



**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
THAPATHALI CAMPUS**

**Proposal  
On  
Banking Management System**

**Submitted by:**

Raman Rajbanshi (THA081BCT029)

Anuj Singh (THA081BCT003)

Rajan Neupane (THA081BCT027)

Purushottam Raj (THA081BCT026)

**Submitted to:**

Department of Electronics and Computer Engineering Thapathali Campus  
Kathmandu, Nepal

February, 2025

## **ABSTRACT**

It comes from the fact that building complex programs with C requires a lot of knowledge, functions. Although we are deeply integrated in systems everywhere around us we still find it difficult to question how it's formed. BMS (Banking Management System) is the proposed system to interact with the banking system covering monetary assets. We are familiarizing with this system so that from the project we can acknowledge the different systems and its formation

## **Table of Contents**

<b>ABSTRACT .....</b>	<b>i</b>
List of Figures .....	ii
List of Tables .....	iii
<b>LIST OF ABBREVIATIONS.....</b>	<b>iv</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 Background .....	1
1.2 Motivation .....	1
1.3 Problem Definition .....	1
1.4 Objectives .....	1
<b>2. LITERATURE REVIEW .....</b>	<b>2</b>
2.1 Introduction .....	2
<b>3. PROPOSED SYSTEM ARCHITECTURE .....</b>	<b>3</b>
3.2 Block Diagram or System Diagram .....	3
3.2 Parts of Program .....	4
3.3 Tools and Environment .....	4
<b>4. METHODOLOGY .....</b>	<b>5</b>
4.1 Header FILES .....	5
4.2Structures .....	5
4.3 File Handling .....	5
4.4 Authentication .....	5
<b>5. SCOPE AND APPLICATIONS .....</b>	<b>6</b>
<b>6. TIME ESTIMATION .....</b>	<b>7</b>
<b>7. FEASIBILITY ANALYSIS .....</b>	<b>8</b>
7.1 technical Analysis .....	8

7.2 Economical Feasibility.....	8
7.3 Operational Feasibility.....	8
7.4 Time feasibility.....	8
<b>8. Requirements .....</b>	<b>9</b>
8.1 Hardware Requirements.....	9
8.2 Software Requirements .....	9
<b>9. References.....</b>	<b>10</b>

## 8 List of Figures

Figure 3-1-1:Flowchart for program.....	3
Figure 6-1: Time Estimation Gantt Chart.....	7

## **List of Abbreviations**

ATM          Automated teller machine

SWIFT        Society for Worldwide Interbank Financial Telecommunication

AI            Artificial Intelligence

DeFi         Decentralized Finance

GCC          GNU Compiler Collection

IDE          Integrated Development Environment

OS           Operating System

# **1. INTRODUCTION**

## **1.1 Background**

A Banking Management System, or BMS, is software that provides services like balance enquiry, balance transfer, opening account, deposit or withdraw money. Traditional banking system has been replaced by digital banking services, ensuring secure, fast and efficient transactions. A digital banking system provides a solution by automating banking operations. The software will be created using C. C provides its own libraries supporting math, strings and many other standard things. Just to make development easier, preventing the programmers for tiresome work the details are stored in the file and accessed later for displaying it. It will be a very functional software which as discussed will implement the technological factors in the banking system. Thanks to C, the functioning and the implementation is smooth.

## **1.2 Motivation**

As our first project on C programming language, we as a group decided to get into system development with C. As the manual banking system are time consuming and prone to human errors. The growth in technologies, a need for a secure and efficient banking system enhance user experience and provides more productivity. We were in search of solution motivated us for this system and we looked out for tutorials and videos on YouTube and the web. This led us to this project ultimately. It was a pretty good opportunity for us to test ourselves in this early phase of learning. Thanks to our teammate who, in the decision phase of the project, introduced us to this BMS system.

## **1.3 Problem Definition**

As our project is based on banking management system, there are inefficient manual banking processes. Traditional banking system face several challenges that impact efficiency, security, and user experience. Manual processes such as account management and transaction handling, are prone to human errors, delays and inefficiencies. Breaches and electronic fraud poses a tremendous threat to bank and user. Non integration of e-banking with real –time monitoring, fails to detect real-time fraud and comply with monetary regulations. This project attempts to deal those issues with secure and user friendly banking management system.

## **1.4 Objectives**

- To develop a secure, user-friendly banking management system.
- To make extensive use of C to give logic and necessary attributes to the system and to give the necessary environment.
- To facilitate the users with these functions of the system

- To integrate secure authentication and transaction processing.

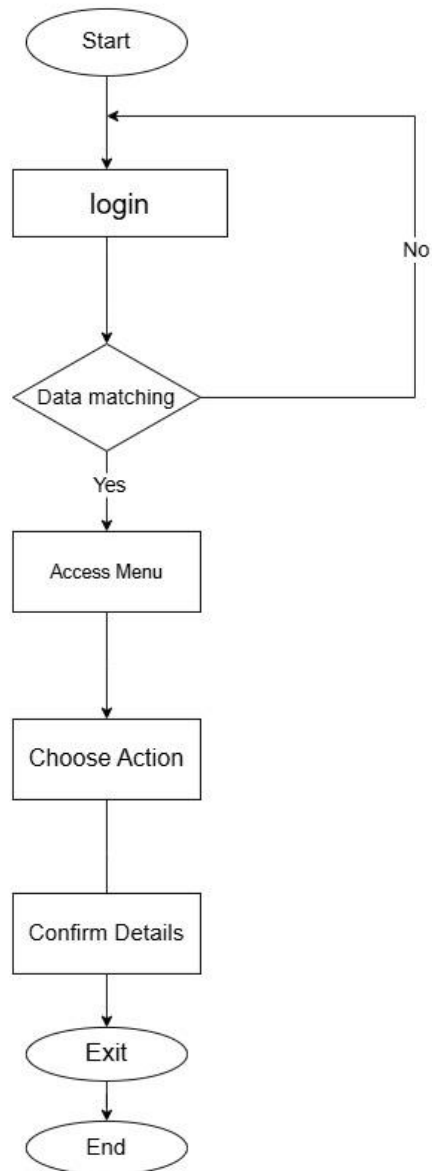
## **2. LITERATURE REVIEW**

Banking management systems continue to provide an indispensable solution in the modern world of finance, facilitating the very way banks run their functionality, luring in highly marketable innovations, having cash flow security issues, and generally improving the banking experience of clients. From those near-distant pasts heavily reliant on human manual banking methods, New Age banking systems have provided an avenue to automate those manual processes as originally pioneered in Fintech to some of the more modern evolutions taking place. Historically, banking management systems were known around the 19th century, with banks using electromechanical machines for keeping records. Computerized banking in the broader context evolved in the 1960s, while other banking technologies developed between the 1970s and 1980s, such as ATM networks and SWIFT systems, became invaluable tools in the global finance market. Somewhere in the early years of the Internet-from the 1990s-going online banking and enterprise approaches furthered the centralized customer experience in brokering self-automated processes. The boom in the 2000s and 2010s was a rise of mobile banking, cloud-based solutions, and AI-based security systems, rendering banking much easier and safer. Banking management systems now use cryptocurrency transactions, decentralized finance (DeFi), and biometric authentication to help build a future with digital banking. Such banking systems have remained critical tools of delivering efficient operations, regulatory compliance, and client quality assurance, at any time as and as they are sought out by financial organizations for technological innovation.



### 3. PROPOSED SYSTEM ARCHITECTURE

#### 3.1 Block Diagram or System Architecture



**Figure 3.1.1: Flowchart for program**

## 3.2 Parts of Program

### 3.2.1 User input

The system takes input from the user in the form of account number, user credentials (username, password or PIN).

### 3.2.2 Data processing

Once the user input is received, the system processes it with various steps, including:

- **Account Verification:** checks if the account number is valid and exists in the system in the database.
- **Authentication & Authorization:** Ensures secure login using multi-factor authentication (MFA) if enabled.
- **Transaction Verification:** Validates transaction details such as recipient account number, amount, and transaction type

### 3.2.3 Main banking system

The system includes some Main important functions

- **Check Account Balance** - Display the available balance to the user and recent transaction done by user.
- **Deposit and withdraw** - Add or withdraw fund through the bank account through different medium online transfer, check or ATM.

### 3.2.4. User Output

- **Display to user** - After processing, the system displays relevant information related to transaction status, Balance update or security alerts.

## 3.3 Tools and Environment

- **VS Code:** A lightweight and powerful code editor for writing and testing the C code.
- **Gcc:** GCC to compile the code into an executable form.
- **GitHub :** GitHub is a platform for version control, sharing and code management.

## **4. METHODOLOGY**

The different methodologies that were used to gather the required tools and features are different types of header files, structures and the use of file handling.

### **4.1 Header Files**

In C language, there are header files that contain a set of predefined standard library functions. The major header files used in this program are:

- i. `stdio.h` - inputs and output functions such as `printf()`, `scanf()`, `fopen()`, `fread()` etc.
- ii. `string.h` - functions like `strcmp`, `strcpy` , for comparing and copying strings.
- iii. `stdlib.h` - functions like `cls`.
- iv. `lms.h` – custom header file

### **4.2 Structures**

Structures are containers that can hold several different types of data types in a single container and thus can be accessed using the ‘.’ operator. It essentially creates a data type of its own which makes it easier to access certain elements without the need for repetition. In this project we have made use of them to create various structures such as student, teacher with elements like name, assignments inside them such as to make it easier to access them.

### **4.3 File Handling**

File handling refers to the use of various in-built functions in C to manipulate files and read, write or append them. In this program we have used file handling to store the data of the students, the teachers and access them whenever needed. Various functions such as `fopen()`, `fclose()` , `fread()`, `fwrite()` etc have been used as the condition requires to efficiently store the data.

### **4.4 Authentication**

As we have also created a login mechanism for students and teachers with their Ids and passwords. For authentication, we’ve made use of the string compare function (`strcmp`). We’ve stored the data in Id and password elements of the structure when signing up and when they try to sign in later, using `strcmp` we compare them to see if the Id and password are correct and authenticate the user and give them access to certain things based on their requirements.

## 5. SCOPE AND APPLICATIONS

The **Banking Management System (BMS)** is designed to streamline banking operations by automating account management, transactions, loans, and security mechanisms. The system aims to provide a secure, efficient, and user-friendly platform for both bank employees and customers. Some key functions of BMS are:

- **User Account Management:** Secure registration, authentication, and profile management for customers and administrators.
- **Transaction Processing:** Supports deposits, withdrawals, fund transfers, and automatic balance maintenance.
- **Loan and Credit Management:** Facilitates loan applications, interest calculation, and repayment tracking.
- **Security and Compliance:** Offers encryption, multi-factor authentication, and regulatory compliance features.
- **Data Analytics and Reporting:** Generates financial statements, transaction history, and predictive analytics for business planning.
- **Multi-Platform Accessibility** : Offers support for desktop banking applications and web applications.

The banking management system can be applied in various financial and banking environment including:

- Retails banks can use this system for managing customers account and processing transactions
- In government financial system for managing pension funds, subsidies, and public sector payment.
- Small microfinance institution for facilitate small scale lending and saving management.
- Corporate banks can use to assist businesses in managing payroll, fund transfer, and investments.

## 6. TIME ESTIMATION

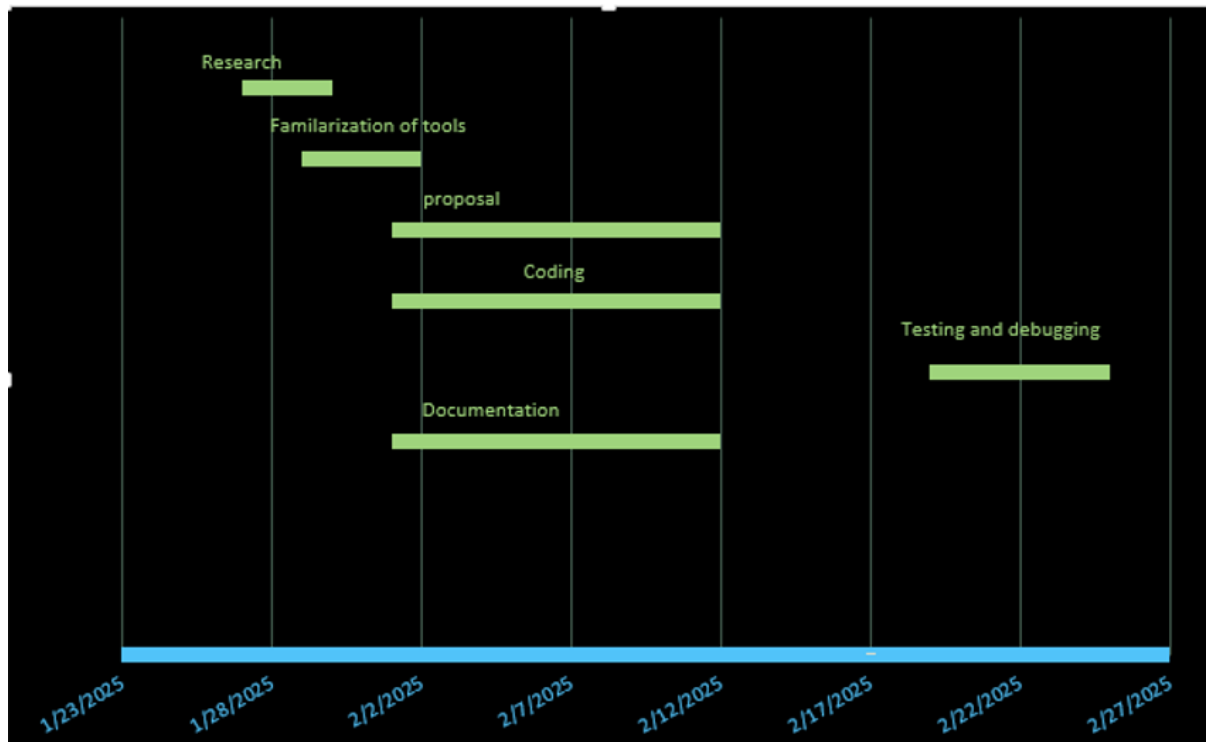


Figure 6.1 : Gantt Chart

## **7. FEASIBILITY ANALYSIS**

An initial investigation in a system that determines whether an alternative system is feasible. A system summarizing feasibility study that describes and evaluate candidate system and provides for the selection of best system that meets system performance requirements.

### **7.1 Technical Feasibility**

Since in technical feasibility we discuss about whether the required technology to run the system is available. System is technically feasible in the sense that it requires the computer which can support the system and internet and non-technical person can also use it without any difficulty.

### **7.2 Economic Feasibility**

One of the most important factors is to consider the monetary factor to check whether the system is economically feasible or not. Since this online system will eradicate the customer problem travelling cost. Customer will get their food sitting at their home. Since it is online system, it provides the long-term benefit to the organization. Thus, it seems to be economically feasible.

### **7.3 Operational Feasibility**

The new system is operationally feasible because it can be used effectively after it has been developed. Management can support this system and easy to operate and reduce the operating time considerably

### **. 7.4 Time Feasibility**

The new system is a bit time efficient and with a simple software yet complex structures it may be a hard jump to the smooth schedule of the the software but overall it can be a determining factor which proves to be time feasibility.

## **8. Requirements**

### 8.1 Hardware Requirement

- i. Intel i3 2.8 GHz Processor and above
- ii. RAM 2 GB and above
- iii. HDD 5GB Hard disc and above

### 8.2 Software Requirements

- i. WINDOWS OS (8 or higher) or Linux or MacOS
- ii. IDE
- iii. GCC-Compiler

## 9. References

- [1] C. R. Kothari, *Research Methodology: Methods and Techniques*, 4th ed. New Delhi, India: New Age International Publishers, 2019.
- [2] K. C. Laudon and J. P. Laudon, *Management Information Systems: Managing the Digital Firm*, 16th ed. Pearson Education, 2020.
- [3] E. Turban, C. Pollard, and G. Wood, *Information Technology for Management: On-Demand Strategies for Performance, Growth, and Sustainability*, 12th ed. Wiley, 2021.
- [4] GeeksforGeeks, “Banking System Database Management in SQL,” Available: <https://www.geeksforgeeks.org/banking-system-database-management-in-sql/>.
- [5] ResearchGate, “Banking Management System Projects and Analysis,” Available: <https://www.researchgate.net/>.
- [6] IEEE Xplore, “Banking Systems and Financial Technologies,” Available: <https://ieeexplore.ieee.org/>.
- [7] Federal Reserve, “Banking System Regulations,” Available: <https://www.federalreserve.gov/>.