-50s","Timing:");

singleline(10,3,60,20);

gotoxy(45,8);printf("%d ",d.doctorid);

gotoxy(45,10);printf("%s",d.name);

gotoxy(45,12);printf("%s", d.experience);

gotoxy(45,14);printf("%s",d.mobile);

gotoxy(45,16);printf("%s",d.specialization);

gotoxy(45,18);printf("%s",d.qualification);

gotoxy(45,20);printf("%s",d.timing);

gotoxy(28,24);system("pause");

}

else{

clrscr();

textcolor(RED+BLINK);

error("Record not found..");

textcolor(WHITE);

}

}

void screen1124(){

int d1, found=0;

system("cls");

title("Delete Record",8, GREEN);

textcolor(WHITE);

gotoxy(20,11);printf("Doctor Id");gotoxy(35,11);box(25);

outline(10,5,60,9);

gotoxy(35,11);fflush(stdin);scanf("%d",&d1);

rewind(fp);

while(fread(&d,sizeof(d),1,fp)){

if(d.doctorid==d1){

found=1;

break;

}

}

if(found){

FILE \*temp=fopen("temp.dat","wb");

rewind(fp);

while(fread(&d,sizeof(d),1,fp)){

if(d.doctorid!=d1){

fwrite(&d,sizeof(d),1,temp);

}

}

fclose(fp);

fclose(temp);

remove("doctor.dat");

rename("temp.dat","doctor.dat");

fp=fopen("doctor.dat","rb+");

gotoxy(22,15);printf("Record Deleted..");system("pause");

}

else{

clrscr();

textcolor(RED+BLINK);

error("Record not found..");

textcolor(WHITE);

}

}

void screen113(){

char choice;

system("cls");

fp=fopen("staff.dat","rb+");

if(fp==NULL){

fp=fopen("staff.dat","wb+");

}

for(;;){

title ("Staff Management", 5, CYAN);

textcolor(LIGHTGREEN);

gotoxy(32,8);cprintf("A");printf("dd Staff");

gotoxy(32,10);cprintf("V");printf("iew List");

gotoxy(32,12);cprintf("S");printf("taff Details");

gotoxy(32,14);cprintf("D");printf("elete Record");

gotoxy(32,16);printf("Return to ");cprintf("M");printf("ain Menu");

textcolor(WHITE);

gotoxy(32,19);printf("Enter choice: ");

message("Select a choice from highlighted characters");

outline(20,3,40,19);

gotoxy(45,19);choice=getchar();

switch(choice){

case 'a': case 'A':

screen1131();break;

case 'v': case 'V':

screen1132();break;

case 's': case 'S':

screen1133();break;

case 'd': case 'D':

screen1134();break;

case 'm': case 'M':

screen11();break;

default:

textcolor(RED+BLINK);

error("Invalid Choice...");

textcolor(WHITE);

fclose(fp);

}

}

}

void screen1131(){

char choice;

system("cls");

s.staffid=nextStaffID();

title("Add Staff",5,GREEN);

textcolor(WHITE);

gotoxy(18,8);printf("Staff Id");gotoxy(33,8);box(30);

gotoxy(18,10);printf("Name");gotoxy(33,10);box(30);

gotoxy(18,12);printf("Contact No");gotoxy(33,12);box(30);

gotoxy(18,14);printf("Role");gotoxy(33,14);box(30);

singleline(14,3,52,16);

gotoxy(33,8);fflush(stdin);printf("%d",s.staffid);

gotoxy(33,10);fflush(stdin);

for(;;){

gotoxy(33,10);box(30);

gotoxy(33,10);

fflush(stdin);gets(s.name);

if(isValidName(s.name))

break;

else

{

gotoxy(30,24);

printf("Sorry! Invalid Name...");

}

}

gotoxy(33,12);

for(;;){

char number[51];

gotoxy(33,12);box(30);

gotoxy(33,12);

fflush(stdin);

scanf("%s",&number);

if((isValidMobile(number))&&(strlen(number)==10)){

strcpy(s.mobile,number);

break;

}

else{gotoxy(28,24);

printf("Only 10 digits allowed!!");

}

}

gotoxy(33,14);fflush(stdin);

for(;;){

gotoxy(33,14);box(30);

gotoxy(33,14);

fflush(stdin);gets(s.role);

if(isValidRole(s.role))

break;

else

{

gotoxy(30,24);

printf("Sorry! Invalid Role...");

}

}

gotoxy(23,17);printf("Enter choice S->Save, D->Discard: ");

textcolor(WHITE);

gotoxy(56,17);

fflush(stdin);

choice=getchar();

switch(choice){

case 's': case 'S':

fseek(fp,0,SEEK\_END); //bottom of file

fwrite(&s,sizeof(s),1,fp);

gotoxy(20,20);printf("Record Saved... Press any key to continue");

getch();

case 'd': case 'D':

clrscr();

return;

}

}

void screen1132(){

system("cls");

title("LIST",3,YELLOW);

textcolor(WHITE);

rewind(fp);

gotoxy(5,5);

printf("%10s %30s %25s \n\n","Staff Id","Staff Name","Contact no");

while(fread(&s,sizeof(s),1,fp)){

//gotoxy(5,7);

printf("%10d %35s %25s\n",s.staffid,s.name,s.mobile);

}

singleline(3,2,72,21);

gotoxy(28,24);system("pause");

}

void screen1133(){

char ch, choice;

int s1, found=0;

system("cls");

title("Staff Details", 8, GREEN);

textcolor(WHITE);

gotoxy(20,11);printf("Staff Id");gotoxy(35,11);box(25);

outline(10,5,60,9);

gotoxy(35,11);fflush(stdin);scanf("%d",&s1);

rewind(fp);

while(fread(&s,sizeof(s),1,fp)){

if(s.staffid==s1){

found=1;

break;

}

}

if(found){

system("cls");

title("Staff Info",6,YELLOW);

textcolor(WHITE);

gotoxy(28,9);printf("%-30s","Staff Id:");

gotoxy(28,11);printf("%-30s","Staff Name:");

gotoxy(28,13);printf("%-30s","Contact No:");

gotoxy(28,15);printf("%-20s","Role:");

singleline(15,3,48,15);

gotoxy(45,9);printf("%d",s.staffid);

gotoxy(45,11);printf("%s",s.name);

gotoxy(45,13);printf("%s",s.mobile);

gotoxy(45,15);printf("%s",s.role);

gotoxy(28,24);system("pause");

}

else{

clrscr();

textcolor(RED+BLINK);

error("Record not found..");

textcolor(WHITE);

}

}

void screen1134(){

int s1, found=0;

system("cls");

title("Delete Record",8, GREEN);

textcolor(WHITE);

gotoxy(20,11);printf("Staff Id");gotoxy(35,11);box(25);

outline(10,5,60,9);

gotoxy(35,11);fflush(stdin);scanf("%d",&s1);

rewind(fp);

while(fread(&s,sizeof(s),1,fp)){

if(s.staffid==s1){

found=1;

break;

}

}

if(found){

FILE \*temp=fopen("temp.dat","wb");

rewind(fp);

while(fread(&s,sizeof(s),1,fp)){

if(s.staffid!=s1){

fwrite(&s,sizeof(s),1,temp);

}

}

fclose(fp);

fclose(temp);

remove("staff.dat");

rename("temp.dat","staff.dat");

fp=fopen("staff.dat","rb+");

gotoxy(22,15);printf("Record Deleted..");system("pause");

}

else{

clrscr();

textcolor(RED+BLINK);

error("Record not found..");

textcolor(WHITE);

}

}

void screen114(){

char choice;

clrscr();

system("cls");

for(;;){

title("Resources Available",5,CYAN);

textcolor(LIGHTGREEN);

gotoxy(32,8);cprintf("B");printf("eds");

gotoxy(32,10);cprintf("V");printf("entilators");

gotoxy(32,12);cprintf("A");printf("mbulances");

gotoxy(32,14);cprintf("O");printf("xygen cylinder");

gotoxy(32,16);printf("Return to ");cprintf("M");printf("ain menu");

textcolor(WHITE);

gotoxy(32,19);printf("Enter choice : ");

outline(20,3,40,17);

message("Select a choice from highlighed characters");

gotoxy(46,19);choice=getchar();

switch(choice){

case 'b': case'B':

screen1141();break;

case 'v': case 'V':

clrscr();

singleline(25,10,28,3);

textcolor(LIGHTCYAN);

gotoxy(27,12);cprintf("Available Ventilators= 100");

sleep(2);textcolor(WHITE);

break;

case 'a': case 'A':

clrscr();

singleline(26,10,26,3);

textcolor(LIGHTCYAN);

gotoxy(28,12);cprintf("Available Ambulances= 10");

sleep(2);textcolor(WHITE);

break;

case 'o': case 'O':

clrscr();

singleline(23,10,33,3);

textcolor(LIGHTCYAN);

gotoxy(25,12);cprintf("Available Oxygen Cylinders= 250");

sleep(2);textcolor(WHITE);

break;

case 'm': case 'M':

screen11();break;

default:

textcolor(RED+BLINK);

error("Invalid Choice...");

textcolor(WHITE);

}

}

}

void screen1141(){

char choice;

system("cls");

for(;;){

title("Beds",5,GREEN);

textcolor(CYAN);

gotoxy(32,8);cprintf("I");printf("CU");

gotoxy(32,10);cprintf("G");printf("eneral");

gotoxy(32,12);cprintf("P");printf("rivate");

gotoxy(32,14);cprintf("R");printf("eturn to previous menu");

gotoxy(32,16);printf("Return to ");cprintf("M");printf("ain menu");

textcolor(WHITE);

gotoxy(32,19);printf("Enter choice : ");

outline(20,3,40,17);

message("Select a choice from highlighed characters");

gotoxy(46,19);choice=getchar();

switch(choice){

case 'i': case'I':

clrscr();

singleline(27,10,24,3);

textcolor(LIGHTCYAN);

gotoxy(29,12);cprintf("Available ICU Beds= 15");

sleep(2);textcolor(WHITE);

break;

case 'g': case 'G':

clrscr();

singleline(25,10,29,3);

textcolor(LIGHTCYAN);

gotoxy(27,12);cprintf("Available General Beds= 100");

sleep(2);textcolor(WHITE);

break;

case 'p': case 'P':

clrscr();

singleline(25,10,28,3);

textcolor(LIGHTCYAN);

gotoxy(27,12);cprintf("Available Private Beds= 30");

sleep(2);textcolor(WHITE);

break;

case 'r': case'R':

screen114();break;

case 'm': case 'M':

screen11();break;

default:

textcolor(RED+BLINK);

error("Invalid Choice...");

textcolor(WHITE);

}

}

}

void title(char \*title, int row, int color){

int col=40-strlen(title)/2;

system("cls");

gotoxy(col,row);

textcolor(color);

cprintf("%s",title);

}

void message(char \*message){

int col=40-strlen(message)/2;

gotoxy(15,24);box(50);

gotoxy(col,24);

printf("%s",message);

textcolor(WHITE);

}

void box(int size){

int i;

textbackground(CYAN);

for(i=1;i<=size;i++)

cprintf(" ");

textbackground(BLACK);

}

void error(char \*message){

int col=27-strlen(message)/2;

system("cls");

gotoxy(15,13);box(50);

gotoxy(col,13);

textcolor(RED+BLINK);

printf("%s",message);

textcolor(WHITE);

system("pause");

clrscr();

}

void outline(int c1, int r1, int c2, int r2){

int i;

textcolor(RED);

gotoxy(c1,r1);

cprintf("%c",201);

for(i=1;i<=c2;i++)

cprintf("%c",205);

cprintf("%c",187);

for(i=1;i<=r2;i++)

{

gotoxy(c1,r1+i);

cprintf("%c",186);

gotoxy(c1+c2+1,r1+i);

cprintf("%c",186);

}

gotoxy(c1,r1+i);

cprintf("%c",200);

for(i=1;i<=c2;i++)

cprintf("%c",205);

cprintf("%c",188);

textcolor(WHITE);

}

void singleline(int c1, int r1, int c2, int r2){

int i;

textcolor(RED);

gotoxy(c1,r1);

cprintf("%c",218);

for(i=1;i<=c2;i++)

cprintf("%c",196);

cprintf("%c",191);

for(i=1;i<=r2;i++)

{

gotoxy(c1,r1+i);

cprintf("%c",179);

gotoxy(c1+c2+1,r1+i);

cprintf("%c",179);

}

gotoxy(c1,r1+i);

cprintf("%c",192);

for(i=1;i<=c2;i++)

cprintf("%c",196);

cprintf("%c",217);

textcolor(WHITE);

}

# Conclusion

The project Hospital Management System (HMS) is for computerizing the working in a hospital. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients, doctors and other staff that came up to the hospital.

It provides the basic information about the patients, doctors and the staff members like their contact no., address, issue they are facing and many more.