

Report on

**HOSPITAL MANAGEMENT SYTSTEM**

*Submitted by:*

Team Prometheus:

Leader:	Shaheer Farrubar Shamsi	ID:2222127642
Member:	Fayzul Haque Mayin	ID:2221822642
Member:	Md. Zahid Hassan	ID:2221833642
Member:	Tanbir Islam	ID:2222621042

Under the Supervision of  
Md. Minhazul Islam

DATE:  
07/09/2022

**Department of Electrical & Computer Engineering**

**North South University**

### **DECLARATION**

We hereby certify that the work was completed by the team members. None of the code is plagiarized. The project was completed timely with the objective of learning teamwork, leadership and proficiency in the C programming language.

The work presented in this report has been submitted as the project for CSE115.1L course during the semester Summer 2022 of North South University.

# **CONTENT**

## **INTRODUCTION**

### **1. What is hospital management system?**

A hospital management system (HMS) is an integrated solution that handles different workflows of doctors, patients and administrators. It manages the records of doctors and patients for ideal management. HMS is built with administrative operations as well as features for doctors. The HMS is a cornerstone for the successful operation of the hospital and its patients.

### **2.How is it developed?**

The hospital management system presented in the project was developed using the C programming language using many of its features like loops, functions, conditional input and output, structures and files.

### **Software tools used:**

- 1. C programming language**
- 2. Codeblocks**

**Advantages of hospital management system:**

- (i) Efficient workflow for doctors
- (ii) Clear records for reference
- (iii) Systematic control of operations
- (iv) Ease of use
- (v) Administrative control over all records
- (vi) User-friendly software
- (vii) Cost-effective and easy to install
- (viii) Any information through this system
- (ix) Saves human effort and time
- (x) Reduces the chances of error.
- (xi) Acts as anti-theft and prevents mismanagement

## **FUNCTION**

### **MISCELLANEOUS FUNCTIONS:**

void loading() : loading screen

int anyKey() : press any key function

char animatedText(char\* txt) : animates text

### **DOCTOR RECORD FUNCTIONS:**

void docAdd(),void docView(),void docEdit(),void  
docSearch(),docDel(),void docRead(),void docWrite()

### **PATIENT RECORD FUNCTIONS:**

void patAdd(),void patView(),void patEdit(),void  
patSearch(),patDel(),void patRead(),void patWrite()

## **WORK DISTRIBUTION:**

### **Shaheer Farrubar Shamsi: (60% of the Project)**

features implementation, patient functions, doctor functions, defining structures, login management and verification, writing to file, report writing, team leading, debugging code

### **Fayzul Hoque Mayin (2221822642): (15% of the Project)**

notes from class, collection of labwork example code provided by instructor, group discussion, idea for features

### **Md. Zahid Hassan (2221833642): (12.5% of the Project)**

attempting some functions, group discussion, idea for features

### **Tanbir Islam (2222621042): (12.5% of the Project)**

welcome message, group discussion, idea for features

## SOFTWARE TOOLS USED:

### C

C ([/ˈsiː/](#), as in the letter *c*) is a general-purpose computer programming language. It was created in the 1970s by Dennis Ritchie, and remains very widely used and influential. By design, C's features cleanly reflect the capabilities of the targeted CPUs. It has found lasting use in operating systems, device drivers, protocol stacks, though decreasingly[6] for application software. C is commonly used on computer architectures that range from the largest supercomputers to the smallest microcontrollers and embedded systems.

A successor to the programming language [B](#), C was originally developed at [Bell Labs](#) by [Dennis Ritchie](#) between 1972 and 1973 to construct utilities running on [Unix](#). It was applied to re-implementing the kernel of the Unix operating system.[7] During the 1980s, C gradually gained popularity. It has become one of the [most widely used programming languages](#),[8][9] with C [compilers](#) available for almost[*citation needed*] all modern [computer architectures](#) and operating systems. C has been standardized by [ANSI](#) since 1989 ([ANSI C](#)) and by the [International Organization for Standardization](#) (ISO).

C is an [imperative procedural](#) language supporting [structured programming](#), [lexical variable scope](#), and [recursion](#), with a [static type system](#). It was designed to be [compiled](#) to provide [low-level](#) access to [memory](#) and language constructs that map efficiently to [machine instructions](#), all with minimal [runtime support](#). Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A [standards-compliant](#) C program written with [portability](#) in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code.[10]

Since 2000, C has consistently ranked among the top two languages in the [TIOBE index](#), a measure of the popularity of programming languages.[11]

### Code::Blocks

Code::Blocks is a [free, open-source cross-platform IDE](#) that supports multiple [compilers](#) including [GCC](#), [Clang](#) and [Visual C++](#). It is developed in [C++](#) using [wxWidgets](#) as the [GUI](#) toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards [C](#), [C++](#), and [Fortran](#). It has a custom [build system](#) and optional [Make](#) support.

Code::Blocks is being developed for [Windows](#) and [Linux](#) and has been ported to [FreeBSD](#),[2] [OpenBSD](#)[3] and [Solaris](#). [4]