

**PURBANCHAL UNIVERSITY**

GOMENDRA MULTIPLE COLLEGE

Birtamode-4, Jhapa

BANK MANAGEMENT SYSTEM

Project Proposal for BCA 2nd Semester

**Submitted By:**

Prabin Karki ()

Sujan Rai()

Saurav Kumar Uprety()

**Under Keen Supervision of:**

Mr. Jhalnath Chapagain

**ABSTRACT**

The "Bank Management System" is a program designed to provide a better way to manage detailed information about bank accounts and transactions. The main aim of this project is to develop software for financial institutions to handle customer accounts efficiently. This program provides an easy and reliable way of working with customer data, account balances, and transaction histories. The software helps access their information easily, and it can be updated and maintained by the admin of the system. This ensures that the data remains up- to-date and user-friendly, making the software easier to use. It reduces the possibility of errors during calculations and provides a reliable method for handling financial transactions. The Bank Management System can be an essential tool for banks and financial institutions.

The system has been designed in a simple manner so that users won’t face any difficulties while using the program. In the past, banks recorded customer data and transactions manually, which led to inefficiencies and errors. The main problem with manual systems is that data cannot be updated, deleted, or retrieved systematically, making the process complex and time- consuming. It is also challenging to search for specific customer data quickly. This system will help banks maintain records in a systematic and organized manner.

Table of Contents

[Chapter 1 | Introduction 1](#_bookmark0)

* 1. [Problem Statement 1](#_bookmark1)
  2. [Bank Management System 1](#_bookmark2)
  3. [Project Scope 1](#_bookmark3)
  4. [Key features 2](#_bookmark4)

[User-Friendly Interface 2](#_bookmark5)

[Data Management 2](#_bookmark6)

[Data Security 2](#_bookmark7)

* + 1. [System Design and Methodology 3](#_bookmark8)
       1. [System Design 3](#_bookmark9)
          1. [User Interface Design 3](#_bookmark10)
          2. [Database Design 3](#_bookmark11)
       2. [Methodology 3](#_bookmark12)
          1. [Requirements Gathering 4](#_bookmark13)
          2. [System Development Process 4](#_bookmark14)

[Chapter 3 | Functional Implementation 5](#_bookmark15)

* 1. [Account Management 5](#_bookmark16)
  2. [Functional requirements 5](#_bookmark17)
     1. [Create Account 5](#_bookmark18)
     2. [Delete Account 5](#_bookmark19)
     3. [Transaction Management 5](#_bookmark20)
     4. [Deposit Funds 5](#_bookmark21)
     5. [Withdraw Funds 5](#_bookmark22)
     6. [Check Balance 6](#_bookmark23)
     7. [Display All Accounts 6](#_bookmark24)

[Chapter 4 | System Analysis and Requirements 7](#_bookmark25)

* 1. [Functional Requirements 7](#_bookmark26)
  2. [Software Requirements 7](#_bookmark27)
  3. [Hardware Requirements 7](#_bookmark28)

[Chapter 5 | Conclusion 8](#_bookmark29)

[Chapter 6 |References 9](#_bookmark30)

# Chapter 1 | Introduction

## Problem Statement

The Bank Management System is designed to address the inefficiencies and challenges of traditional banking processes. Manual systems often lead to human errors, slow transaction processing, and difficulty in managing a large number of customer accounts and financial records. These problems result in delays, inaccuracies, and reduced customer satisfaction. The system aims to automate core banking functions, reducing errors, increasing efficiency, and providing a more reliable banking experience for both staff and customers.

## Bank Management System

The Bank Management System is a software solution that automates and manages banking activities such as account creation, transaction processing and balance inquiries. It provides a centralized platform where both bank employees and customers can interact with banking services. The system ensures that transactions are processed securely and efficiently, data is stored systematically, and banking operations are streamlined to save time and resources.

## Project Scope

The scope of the Bank Management system includes:

* + - To manage user Account
    - Easy to navigate accessing and updating Bank information
    - Works can be more efficiently and accurately
    - Can add more functionality according to user needs
    - To get rid of maintaining manual books
    - Data privacy and security

## Key features

### User-Friendly Interface

As we make the code easier to preview and update for future updates to new functionality, we try to make the interface with an easy structure that can be easily recognized by the user’s selection. As we understand the user’s requirement and their way of understanding the system, we provide them with a smooth and reliable interface for easier and time-saving usage.

### Data Management

This program provides easy access to user data, allowing for efficient maintenance and manipulation. It enables users to store data in a well-organized manner, saving time when accessing, searching, and updating information. By building the program in structural way makes the data management more comfortable and effective.

## Data Security

The program offers strong security measures to protect and handle data effectively. Users can rely on the system to securely store their information, with easy options for data backup and protection.

# Chapter 2 | System Design and Methodology

## System Design

System design refers to the overall structure of the Bank Management System. It includes planning how different components of the system, such as the user interface and file management interact with each other to achieve smooth and efficient operation. The design ensures that the system is scalable, secure, and easy to use.

## User Interface Design

The user interface (UI) design focuses on creating an intuitive and easy-to-navigate platform for users. It involves designing clear input that allow users to perform tasks like account creation, transaction processing, and balance inquiries with minimal effort. The UI is designed to be user- friendly for both bank employees and customers, ensuring a seamless user experience.

## Database Design

In a Bank management system developed in **C++**, the database module plays a crucial role in storing and managing customer records, transactions, and account details. Since this system uses files instead of a traditional database management system (DBMS)**,** all data is stored in structured file formats such as text files or binary files**.** This approach ensures that data remains persistent even after the program is closed.

## Methodology

The methodology describes the approach used to develop the Bank Management System. It includes the different phases of the project, from gathering requirements to system deployment. A structured methodology ensures that the system is developed systematically, meeting all user needs and technical requirements.

## Requirements Gathering

Requirements gathering is the process of collecting information about what the system needs to do. This involves consulting with stakeholders such as bank staff and potential users to understand their expectations and specific needs. The collected requirements guide the design and development of the system to ensure it solves the identified problems effectively.

## System Development Process

The system development process includes the steps taken to build the system, such as system analysis, design, coding, testing, and implementation. During this process, the system is continuously tested and refined to ensure it meets all functional and non-functional requirements. The development process also involves documenting the system’s features and functionalities for future maintenance.

# Chapter 3 | Functional Implementation

## Account Management

Account management refers to the features of the system that handle customer accounts. It includes functionalities for creating, deleting, and managing account information. The system ensures that account data is stored securely and can be accessed or modified easily when needed.

## Functional requirements

* + 1. **Create Account**

The create account functionality allows users to input customer information, such as name, address, contact details, and an initial deposit amount, to open a new bank account. The system validates the entered data to ensure accuracy and completeness before proceeding. Once the information is confirmed, the system generates a unique account number, assigns it to the new customer, and securely stores all relevant details in the database. Additionally, a welcome email or SMS notification may be sent to the customer, confirming the successful creation of the account.

* + 1. **Delete Account**

The delete account feature allows users to remove an account from the system when a customer requests account closure. Before deletion, the system verifies that the account has no outstanding loans, pending transactions, or negative balances. If all criteria are met, the system securely removes the account details while maintaining necessary records for audit and compliance purposes. A confirmation message is displayed, and a closure receipt may be generated for the customer. Additionally, an email or SMS notification can be sent to the customer to confirm the successful closure of their account.

* + 1. **Transaction Management**

Transaction management is a critical component of the banking system, ensuring that all financial transactions, such as deposits, withdrawals, and fund transfers, are processed efficiently and accurately. The system enforces security checks, verifies account details, and updates account balances in real-time. It also logs all transactions for auditing and reporting purposes. Customers and bank staff can track transaction history, generate statements, and receive alerts for any suspicious or unauthorized activities. To enhance security, the system may implement multi-factor authentication for high-value transactions.

* + 1. **Deposit Funds**

The deposit funds feature enables users to add money to a customer’s account through various channels, such as cash deposits, online transfers, and mobile banking. The system verifies the deposit amount, updates the account balance in real-time, and generates a receipt for the transaction. Customers may receive a confirmation SMS or email upon successful deposit. Additionally, the system maintains a detailed log of all deposit transactions, allowing customers to view their deposit history and reconcile their account balances as needed.

* + 1. **Withdraw Funds**

The withdraw funds functionality enables users to take money out of an account through cash withdrawals at a bank branch, ATMs, or online fund transfers. Before processing a withdrawal, the system checks if the account has sufficient funds to prevent overdrafts. If funds are available, the system deducts the specified amount, updates the balance in real-time, and generates a withdrawal receipt. Customers may receive a notification confirming the transaction. Security measures, such as OTP verification or biometric authentication, can be implemented to prevent unauthorized withdrawals.

# 

# Chapter 4 | System Analysis and Requirements

## Functional Requirements

Functional requirements describe the specific features and actions the system must perform. This includes creating and managing accounts, processing deposits and withdrawals, checking balances, and generating transaction reports. These requirements ensure that the system fulfills its intended purpose.

## Software Requirements

* + - Visual Studio Code as the main development environment.
    - MinGW for compiling C++ code.
    - C/C++ extension for IntelliSense, debugging, and syntax highlighting.
    - File library for file handling and data management.
    - Git for version control and project tracking.

## Hardware Requirements

* + - A multi-core processor for efficient compilation and execution.
    - At least 8GB RAM for smooth performance in VS Code.
    - SSD storage for faster file operations and data access.
    - Sufficient disk space for storing and managing files used as a database.

# Chapter 5 | Conclusion

The conclusion summarizes the entire project, highlighting the main objectives, development process, and outcomes. It reflects on how the Bank Management System successfully addresses the problems identified in the problem statement, improves the efficiency of banking operations, and enhances the user experience. The conclusion may also discuss any challenges faced during development and potential future improvements to the system. Throughout the development process, various challenges were encountered, including ensuring data security, maintaining real-time transaction updates, and optimizing the system for high performance. However, these challenges were addressed through robust database management, secure authentication protocols, and efficient system design.

Looking ahead, future improvements may include integrating AI-driven customer support, expanding multi-platform accessibility, and incorporating advanced analytics for fraud detection and financial forecasting. Continuous updates and enhancements will ensure that the Bank Management System remains efficient, user-friendly, and secure, adapting to evolving technological and banking industry requirements.

# Chapter 6 |References

Website: https://[www.w3schools.com](http://www.w3schools.com/) https://[www.programiz.com](http://www.programiz.com/) https://[www.learn-c.org,](http://www.learn-c.org/) YouTube Chanel:

[https://www.youtube.com/@ApnaCollegeOfficial](https://www.youtube.com/%40ApnaCollegeOfficial) [https://www.youtube.com/@CodeWithHarry](https://www.youtube.com/%40CodeWithHarry) [https://www.youtube.com/@TechnologyChannel](https://www.youtube.com/%40TechnologyChannel)