

**LA GRANDEE INTERNATIONAL COLLEGE**

**Simalchaur, Pokhara, Nepal**

A Project Proposal/Report

on

**Patient Record System**

**Submitted to**

Ramesh Chalise

Bachelor of Computer Application (BCA) Program

In partial fulfillment of the requirements for the degree of BCA under

Pokhara University

**Submitted by**

Deven Shrestha

Sandesh Bastola

Nobel Baral

Susil Neupane

BCA, II

May 2, 2024

# Project Summary

In short, our program “Patient Record System” will assist in easier management of Patient’s record using a computer system. This program will allow the user to add new patient with their details which are their name, address, disease, age, sex, severity and prescribed medicine and the program will automatically assign the patient with an unique code which can later be used to easily identify a specific user easily within our program. You can also search for an already existing patient using their patient code or general information to either update their data, show their data or remove them from the “ongoing treatment” section and move them to “treated” or if they transferred to another hospital then move them to the “transferred” section. If a patient has already been in the hospital before and has arrived again, you can search within the “treated or transferred” to view their previous data and update the data and move him/her to “ongoing treatment” section and assign them a new patient code. The user can even remove the patient’s data.

# Table of Contents

[Project Summary ii](#_Toc165534672)

[Table of Contents iii](#_Toc165534673)

[List of Figures iv](#_Toc165534674)

[List of Abbreviation v](#_Toc165534675)

[1. Introduction 1](#_Toc165534676)

[2. Problem Statement 2](#_Toc165534677)

[3. Objective 3](#_Toc165534678)

[4. Methodology 4](#_Toc165534679)

[4.1 Algorithm 4](#_Toc165534680)

[4.2 Flowchart 6](#_Toc165534681)

[4.3 Data Flow Diagram 8](#_Toc165534682)

[5. Project Gantt Chart 9](#_Toc165534683)

[References 10](#_Toc165534684)

# List of Figures

[Figure 4.2.1 – Flowchart for adding patient to the database 6](#_Toc165534925)

[Figure 4.2.2 – Flowchart for Search/Update/View/Move/Remove data 7](#_Toc165534926)

[Figure 4.3.1 – Level 0 DFD for adding patient to database 8](#_Toc165534927)

[Figure 4.3.2 – Level 0 DFD for search/view/update/move/remove data 8](#_Toc165534928)

[Figure 4.3.3 – Level 0 DFD for re-adding previous patient 8](#_Toc165534929)

[Figure 5.1 - Project Gantt Chart for PRS 9](#_Toc165534930)

# List of Abbreviation

PRS: Patient Record System

BCA: Bachelor of Computer Application

DFD: Data Flow Diagram

Repo: Repository

Info: Information

# Introduction

Patient Record System provides an easy to use and efficient interface to perform various tasks related to patient’s data which are to be stored in a computer system. Our program aims to make hospital system more advanced and save time and resources of a hospital, which can be used to focus on more important and crucial tasks inside a hospital such as tending to a needy patient. This program will be written in C programming language, which is a middle level language to ensure great performance and efficiency of such program. This program using such low resources in a computer will allow other crucial hospital services which might be running under the same computer more smoothly and interrupt free. When, the program runs, it will ask for patient’s information such as name, age, sex, address, prescription, and severity and store it in the database. Conveniently the program will assign the patient their unique code which can easily be used to identify and perform any changes on their data. In addition to that, the program will allow the user to search the patient using their code and provide option to view, update, move or remove data. When “view” is chosen, then the program will show the data of the patient assigned to that code to the screen, when “update” is chosen, then the user will be provided an interface where they can update the parts of data they will like to, when the user chooses move, then they will be presented with the option to either move the patient to “cured” or “transferred” section. “cured” option is to be chosen, when the patient has fully recovered and can be discharged from the hospital safely, while “transferred” section is to be chosen when the patient has been transferred to another hospital and are still undergoing treatment over there. On the other hand, when “remove” is chosen, then the patient’s data is removed from the system and is no longer stored. Moreover, this system also facilitate the situation when the patients who have been to the hospital before have returned with newer diseases. When they have returned the system asks for their code which they got when they were staying in the hospital before and checks if it is valid. If it is, then the program provides and interface to update their required information which might have changed over time and the program moves them from the previous patient database to the current one. So, our system “Patient Record System” is a feature-full and user-friendly system which will facilitate the hospital which implement it to be more advanced and efficient. It is also very fast and optimized allowing for the computer to efficiently perform various activities without being pulled down by it.

# Problem Statement

A lot of people come in a hospital to get checked and cure their diseases. Of course, the hospital can manage the patient’s data in a traditional system using paper files but that way is more risky and time consuming. Papers are prone to damage through fire, water and other components and can easily be destroyed. In addition to that searching through different files just to update one patient’s information can be a time consuming process. Furthermore some files can be lost and cause massive problem to both the patient and the hospital. We made this program, so that the hospitals won’t have to worry about the files being lost or damaged from various reason. In addition to that, we hope that the time saved from the usage of this program while performing addition, modification, removal and transfer of patient and their data will cause the improved productivity and lead to further technological advancements in the hospital.

# Objective

The objective of creating this program are as follows:

* Provide easy to use and manage interface to record and manage patient’s data
* Save time and resources of a hospital

# Methodology

We are going to use various programming techniques we will learn through the internet and in college to create the PRS system. Here are the current methodology we have planned that we will use to create the system:

* All members of our group will create a github account and commit all their contributions in a repo named Patient-Record-System-in-C
* There will be different folders for each members corresponding to our names in the repository, each members will commit only in their corresponding repository and after discussion with each member the approved code will be applied in the main program.
* The program will store previous patient’s data and current patient’s data.
* Each patient will have their own unique patient code.
* When the program runs, it will first ask if the patient has been in our hospital before and if he/she has, the program will ask the patient code of the patient while he/she were admitted in the hospital and search through the previous patient’s data and the user can update the data and the patient will be assigned a new code.
* If the patient has not been in our hospital then it will ask the user to enter patient’s name, address, age, sex, disease, severity and prescribed medicine.
* The user can search for current and previous patient records using the patient’s code to update, view, move or remove them.

A program is easier to build when you have laid your plans in a form of algorithms, flowcharts and DFD. Algorithms, flowcharts and DFD can avoid burn out from being overwhelmed by a project.

## Algorithm

* **Add patient to the database:**

Step 1: Start

Step 2: Declare variables name, age, address, disease, severity and prescription

Step 2: Ask if the patient has been to our hospital before

Step 3: If yes then ask for his/her code, search through the “cured or transferred” section, if the patient is not found then go to step 2, if the patient is found then ask for new age, disease, severity and prescribed medicine, if no then ask to input patient’s name, age, address, disease, severity and prescription.

Step 4: Assign the patient his/her patient code which will be the array number they happen to have their data stored on

Step 5: Ask if the user wants to add another patient

Step 6: If yes then go to Step 2, if no then move onto the next step

Step 7: End

* **Search/View/Update/Move/Remove patient’s data:**

Step 1: Start

Step 2: Ask to input patient’s code

Step 3: If patient’s code exist, then show the patient’s data, if this doesn’t then print “Wrong or Invalid Code” and go to Step 2

Step 4: Ask if the user wants to update, move or remove the patient’s data

Step 5: If the user chooses to update, then it will allow the user to type the patient’s newer information, if the user chooses to move, it will ask whether the user wants to move the data to “cured” or “transferred”, if the user chooses “cured” then the patient’s data will be moved to the cured section and if the user chooses “transferred” then the patient’s data will be moved to the transferred section, if the user chooses remove, then the patient’s data will be deleted.

Step 6: Ask if the user wants to modify other patient’s data

Step 7: If yes then go to Step 2, if no then move onto the next step

Step 8: End

## Flowchart

* **Add patient to the database**

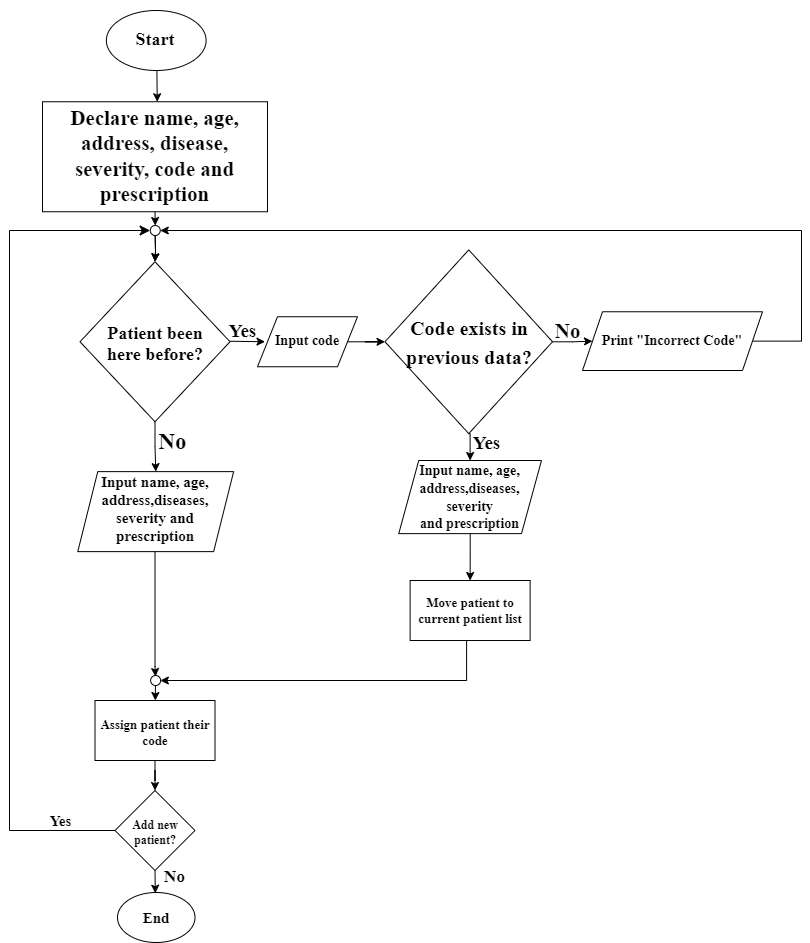


Figure 4.2.1 – Flowchart for adding patient to the database

* **Search/View/Update/Move/Remove patient’s data**

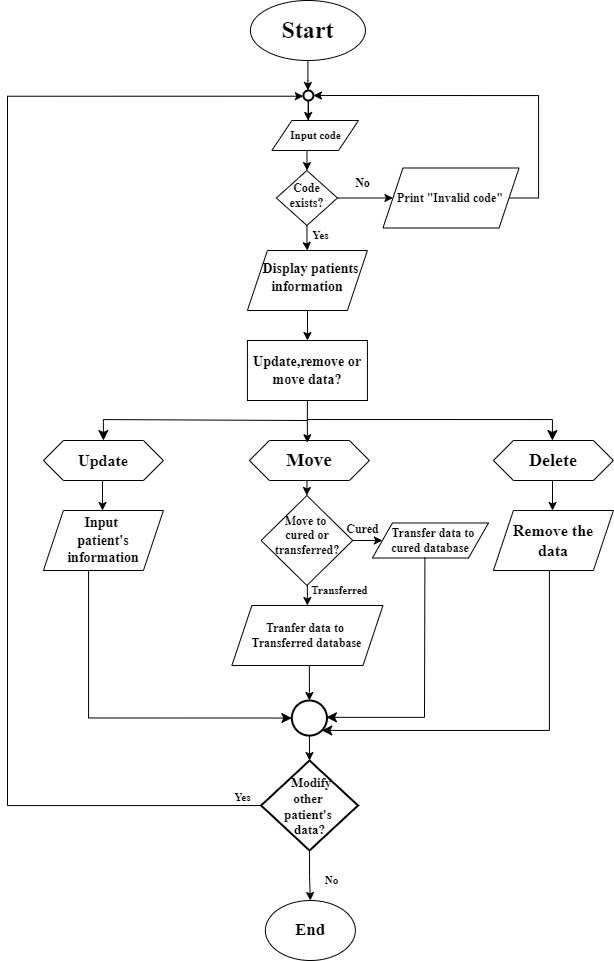


Figure 4.2.2 – Flowchart for Search/Update/View/Move/Remove data

## Data Flow Diagram

* **Add patient to the database**

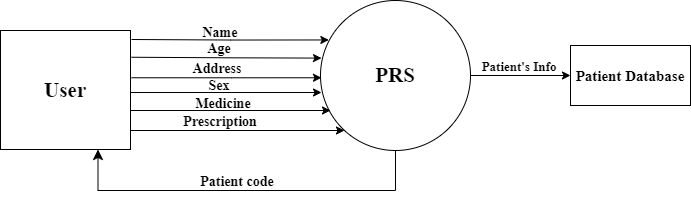


Figure 4.3.1 – Level 0 DFD for adding patient to database

* **Search/View/Update/Move/Remove Data**



Figure 4.3.2 – Level 0 DFD for search/view/update/move/remove data

* **Re-adding Previous Patient**



Figure 4.3.3 – Level 0 DFD for re-adding previous patient

# Project Gantt Chart

Figure 5.1 - Project Gantt Chart for PRS

# References

Dick, R. S., Steen , E. B., & Detmer, D. E. (1997). *The Computer-Based Patient Record: Revised Edition: An Essential Technology for Health Care.* Retrieved from National Library of Medicine: https://www.ncbi.nlm.nih.gov/books/NBK233055/