THAPATHALI CAMPUS
Institute of Engineering

A PROJECT PROPOSAL

PATIENT MANAGEMENT SYSTEM

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE COURSE OF CT 401 COMPUTER PROGRAMMING

BACHELOR OF ELECTRONICS, COMMUNICATION AND INFORMATION ENGINEERING

SUBMITTED BY

DIPESH GIRI THA075BEI016

RIWAJ NEUPANE THA075BEI034

CHARCHIT REGMI THA075BEI014

SUBMITTED TO

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

THAPATHALI CAMPUS

MARCH, 2019

DECLARATION

We hereby declare that the project work report entitled “patient management system” submitted for the partial fulfillment of the requirements for the course of CT 401 Computer Programming is our original work and the Project Work Report has not formed the basis for the award of any degree, diploma, or other similar titles.

Signature

Name of Student: Charchit Regmi

Signature

Name of Student: Dipesh Giri

Signature

Name of Student: Riwaj neupane

ACKNOWLEDGEMENTS

Without encouragement and inspiration, it is very difficult to build anything. A suitable method for patient management was felt to be necessary from a long time. The project “PATIENT MANAGEMENT SYSTEM’’ is one of the most reliable, secure and fastest method to fulfill user needs regarding patient management. It is based on easy interface for data entry of the patient .

We would like to thank our teachers for providing us project work which helped in building our knowledge regarding programming on the other level. We would like to thank our classmates who assisted us in our various problems.

ABSTRACT

“The purpose and essence of any Records Management system is the right information in the right place in the right order, at the right time for the right person at the lowest cost.” patient management system is a computerized system designed and programmed to deal with day to day operations taking place. The program can look after patient’s records, database treatments, status the purpose of the project is to computerize the Front Office Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The project outlines all the process followed to come up with the software that is from analysis to testing the system.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background

Hospitals are essential part of our lives, providing best medical facilities to people suffering from various kinds of diseases, which may be due to change in climatic condition, increased work load etc. patient management system is a computerized system designed and programmed to deal with day to day operations taking place. The program mainly focuses on managing patient details, editing and deleting those details. The purpose of the project entitled as “patient Management System” is to computerize the front office of management of hospitals and to develop a system which is user friendly, simple, fast and cost effective. It deals with the collection of patient information. Traditionally, it was done manually which is a paper- based system. It is too slow and cannot provide updated list of patients in a reasonable time frame. The intentions of the system are to reduce over-pay time pay and increase patients that can be treated accurately.

Hospital currently uses a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spreads throughout the hospital management infrastructure. It is also very inefficient and a time -consuming process. Recording and maintaining all these records is very unreliable, inefficient and error prone. It is also not economically and technically feasible to maintain records on paper. Often information is incomplete, or does not follow the management standard. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. The loss of data can cause various impacts on data management system. Multiple copies of same information exist in the hospitals may lead to inconsistencies in data.

Patient management system is powerful, flexible and easy to use and is designed and developed to deliver real conceivable benefits to hospitals and most importantly it is backed by reliable and dependable. Patient management system is designed for

multi- specialty hospitals, to cover a wide range of patient problems and management processes. It is an integrated end-to-end Patient Management system that provide relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting. This system is best suited to improve the quality and management of clinical care and hospital management care. This system allows to develop the organization and improve the effectiveness and quality of work. This system also provides low-cost reliable information of the existing systems. This system also provides excellent security of data at every level of user system interaction & also provides reliable storage and backup facilities. Due to backup facilities the data stored can be used as a future reference.

1.2 Problem Statement

Keeping track of all the activities and their records on a paper was very tedious task and error prone. It is also difficult to handle the whole system manually and is less accurate and to keep the data in case files for future reference because it may get destroyed. Moreover it is very difficult to retrieve data. The manual system is time consuming. To overcome the challenges faced in manual method we have developed an automated version of the manual system, named as “Hospital Management System”. The proposed system is very easy to operate, speed and accuracy are the main advantages of proposed system. The data are stored in the computer memory. It also helps on proper documentation of the records because the records can be used as a future reference at any given time. The proposed system will easily handle all the data and work done by the existing systems. It avoids the data inconsistency. It also helps to provide more security and integrity to data.

1.3 Objectives

The main objectives of our project are:

1. To design a system for better patient care.
2. To reduce hospital operating cost

1.4 Applications

In the context of Nepal where most of the hospital is not using computerized system to manage hospitals. This program will be a perfect way to manage the hospital system. It is found to be used in hospitals to keep record of its day-to-day activities and record of its patients and to add new patient record, search or edit patient record and list record of patients. It mainly uses file handling to perform basic operations like how to add, edit, search and delete record using file. It can revolutionize the way different hospitals and clinics are imparting healthcare services. It works towards making processes organized and productive by managing every aspect of hospital administration

1.5 Project Features

The main features of our project "Hospital Management System" are:

1. It improves the visibility and transparency in the complete management process and in all records.
2. It helps in streamlining the accurate reporting with the help of updated and accurate records.
3. It improves the customer services of the hospitals because it provides complete and accurate information at once
4. It also improves the management visibility of hospital, all information, and data regarding the patient can be seen by any department easily.
5. It allows unlimited user support whenever assistance is required by the client.
6. It allows complete data of system and store it in the same database which increases the efficiency, reduce the cost, simplifies the infrastructure and helpful in multiple recording.
7. It improves the time management because it allows getting access to any required data. It also allows to add or update information in few clicks.
8. It is easy to get access to the management system facilitates for the authorized users and keep it safe from unauthorized users.

1.6 Feasibility Analysis

1.6.1 Economic Feasibility

Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weight the cost and benefits associated with the candidate system and it suits the basic purpose of the organization. This feasibility checks whether the system can be developed with the available funds. The project does not require enormous amount of money to be developed. This can be done economically if planned properly so it is economically feasible.

1.6.2 Technical feasibility

Technical feasibility centres on the existing computer system and to what extent it can support the proposed system. This is concerned with specifying equipment and software that will meet the user requirement. The system maintains data in digital form so maintaining of information will be easy and retrieval of the information is fast. The system also has the facility to provide the required output in least time. In the case of this system, the required infrastructure i.e. hardware, software application already exist. Thus, the project is technically feasible.

1.6.3 Operational feasibility

This system reduces the workload on the staff because on a mouse click he can get the desired output. The system will be built in computer system which will make it more efficient, cheap and easy to use. The results obtained are more efficient and contains less errors than in manual system. The proposed system is better in use and user friendly as it generates proper messages at run time. Hence, for the following reasons the system is feasible.

1.7 System Requirement

1.7.1 Software Requirement

Operating system: DOS, Windows 7/8/9/10

Compiler: Turbo C

1.7.2 Hardware Requirement

Operating system: Windows

Hard disk: 40 GB

RAM: 25.6 MB

Processor: Pentium(R) Dual-core CPU

CHAPTER 2: LITERATURE REVIEW

2.1 Overview

In order to understand the concept associated with record management and or computer based records management system,it is imperative to examine and analyze published material from the expert regarding the field.The putpose of this review is to analyze and examine and obtain experience as regards the creation and archival processing of the electronic records.The review is based on an exhaustive assessment of the literature on computerized electronic management and electronic records,and contains an overview of the main concepts associated with the creation of an electronic record management system from the perspective of published experts.

2.2 RECORDS

A record is recorded information produced or received in the initiation, conduct or completion of an institutional or individual activity and that comprises content, context and structure sufficient to provide evidence of the activity regardless of the form or medium.

According to the National Archive and Records Administration (NARA) records include “all books, paper, maps, photographs, machine readable materials or other documentary materials,regardless of physical form or characteristics made or received or in connection with the transaction of public business and preserved or appropriate for preservation by that agency or its legimate successor as evidence of the organisation, functions, policies, decision, procedure, operations, or other activities of the Goverment or because of the informational value of the data in them”.

The International Council on Archives (ICA) committee on Electronic Records defines a record as “recorded information procedure or received in the initiation conduct or completion of an institutional or individual activity and that comprises content, context and strucuture sufficient to provide evidence of the activity.”The key word in these definitions is evidence put simply a record can be defined as evidence of even”

2.3 EXISTING SYSTEM

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, With data stores spread throughout the hospital management infrastructure.Often information (on forms) its incomplete,or does not follow management standards.Forms 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.Similarly,other hospital management system only shows the patient information with the patient invoice.

2.4 PROPOSED SYSTEM

The Patient Management System (PMS) is designed for Any Hospital to replace their existing manual paper based system.The new system is to control the following information, patient information, room availability, patient admitted room and which doctor is checking which patient etc. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently requried for such tasks.

2.5 OBJECTIVE OF THE SYSTEM

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, With data stores spread throughout the hospital management infrastructure.Often information (on forms) its incomplete,or does not follow management standards.Forms 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.

A significant part of the operation of any Hospital involves the acquisition, management and timely retrieval of great volumes of information.This information typically involves; Patient personal information and medical history etc.All of this

information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized PMS will automate the management of the hospital making it more efficeint and error free.It aims at standaridizing data,consolidating data ensuring data integrity and reducing inconsistencies.

CHAPTER 3: METHODOLOGY

2.1 System design

The purpose of design phase is to plan a solution for problem specified by the requirements. System design aims to identify the modules that should be in the system, the specification of those modules and how the interact with each other to produce the results. The goal of the design process is to produce a module that can be used later to build the system. The purposed model is called design of the system. System design is the process of defining the architecture, components, interfaces, and data for a system.

2.2 Software Requirement Specification

A software requirements a requirements for a system, is a complete description of the behaviour of the system to be developed and may include a set of use cases that describes interactions the users will have with the system. In addition it may also contain the non-functional requirements. Non-functional requirements impose constraints on the design or implementation such as performance enhancing requirements, quality standards, or other designer constraints. The above relation of software requirement specifications document should facilitate in providing the entire overview of the information system under development.

2.3 Purpose

The main purpose of system is to describe in a precise manner all the capabilities of the proposed system. It also states the various constraints which the system will be abide on. This document further leads to clear vision of software requirements, specifications and capabilities. These are to be expressed to the development testing and users software.

2.4 Overall description of the proposed system

The Patient Management System (PMS) is designed for Any Hospital to replace their existing manual paper based system.The new system is to control the following information, patient information, room availability, patient admitted room and which doctor is checking which patient etc. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently requried for such tasks. Patient management system is powerful, flexible and easy to use and is designed and developed to deliver real conceivable benefits to hospitals and most importantly it is backed by reliable and dependable. Patient management system is designed for multi- specialty hospitals, to cover a wide range of patient problems and management processes. It is an integrated end-to-end Patient Management system that provide relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting. This system is best suited to improve the quality and management of clinical care and hospital management care. This system allows to develop the organization and improve the effectiveness and quality of work. This system also provides low-cost reliable information of the existing systems. This system also provides excellent security of data at every level of user system interaction & also provides reliable storage and backup facilities. Due to backup facilities the data stored can be used as a future reference

Interfaces:

The applications will have a user friendly and menu based interface. Following screens will be provided;

1. A login screen for entering username, password where the maximum limit is 3 if it is crossed then the system will be automatically closed.
2. Add patient records: this menu allows us to enter the details about the user it asks the following information as:

a. First name.
b. Last name.
c. Age.

d. Gender.
e. Problem.
f. Contact no.
g. Email.
h. Date.

3. List patient’s records.
4. Search patient records.
5. Edit patient records.
6. Delete patient records.
7. Doctor lists.
8. Patient admit.
9. Admitted patient room search.
10. Lab test.
11. Discharge report.
12. Exit menu.

CHAPTER 4: IMPLEMENTATION AND RESULT

4.1 Implementation detail

The problem which made us to choose and work on these projects was to overcome the problems faced in patient management which was done is done manually. The list of library/header files used can be presented in the table below:

Header files	Main function
1.1 stdio.h	It is used to handle standard input and output operations.
1.2 conio.h	It is used declare several useful library functions for performing "console input and output".
1.3 windows.h	It contains decleration for all functions in c.
1.4 ctype.h	It contains functions for character handling.
1.5 string.h	It contains string handling functions like: strcpm,strcat,strev
1.6 stblib.h	It defines several general functions including dynamic memory management

List of various functions used are:

Name of function	Main task of function
Voidwelcome screen (void)	Welcome screen decleration,
void Title(void)	It is used to display title and main menu of the program.

void LoginScreen(void)	It is used to access the main login page of the function.
Void add rec(void)	It is used to add records.
Void search rec(void)	It is used to search the records.
void Edit_rec(void)	It is used to edit the already present records
void Dlt_rec(void)	It is used to already existing records.

Flow of the program

At first we create a structure named patient which contains its elements as:

a. First name
b. Last name
c. Age
d. Gender
e. Contact no
f. Email
g. Address
h. Phone number
i. Doctor name
j. Problem

1. **Welcome screen**: it simply displays the welcome and login page.

1.1 Main menu

It is used to perform various functions related to system. It allows the user to choose numbers from 1 to 6 where:

1-Add patient records.

2-list patient records.

3-Search patient records.

4-Edit records.

5-Delete records.

6-Exit.

Login screen page

It lets the user to a login page where they have to enter the username and password and the system compares the entered username and password to the original username and password. If it matches then the system is login but if not found same until three times the system automatically exit. So with the use of this page it ensures security to the system where the unauthorized users cannot access the system.

1.2 Add record

It is used to add records related to the patients. It is used to add details like:

1.2.1 First name: It urges the user to enter the first name of the patient. The max range of first name is 20 and min range is 2.

1.2.2 Last name: It urges the user to enter the last name of the patient. The max range of first name is 20 and min range is 2.

1.2.3 Gender: It asks the user to enter the gender. The only character valid are M and F.

1.2.4 Age: It asks the user to enter the age.

1.2.5 Address: It asks the user to enter the address. The max range of address is 20 and min range is 4.

1.2.6 Contact no: It asks the user to enter the phone no. It must contain 10 numbers.

1.2.7 Email: It asks the user to enter the email. The max range of email is 30 and min range is 8.

1.2.8 Problem: It asks the user to enter the problem. The max range of problem is 20 and min range is 4.

1.2.9 Prescribed doctor: It shows the prescribed doctor for the problems.

The above provided information can be edited by another provided facility of editing records which open a interference and asks the user if they want or don't want to edit their records.

1.3 Delete records

It is used to delete all the existing record you can not only add and edit your records but can also delete them and add a new record.

1.4 Exit menu

When all the options are checked and the user wants to exit the system they can exit using this system.

1.5 Doctors list

It opens upon a interference which presents the list of doctors of different specifications it allows the user to choose from number from 1 to 4 where

- Neurologist.
- Cardiologist.
- General physician.
- Surgeon.

1.6 Patient admit

It asks the user to enter the following informations

a. First name.
b. Last name.
c. Gender.
d. Age.
e. Address.
f. Contact number.
g. Email.
h. Problem.
i. Prescribed doctor.
j. Date.
k. Room no

1.7 Search admitted patient room

It asks the name of the user whose room number is to be searched

1.8 lab test of admitted patients

It asks to enter name of the admitted patient and it gives a list of tests to be performed

a. Urine test.
b. Blood test.
c. Thyroid test.
d. X-ray.
e. Stool test.
f. Ultrasound.

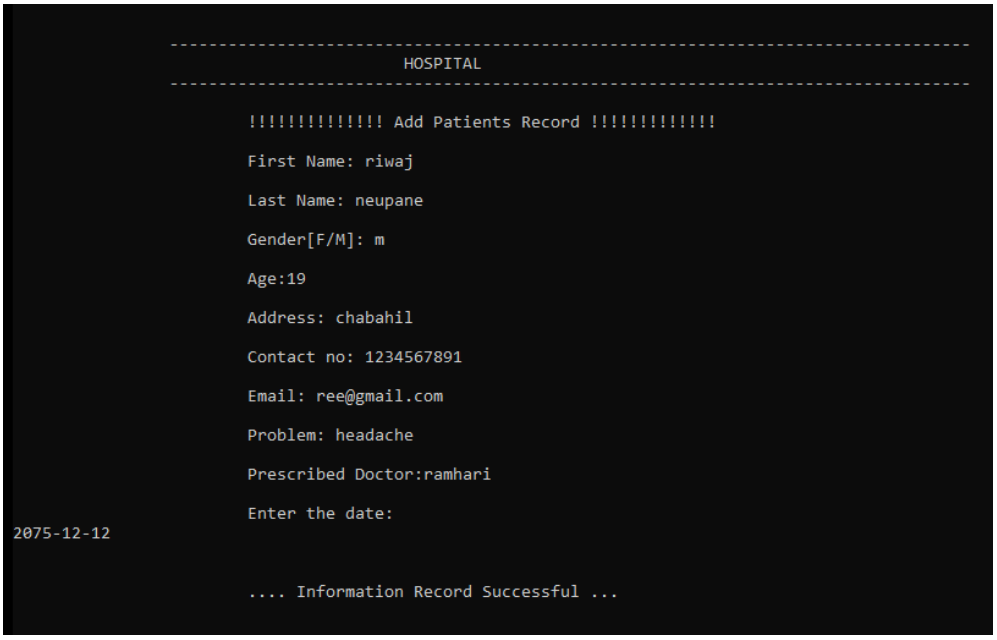
1.9 Discharge report

It asks the details of the patient and presents the total fee during process of admit.

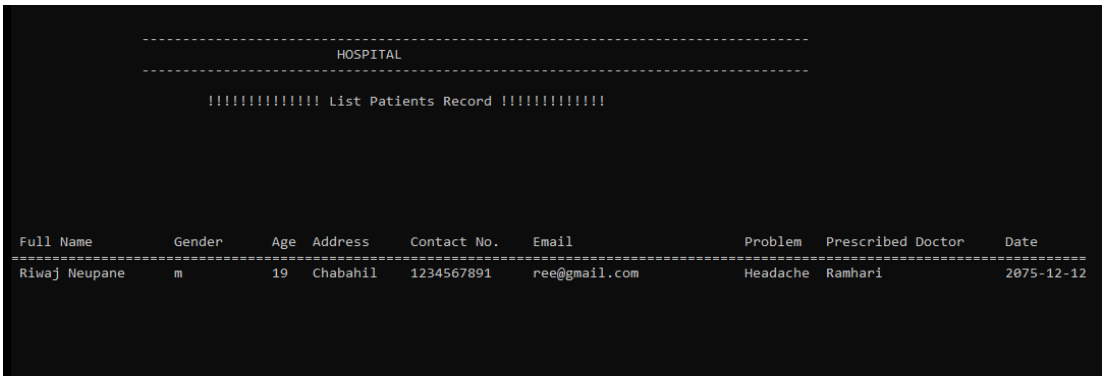
4.2 Result analysis

Main interference of system:

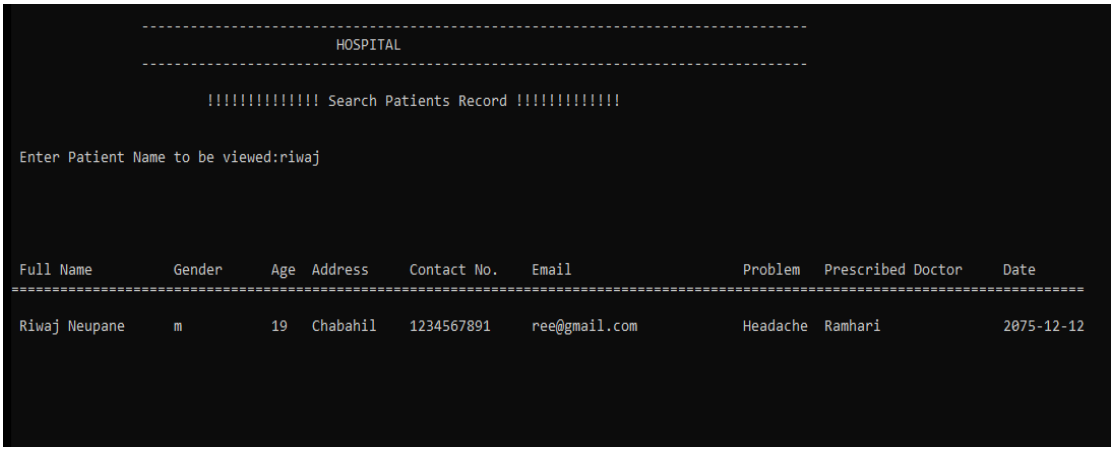
1. Add patient's records.



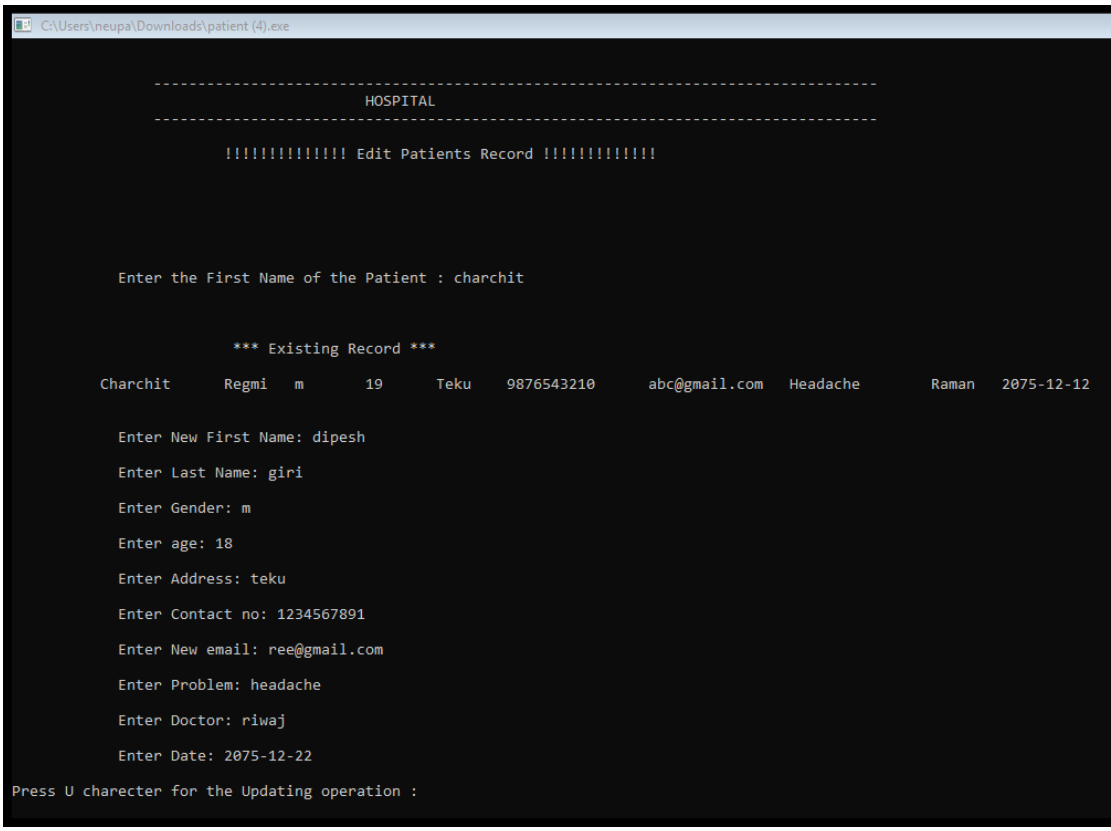
2. List patient records.



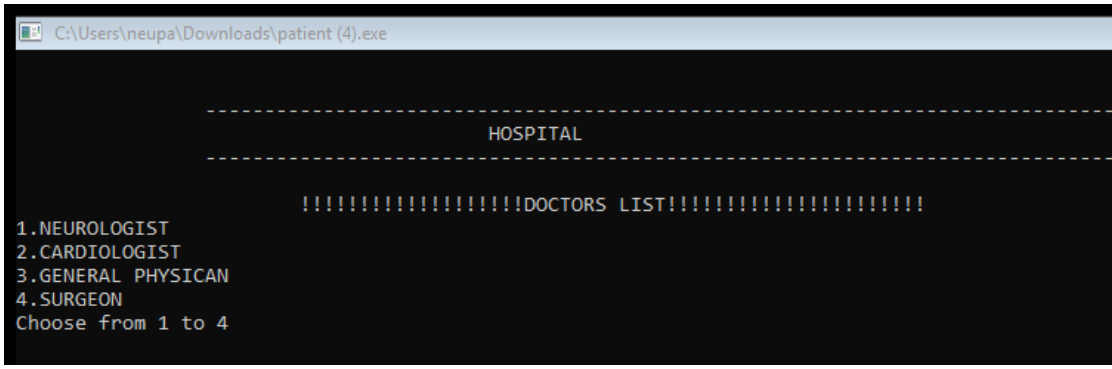
3. Search patient records



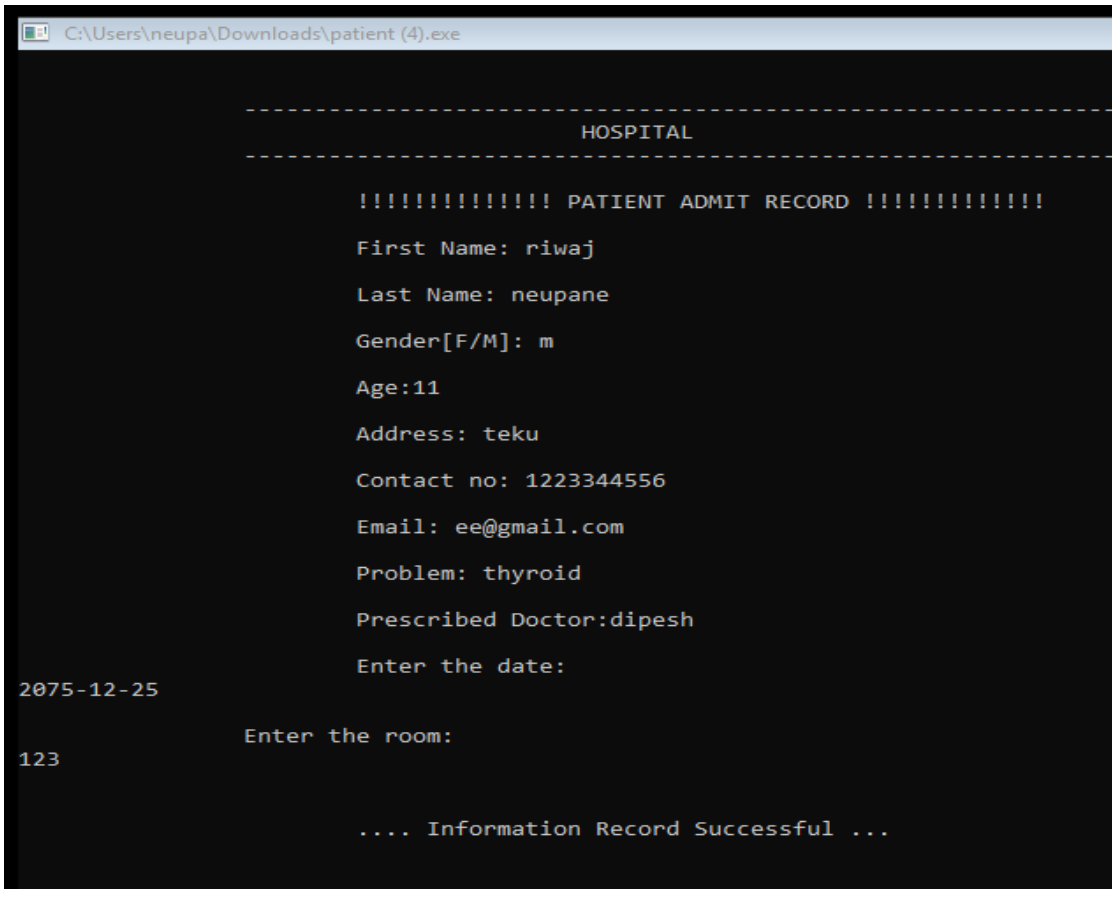
4. Edit record



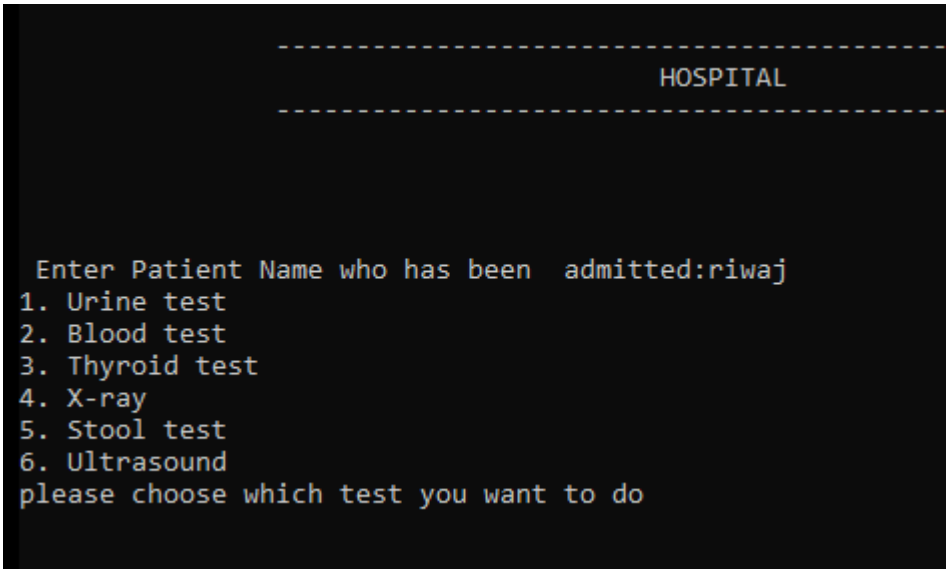
5. Doctor list



6. Patient admit



7. Lab test


HOSPITAL
Enter Patient Name who has been admitted:riwaj
1. Urine test
2. Blood test
3. Thyroid test
4. X-ray
5. Stool test
6. Ultrasound
please choose which test you want to do

CHAPTER 5: CONCLUSION AND FURTHER WORKS

5.1 Conclusion

Keeping track of all the activities and their records on a paper was very tedious task and error prone. It is also difficult to handle the whole system manually and is less accurate and to keep the data in case files for future reference because it may get destroyed. Moreover it is very difficult to retrieve data. The manual system is time consuming. To overcome the challenges faced in manual method we have developed an automated version of the manual system, named as "patient Management System". The proposed system is very easy to operate, speed and accuracy are the main advantages of proposed system. The data are stored in the computer memory. It also helps on proper documentation of the records because the records can be used as a future reference at any given time. The proposed system will easily handle all the data and work done by the existing systems. It avoids the data inconsistency. It also helps to provide more security and integrity to data.

5.2 Further works

Many features can be added to the system like referring a doctor stating about the overall departments of hospital keeping track of various activities that are to be performed on day to day basis operation of the hospital. It can also keep the track record of various financial transactions during execution of the program. It can also be further modified to a better system that can help to make smooth operation of the various tasks. We can perform improvements in billing process which will be faster than the manual method. Patient data can be more systemically captured at time of registrations that helps to provide more specific information about the patient. The system can be improved in such a way that it can provide feedbacks.

REFRENCES

[1] K. Kandel,"Learning C by examples, 6th edition", Kathmandu : Shree Chandeswori Publication's Press, 2062 B.S. [Accessed:March 2019]

[2] Slide share,"Hospital Management System Project",March,2018.[online]. Available: http://www.slideshare.net [Accessed: March, 2019.]

[3] Scribd,"Project-Hospital Management",March 2018.[online]. Available: http://www.scribd.com [Accessed:March,2019].

[4] Academia,"Hospital Management System",March,2019. Available: http://www.academia.edu [Accessed: March, 2019].

[5] Braian w. Kernighan and Dennis Ritchie, "The C Programming Language", Second Edition. [Accessed: March,2019].

[6] Robort lafore , " The waite groups C programming using turbo C++" second Edition. [Accessed: March,2019].

[7] Yashavant kanetkar , " let us C" ,seventh edition. [Accessed: March,2019].

[8] AI Kelly and Ira pohl, " A Book on C", fourth edition. [Accessed: March,2019].

[9] Wikipedia,"Literature Review",March,2019.[online]. Availabe: http://www.wikipedia.org [Accessed: March, 2019].

APPENDICES

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<ctype.h>

#include<windows.h>

#include<string.h>

char ans=0;

int choose;

int ok;

int b,valid=0;

void WelcomeScreen(void);

void Title(void);

void Main_Menu(void);

void LoginScreen(void);

void discharge(void);

void Add_rec(void);

void func_list(void);
```

```
void Search_rec(void);

void Edit_rec(void);

void Delete_rec(void);

void ex_it(void);

void Room_search(void);

void Patient_admit(void);

void lab(void);

void Doctor(void);

void NEUROLOGIST(void);

void CARDIOLOGIST(void);

void GENERAL_PHYSICAN(void);

void SUREGON(void);

void gotoxy(short x, short y)

{

COORD pos = {x, y};//sets co-ordinates in (x,y).

SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);

}
```

```
void Main_Menu(void)//function decleration
{
	system("cls");
	Title();// call Title function
	printf("\n\n\n\n\n\t\t\t\t1. Add Patients Record\n");
	printf("\n\t\t\t\t2. List Patients Record\n");
	printf("\n\t\t\t\t3. Search Patients Record\n");
	printf("\n\t\t\t\t4. Edit Patients Record\n");
	printf("\n\t\t\t\t5. Delete Patients Record\n");
	printf("\n\t\t\t\t6. Doctor list\n");
	printf("\n\t\t\t\t7. Patient admit\n");
	printf("\n\t\t\t\t8. Admitted patient room search\n");
	printf("\n\t\t\t\t9. Lab test\n");
	printf("\n\t\t\t\t10. Discharge report\n");
	printf("\n\t\t\t\t11. Exit\n");
	printf("\n\n\n\n\t\t\t\tChoose from 1 to 10:\n\n");
	scanf("%i", &choose);
	getch();
```

```
                switch(choose)//switch to differeht case
{
case 4:
    Edit_rec();
    break;
case 5:
    Delete_rec();
    break;
case 8:
    Room_search();
    break;
case 7:
    Patient_admit();
    break;
case 6:
     Doctor();
     break;
case 1:
```

```
        Add_rec();
        break;
case 2:
        func_list();
        break;
case 3:
       Search_rec();
       break;
case 9:
        lab();
        break;
case 11:
        ex_it();
        break;
case 10:
       discharge();
       break;
```

```
default:
   printf("invalid  character entered please entered correct character again\n");
   Main_Menu();
        }
}
void Patient_admit()
{
   system("cls");
        Title();// call Title function
        //list of variables
        char ans;
        FILE*op;
        op=fopen("Record1.dat","a");//open file in write mode
        printf("\n\n\t\t\t!!!!!!!!!!!!!! PATIENT ADMIT RECORD !!!!!!!!!!!!!\n");



        A:
        printf("\n\t\t\tFirst Name: ");
```

```
scanf("%s",p.First_Name);

p.First_Name[0]=toupper(p.First_Name[0]);

if(strlen(p.First_Name)>20||strlen(p.First_Name)<2)

{

printf("\n\t Invalid :( \t The max range for first name is 20 and min range is 2 :)");

goto A;

}

else

{

for (b=0;b<strlen(p.First_Name);b++)

{

if (isalpha(p.First_Name[b]))

{

valid=1;

}

else

{
```

```
                    valid=0;
                    break;
                }
            }
            if(!valid)
            {
                printf("\n\t\t First name contain Invalid character :(  Enter again
:)");
                goto A;
            }
        }

        /* ********************************************** Last name
********************************************** */
        B:
        printf("\n\t\t\tLast Name: ");
   scanf("%s",p.Last_Name);
   p.Last_Name[0]=toupper(p.Last_Name[0]);
   if(strlen(p.Last_Name)>20||strlen(p.Last_Name)<2)
```

```
{
        printf("\n\t Invalid :( \t The max range for last name is 20 and min range is 2 :)");
        goto B;
}
else
{
        for (b=0;b<strlen(p.Last_Name);b++)
        {
                if (isalpha(p.Last_Name[b]))
                {
                        valid=1;
                }
                else
                {
                        valid=0;
                        break;
                }
```

```
                }
                if(!valid)
                {
                        printf("\n\t\t Last name contain Invalid character :(  Enter again
:)");
                        goto B;
                }
        }
/* ************************************** Gender
************************************************************** */
        do
        {
           printf("\n\t\t\tGender[F/M]: ");
                scanf(" %c",&p.Gender);
                if(toupper(p.Gender)=='M'|| toupper(p.Gender)=='F')
                {
                        ok =1;
                }
                else
```

```
                {

                ok =0;

                }

      if(!ok)

           {

                printf("\n\t\t Gender contain Invalid character :(  Enter either F or M
:)");

          }

          }        while(!ok);

/* ************************************** Age
**************************************************************************
* */

   printf("\n\t\t\tAge:");

   scanf(" %i",&p.age);

/* ************************************** Address
************************************************************************ */

   do

   {

   C:

   printf("\n\t\t\tAddress: ");
```

```
    scanf("%s",p.Address);
    p.Address[0]=toupper(p.Address[0]);
    if(strlen(p.Address)>20||strlen(p.Address)<4)
        {
            printf("\n\t Invalid :( \t The max range for address is 20 and min range is 4 :)");
            goto C;
        }

}while(!valid);
/* ****************************************** Contact no.
****************************************** */
do
{
        D:
    printf("\n\t\t\tContact no: ");
    scanf("%s",p.Contact_no);
    if(strlen(p.Contact_no)>10||strlen(p.Contact_no)!=10)
        {
```

```
            printf("\n\t Sorry :( Invalid. Contact no. must contain 10 numbers.
Enter again :)");
            goto D;
        }
        else
        {
            for (b=0;b<strlen(p.Contact_no);b++)
            {
                if (!isalpha(p.Contact_no[b]))
                {
                    valid=1;
                }
                else
                {
                    valid=0;
                    break;
                }
            }
```

```
                if(!valid)
                {
                        printf("\n\t\t Contact no. contain Invalid character :(  Enter
again :)");
                        goto D;
                }
        }
}while(!valid);
/* ********************************************** Email
********************************************** */
do
{
   printf("\n\t\t\tEmail: ");
   scanf("%s",p.Email);
   if (strlen(p.Email)>30||strlen(p.Email)<8)
   {
      printf("\n\t Invalid :( \t The max range for email is 30 and min range is 8 :)");
         }
}while(strlen(p.Email)>30||strlen(p.Email)<8);
```

```
/* ********************************************************** Problem
********************************************** */

   E:

   printf("\n\t\t\tProblem: ");

   scanf("%s",p.Problem);

   p.Problem[0]=toupper(p.Problem[0]);

   if(strlen(p.Problem)>15||strlen(p.Problem)<3)

         {

               printf("\n\t Invalid :( \t The max range for first name is 15 and min
range is 3 :)");

               goto E;

         }

         else

         {

               for (b=0;b<strlen(p.Problem);b++)

               {

                     if (isalpha(p.Problem[b]))

                     {

                           valid=1;
```

```
                }
                else
                {
                        valid=0;
                        break;
                }
        }
        if(!valid)
        {
                printf("\n\t\t Problem contain Invalid character :(  Enter again
:)");
                goto E;
        }
    }
/* ****************************************** Prescribed Doctor
****************************************** */
    F:
   printf("\n\t\t\tPrescribed Doctor:");
   scanf("%s",p.Doctor);
```

```
p.Doctor[0]=toupper(p.Doctor[0]);

if(strlen(p.Doctor)>30||strlen(p.Doctor)<3)

        {

                printf("\n\t Invalid :( \t The max range for first name is 30 and min range is 3 :)");

                goto F;

        }

        else

        {

                for (b=0;b<strlen(p.Doctor);b++)

                {

                        if (isalpha(p.Doctor[b]))

                        {

                                valid=1;

                        }

                        else

                        {

                                valid=0;
```

```
                    break;
                }
            }
            if(!valid)
            {
                printf("\n\t\t Doctor name contain Invalid character :(  Enter again :)");
                goto F;
            }
        }
        G:
            printf("\n\t\t\tEnter the date:\n");
            scanf("%s",p.date);
             if(strlen(p.date)<9)
       {
          printf("Invalid date please enter again\n");
          goto G;
       }
       H:
```

```
        printf("\n\t\tEnter the room:\n");

        scanf("%s",p.room);

    fprintf(op," %s %s %c %i %s %s %s %s %s %s %s %s\n", p.First_Name,
p.Last_Name, p.Gender, p.age, p.Address, p.Contact_no, p.Email, p.Problem,
p.Doctor, p.room, p.date);

    printf("\n\n\t\t\t.... Information Record Successful ...");

    fclose(op);//ek file is closed

    sd:

    getch();

    printf("\n\n\t\t\tDo you want to add more[Y/N]?? ");

    scanf(" %c",&ans);

    if (toupper(ans)=='Y')

        {

        Patient_admit();

        }

    else if(toupper(ans)=='N')

        {

                printf("\n\t\t Thank you :) :)");

        getch();
```

```
            Main_Menu();
            }
    else
    {
       printf("\n\t\tInvalid Input\n");
       goto sd;
    }
}
void Room_search()
    {
            char name[20];
            system("cls");
            Title();
            FILE *op;
            op=fopen("Record1.dat","r");
            printf("\n\n\t\t\t!!!!!!!!!!!!!! Search Admitted Patient Room !!!!!!!!!!!!!\n");
            gotoxy(12,8);
            printf("\n Enter Patient Name whose room has to be searched:");
```

```
scanf("%s",name);

fflush(stdin);

name[0]=toupper(name[0]);

while(fscanf(op,"%s %s %c %i %s %s %s %s %s %s %s\n", p.First_Name,
p.Last_Name, &p.Gender, &p.age, p.Address, p.Contact_no, p.Email, p.Problem,
p.Doctor, p.room, p.date)!=EOF)

{

        if(strcmp(p.First_Name,name)==0)

        {



                printf("Full Name:");

                printf("%s %s\n",p.First_Name, p.Last_Name);

                printf("Gender:");

                printf("%c\n",p.Gender);

                printf("Age:");

printf("%i\n",p.age);

                printf("Address:");

printf("%s\n",p.Address);

                printf("Room no:");
```

```
                printf("%s\n",p.room);
                printf("\n");
                break;
            }
        }
        if(strcmp(p.First_Name,name)!=0)
        {
            gotoxy(5,10);
            printf("Record not found!");
            getch();
        }
    fclose(op);
    L:
    getch();
    printf("\n\n\t\t\tDo you want to view more[Y/N]??");
scanf("%c",&ans);
if (toupper(ans)=='Y')
{
```

```
        Room_search();
    }
        else if(toupper(ans)=='N')
        {
                printf("\n\t\t Thank you :) :)");
        getch();
                Main_Menu();
    }
        else
    {
        printf("\n\tInvalid Input.\n");
        goto L;
    }
}
```

Email: dipeshgiri492@gmail.com

Email: riwajneupane34@gmail.com

Email: charchitregmi10@gmail.com