

## 

## G.H. Raisoni College of Engineering & Management

Department of Computer Engineering

SOFTWARE REQUIRMENT SPECIFICATION (SRS)

Topic: Hospital Management System

Subject: object oriented programming

**Submitted to**:

Ms.Diksha Boyar

**Submitted By:- Group B4**

1. Naveen Menon (scoa29)

2. Rushikesh More (scoa31)

3. Ganesh Pavne (scoa36)

**Table of Contents**

**1. INTRODUCTION**

1.1 Objective

1.2 Scope

**2. ABSTRACT**

**3. REQUIREMENT SPECIFICATION**

3.1 Software Requirement

3.2 Hardware Requirement

**4. ADVANTAGES**

**5. LIMITATION**

**6. REFERENCES**

**1. Introduction**

A **Software Requirement specification (SRS)** is a complete set of information about the system the system on which the developed project will be running. It include all software requirement the minimum system requrment as well as the recommended system requirement for running software are also mentioned in details separately. The aims of this document is to gather and analyze and give an in-depth insight of the complete software of the **Hospital Management System.**

1.1 Objective

It is use friendly application for Hospital which reduce the burdun and helps to manage all section of Hospital like reception, lab, inpatient/outpatient management & Billing etc., which improves the processing efficiency.

1.2 Scope

Hospital and healthcare center have undergone a change for its betterment .the administration of healthcare sector are opting IT solution for the better management & patient care in their hospital campus.

. Patient Registration

. Security of the whole system

. Billing and payment

. Resource allocation (booking rooms, operating theatres, etc.

**2. Abstract**

Many Hospital currently use a manual system for the management & maintenance of critical information. The current system numerous paper forms, with data stores spread through the hospital management infrastructure .often information is incomplete. Form are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

Analysis and Design of hospital management system is based on UML diagram.

A project of hospital management system includes registration of patient storing their details into the system and also computerized billing in the pharmacy .and test labs.

The main objectives of the system is which and helps us to collect most of the information about Hospitality & Medical services the system is very simple and design and to implement .

**3. Requirement Specification**

SRS001 **Add patients**

The HPIMS shall allow front– desk staff to add new patient to the system.

SRS002 **Assign ID**

Each patient a id and add it to the patient record, this id is used by the patient throughout his/her stay in hospital

SRS003 **Patient information**

Each patient shall have the following information: first name, last name, phone no., personal health no, address, postal code, ward name, bed no., and doctor’s name which was assigned.

SRS004 **Data Base**

The system shall use the MySQL database, which is open source and free

SRS005 **Operating System**

The Development environment shall be Windows 2000.

3.2 Software Requirements

. Windows 7 or above operating system

. JRE 1.8

. MySQL server

3.3 Hardware Requirement

. Core i5 processor

. 4GB Ram

. 20GB of hard disk space in terminal machines

. 1TB hard disk space in Server Machine

**4. Advantages**

- Immediate Access of data

- Friendly user interface

- Time Saving

- Data can be easily insert/update/delete

- Saving paper work

- Give facility of Different types of enquiry

- Data are easily approachable

# **5. Limitation**

1) Finance: the high cost of material for the project & everyday constraint increase in transportation tax.

2) Data Collection: The research work had a problem in collecting data form staff.

3) Material: it contains play a major role in that limiting some research that would have contributed immensely.

**6. References**

1. [www.google.co.in](http://www.google.co.in)
2. <https://www.slideshare.net>
3. <http://www.itu.dk/iEE> tasks

**7.Program Code**

#include<iostream>

#include<fstream>

#include<conio.h>

#include<process.h>

using namespace std;

class all

{

private:

struct address

{

int house;

char street[30];

char city[30];

char state[30];

char country[30];

};

struct age

{

int day;

int month;

int year;

};

struct patient\_info

{

char name[50];

address AD1;

age A1;

int martial\_status;

int reg\_no;

int bld\_group;

int sex;

}PI[100];

int task;

protected:

void enter\_patient\_info();

void show\_patient\_detail();

public:

all()

{ address a;

age b;

a.house=0;

b.day=0;

b.month=0;

b.year=0;

}

void software\_detail();

void tasks();

char answer;

char answer1;

char ch;

int serial;

};

class date

{

private:

int dat;

int month;

int year;

public:

date()

{

dat=0;

month=0;

year=0;

}

void enter\_date();

void show\_date();

};

class dob : public date

{

private:

struct dob1

{

int date;

int month;

int year;

int rem;

}DOB11[100];

public:

void enter\_date();

void show\_date();

};

int i=0;

int rem;

int count;

int regis;

int attempt;

int temp;

int show\_count=0;

all A1;

date D1;

dob DOB1;

int main()

{

count=0;

cout<<" \*\*\*HOSPITAL MANAGEMENT SOFTWARE\*\*\*"<<endl;

cout<<" By Group B-4 "<<endl;

D1.enter\_date();

A1.tasks();

}

void all::tasks()

{

attempt=0;

D1.show\_date();

cout<<"\*\*\*HOSPITAL MANAGEMENT SOFTWARE\*\*\*"<<endl;

cout<<" By Group B-4 "<<endl;

cout<<" \*\*Hospital Management Tasks\*\*"<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Please select a task to do...."<<endl;

cout<<"1. Enter a new patient information ."<<endl;

cout<<"2. View detail of existing patient ."<<endl;

cout<<"3. View detail about the program ."<<endl;

cout<<"4. Exit from the program ."<<endl;

cout<<"Enter your task serail :"<<endl;

cin>>task;

switch(task)

{

case 1:{

A1.enter\_patient\_info();

break;

}

case 2:{

A1.show\_patient\_detail();

break;

}

case 3:{

A1.software\_detail();

break;

}

case 4:{

//clrscr();

cout<<" Thank You for trying this program !!!"<<endl;

cout<<" This is the end of program...."<<endl;

cout<<"Press any key to exit....."<<endl;

getch();

exit(0);

break;

}

default:{

//clrscr();

cout<<"Invalid task serial ."<<endl;

cout<<"Press any key to continue...."<<endl;

getch();

// clrscr();

A1.tasks();

}

}

}

void all::enter\_patient\_info()

{ fstream file;

//clrscr();

answer='y';

if(count==0)

{

serial=1;

}

else

{

i=serial;

}

for(i=serial;answer=='y'||answer=='Y';i++)

{

PI[i].reg\_no=i;

temp=serial;

cout<<" \*\*\*ENTERING INFORMATION FOR PATIENT SERIAL NUMBER "<<i<<"\*\*\*"<<endl;

cin.get(ch);

cout<<"Registration Number : "<<PI[i].reg\_no<<endl;

cout<<"Enter the name of patient :"<<endl;

//clreol();

cin.getline(PI[i].name,50);

cout<<"Sex (1-Male 2-Female) :"<<endl;

//clreol();

cin>>PI[i].sex;

while(PI[i].sex!=1&&PI[i].sex!=2)

{

cout<<"Invalid input for sex of patient!!!"<<endl;

cout<<"Sex :"<<endl;

//clreol();

cin>>PI[i].sex;

}

cout<<"\*\*\*ENTERING ADDRESS\*\*"<<endl;

cout<<"House number :"<<endl;

//clreol();

cin>>PI[i].AD1.house;

while(PI[i].AD1.house<=0)

{

cout<<"Invalid input for house number ."<<endl;

cout<<"Again enter the house number ."<<endl;

//clreol();

cin>>PI[i].AD1.house;

}

cin.get(ch);

cout<<"Street :"<<endl;

//clreol();

cin.getline(PI[i].AD1.street,30);

cout<<"City :"<<endl;

//clreol();

cin.getline(PI[i].AD1.city,30);

cout<<"State :"<<endl;

//clreol();

cin.getline(PI[i].AD1.state,30);

cout<<"Country :"<<endl;

//clreol();

cin.getline(PI[i].AD1.country,30);

DOB1.enter\_date();

cin.get(ch);

cout<<"Martial status(1-Married,2-Not Married ):"<<endl;

if(count!=0)

{

//clreol();

}

cin>>PI[i].martial\_status;

while(PI[i].martial\_status<1||PI[i].martial\_status>2)

{

cout<<"Invalid input for martial status ."<<endl;

cout<<"Enter a valid martial status :"<<endl;

//clreol();

cin>>PI[i].martial\_status;

}

cin.get(ch);

if(count!=0)

{

// clreol();

}

//clreol();

cout<<"Blood group :"<<endl;

//clreol();

cout<<"1. A+ "<<endl;

//clreol();

cout<<"2. A- "<<endl;

//clreol();

cout<<"3. B+ "<<endl;

//clreol();

cout<<"4. B- "<<endl;

//clreol();

cout<<"5. AB+ "<<endl;

// clreol();

cout<<"6. AB- "<<endl;

//clreol();

cout<<"7. O+ "<<endl;

//clreol();

cout<<"8. O- "<<endl;

// clreol();

cout<<"Enter :"<<endl;

// clreol();

cin>>PI[i].bld\_group;

switch(PI[i].bld\_group)

{

case 1:

case 2:

case 3:

case 4:

case 5:

case 6:

case 7:

case 8:{

break;

}

default:{

while(PI[i].bld\_group!=1&&PI[i].bld\_group!=2&&PI[i].bld\_group!=3&&

PI[i].bld\_group!=4&&PI[i].bld\_group!=5&&PI[i].bld\_group!=6&&

PI[i].bld\_group!=7&&PI[i].bld\_group!=8)

{

// clreol();

cout<<"Invalid input !"<<endl;

cout<<"Blood Group :"<<endl;

// clreol();

cin>>PI[i].bld\_group;

}

break;

}

}

cin.get(ch);

cout<<"Want to enter information for another patient ? "<<endl;

//clreol();

cin>>answer;

count++;

serial++;

}

//clrscr();

A1.tasks();

file.open("E:\A29\hospital.txt",ios::out|ios::app);

file.seekp(0,ios::beg);

file.write((char \*)this,sizeof(all));

file.close();

}

void dob::enter\_date()

{

//clreol();

cout<<"Date of birth"<<endl;

//clreol();

cout<<"Year :";

//clreol();

//clreol();

cin>>DOB11[temp].year;

if(DOB11[temp].year<=0||DOB11[temp].year>10000)

{

do

{

//clreol();

cout<<"Invalid input for year !"<<endl;

cout<<"Please enter the year correctly :"<<endl;

cin>>DOB11[temp].year;

}while(DOB11[temp].year<0||DOB11[temp].year>10000);

}

//clreol();

cout<<"Month :";

//clreol();

cin>>DOB11[temp].month;

if(DOB11[temp].month<=0||DOB11[temp].month>12)

{

do

{

// clreol();

cout<<"Invalid input for month !"<<endl;

cout<<"Again enter the month :"<<endl;

// clreol();

if(count!=0)

{

// clreol();

}

cin>>DOB11[temp].month;

}while(DOB11[temp].month<0||DOB11[temp].month>12);

}

cout<<"Date :";

//clreol();

switch(DOB11[temp].month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:{

cin>>DOB11[temp].date;

while(DOB11[temp].date<1||DOB11[temp].date>31)

{

// clreol();

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

// clreol();

cin>>DOB11[temp].date;

}

break;

}

case 2:{

cin>>DOB11[temp].date;

if(DOB11[temp].year%4==0)

{

while(DOB11[temp].date<0||DOB11[temp].date>29)

{

// clreol();

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

// clreol();

cin>>DOB11[temp].date;

}

}

else

{

while(DOB11[temp].date<0||DOB11[temp].date>28)

{

// clreol();

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

// clreol();

cin>>DOB11[temp].date;

}

}

break;

}

default:{

cin>>DOB11[temp].date;

while(DOB11[temp].date<1||DOB11[temp].date>30)

{

// clreol();

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

// clreol();

cin>>DOB11[temp].date;

}

break;

}

}

//clreol();

}

void date::enter\_date()

{

cout<<"First of all I need the current date ..."<<endl;

cout<<"Year :";

cin>>year;

if(year<=0||year>10000)

{

do

{

cout<<"Invalid input for year !"<<endl;

cout<<"Please enter the year correctly :"<<endl;

cin>>year;

}while(year<0||year>10000);

}

cout<<"Month :";

cin>>month;

if(month<=0||month>12)

{

do

{

cout<<"Invalid input for month !"<<endl;

cout<<"Again enter the month :"<<endl;

cin>>month;

}while(month<0||month>12);

}

cout<<"Date :";

switch(month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:{

cin>>dat;

while(dat<1||dat>31)

{

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

cin>>dat;

}

break;

}

case 2:{

cin>>dat;

if(year%4==0)

{

while(dat<0||dat>29)

{

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

cin>>dat;

}

}

else

{

while(dat<0||dat>28)

{

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

cin>>dat;

}

}

break;

}

default:{

cin>>dat;

while(dat<1||dat>30)

{

cout<<"Invalid date !"<<endl;

cout<<"Again enter the date :"<<endl;

cin>>dat;

}

break;

}

}

}

void date::show\_date()

{

//clrscr();

cout<<"Hello....It's ";

cout<<dat;

rem=dat%10;

switch(dat)

{

case 11:

case 12:

case 13:

case 14:

case 15:

case 16:

case 17:

case 18:

case 19:

case 20:{

cout<<"th ";

goto over;

}

}

switch(rem)

{

case 1:{

cout<<"st ";

break;

}

case 2:{

cout<<"nd ";

break;

}

case 3:{

cout<<"rd ";

break;

}

default:{

cout<<"th ";

break;

}

}

over:

switch(month)

{

case 1:{

cout<<"January , ";

break;

}

case 2:{

cout<<"February , ";

break;

}

case 3:{

cout<<"March , ";

break;

}

case 4:{

cout<<"April , ";

break;

}

case 5:{

cout<<"May , ";

break;

}

case 6:{

cout<<"June , ";

break;

}

case 7:{

cout<<"July , ";

break;

}

case 8:{

cout<<"August , ";

break;

}

case 9:{

cout<<"September , ";

break;

}

case 10:{

cout<<"October , ";

break;

}

case 11:{

cout<<"November , ";

break;

}

case 12:{

cout<<"December , ";

break;

}

}

cout<<year<<" ";

}

void all::show\_patient\_detail()

{

do

{

//clrscr();

cout<<"Enter registration number :"<<endl;

//clreol();

cin>>regis;

cin.get(ch);

show\_count++;

if(regis>0 && regis<serial)

{

//clreol();

cout<<" \*\*\*INFORMATION FOR PATIENT REGISTRATION NUMBER"<<regis<<"\*\*\*";

//clreol();

cout<<"Name : "<<PI[regis].name<<endl;

//clreol();

cout<<"Sex : ";

//clreol();

if(PI[regis].sex==1)

{

cout<<"Male "<<endl;

//clreol();

}

if(PI[regis].sex==2)

{

cout<<"Female "<<endl;

//clreol();

}

cout<<"Blood Group : ";

//clreol();

switch(PI[regis].bld\_group)

{

case 1:{

// clreol();

cout<<"A+";

break;

}

case 2:{

// clreol();

cout<<"A-";

break;

}

case 3:{

// clreol();

cout<<"B+";

break;

}

case 4:{

// clreol();

cout<<"B-";

break;

}

case 5:{

// clreol();

cout<<"AB+";

break;

}

case 6:{

// clreol();

cout<<"AB-";

break;

}

case 7:{

// clreol();

cout<<"O+";

break;

}

case 8:{

// clreol();

cout<<"O-";

break;

}

}

//clreol();

cout<<"Date of birth : ";

//clreol();

DOB1.show\_date();

cout<<"Martial Status : ";

//clreol();

if(PI[i].martial\_status==1)

{

cout<<"Married "<<endl;

// clreol();

}

else

{

cout<<"Not married "<<endl;

//clreol();

}

// clreol();

cout<<" \*\*ADDRESS\*\*"<<endl;

// clreol();

cout<<"House no. : "<<PI[regis].AD1.house;

// clreol();

cout<<"Street : "<<PI[regis].AD1.street;

// clreol();

cout<<"City : "<<PI[regis].AD1.city;

// clreol();

cout<<"State : "<<PI[regis].AD1.state;

// clreol();

cout<<"Country : "<<PI[regis].AD1.country;

// clreol();

}

else

{

if(regis==1)

{

cout<<"Database is empty !!!"<<endl;

cout<<"Press any key to exit to main task menu...";

getch();

//clrscr();

A1.tasks();

}

attempt++;

if(attempt==3)

{

cout<<"You have entered wrong registration number 3 times ."<<endl;

cout<<"Access Denied!!! "<<endl;

cout<<"Please try again later. "<<endl;

cout<<"Press any key to exit to main task menu..."<<endl;

getch();

//clrscr();

A1.tasks();

}

//clreol();

cout<<"Sorry, the registration number is invalid ."<<endl;

cout<<"Press any key to continue...."<<endl;

getch();

// clreol();

A1.show\_patient\_detail();

}

//clreol();

cout<<"Want to see information of another patient :"<<endl;

//clreol();

cin>>answer1;

}while(answer1=='y'||answer1=='Y');

// clreol();

// clrscr();

A1.tasks();

}

void dob::show\_date()

{

cout<<DOB11[regis].date;

rem=DOB11[regis].date%10;

switch(DOB11[regis].date)

{

case 11:

case 12:

case 13:

case 14:

case 15:

case 16:

case 17:

case 18:

case 19:

case 20:{

cout<<"th ";

goto over;

}

}

switch(rem)

{

case 1:{

cout<<"st ";

break;

}

case 2:{

cout<<"nd ";

break;

}

case 3:{

cout<<"rd ";

break;

}

default:{

cout<<"th ";

break;

}

}

over:

switch(DOB11[regis].month)

{

case 1:{

cout<<"January , ";

break;

}

case 2:{

cout<<"February , ";

break;

}

case 3:{

cout<<"March , ";

break;

}

case 4:{

cout<<"April , ";

break;

}

case 5:{

cout<<"May , ";

break;

}

case 6:{

cout<<"June , ";

break;

}

case 7:{

cout<<"July , ";

break;

}

case 8:{

cout<<"August , ";

break;

}

case 9:{

cout<<"September , ";

break;

}

case 10:{

cout<<"October , ";

break;

}

case 11:{

cout<<"November , ";

break;

}

case 12:{

cout<<"December , ";

break;

}

}

cout<<DOB11[regis].year<<" ";

}

void all::software\_detail()

{

//clrscr();

cout<<" \*\*\*SOFTWARE DETAILS\*\*\*";

cout<<" Developer : Naveen Menon "<<endl;

cout<<" Developer : Rushikesh More "<<endl;

cout<<" Developer : Ganesh Pavane "<<endl;

cout<<" Programming Language : C++ "<<endl;

cout<<" Aim : Simulation of the software used in Hospital"<<endl;

cout<<"Hope you like it..."<<endl;

cout<<" Thank You for trying this program. "<<endl;

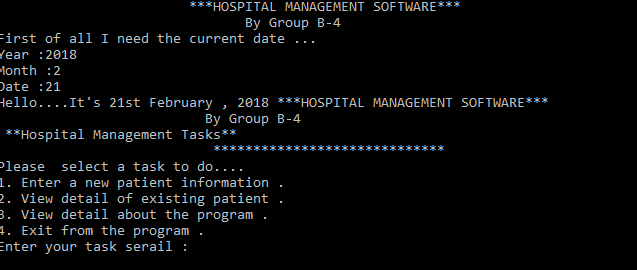
cout<<" Press any key to return to the main task menu......."<<endl;

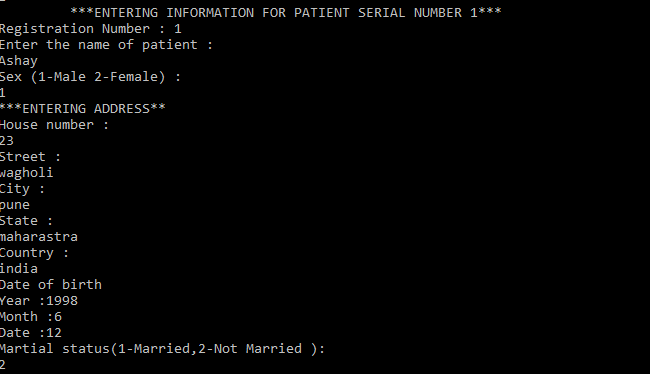
getch();

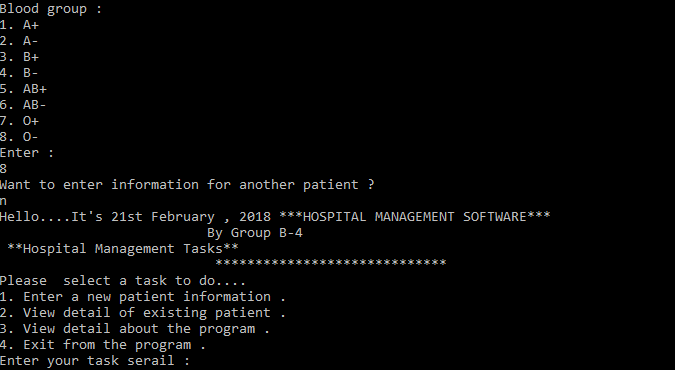
A1.tasks();

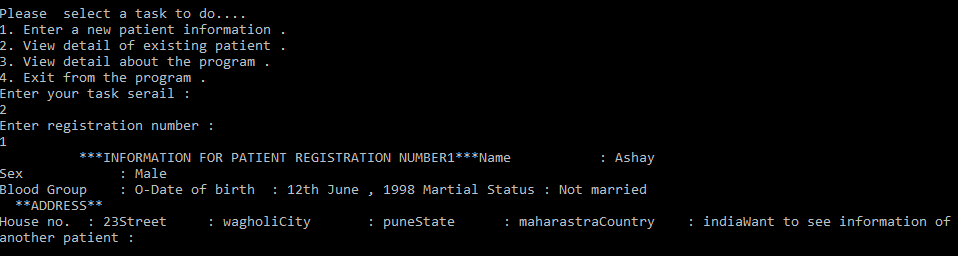
}

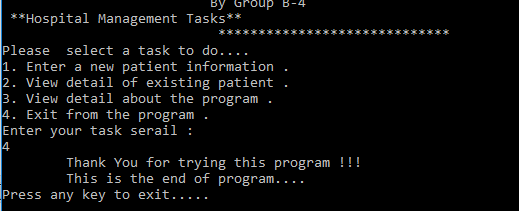
**8.Output**

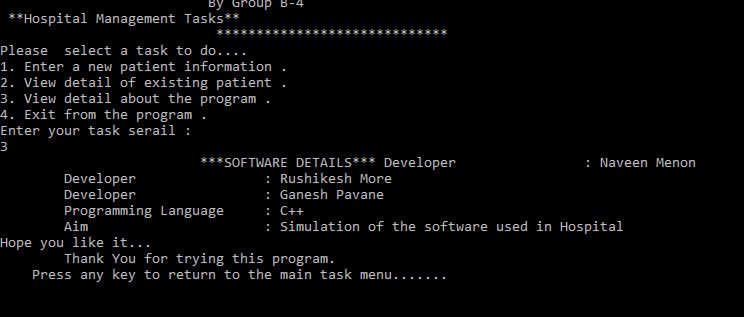


****

****

****

****

****