TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

DHARAN

BANK MANAGEMENT APP

A COURSE PROJECT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE PRACTICAL COURSE ON OBJECT ORIENTED PROGRAMMING [CT 451]

Submitted By:

Prasanna Pradhan(PUR081BCT048)
Nitesh Kumar Shah(PUR081BCT041)
Bishal Bista(PUR081BCT013)
Aryan Rijal(PUR081BCT009)

Submitted to:

Department of Electronics and Computer Engineering, Purwanchal Campus
Institute of Engineering, Tribhuvan University
Dharan, Nepal

Ashad,2081

Acknowledgment

We would like to express our sincere gratitude to Mr. **Tantra Nath Jha** for his invaluable guidance and support throughout our C++ project. His insightful suggestions, constructive feedback, and unwavering encouragement were instrumental in the successful completion of this project. We are deeply appreciative of the time and effort he dedicated to helping us understand complex concepts and navigate challenges. His expertise and mentorship have been truly inspiring, and we are grateful for his contributions to our learning journey.

Contents

Introduction

Banking operations today require efficient and secure software solutions to manage customer accounts, transactions, and history. This project focuses on building a simple Bank Management System in C++ that provides core banking functionalities such as account creation, deposits, withdrawals, transfers, and transaction history management.

Objective

- To develop a console-based bank management application.
- To provide both admin and user functionalities.
- To ensure secure password handling and maintain proper transaction records.
- To generate statements and histories for accountability.

Existing System

Traditional manual banking systems are:
- Time-consuming and prone to errors.
- Difficult to maintain detailed transaction records.
- Lacking automation for generating statements and histories.

Proposed System

The proposed system will:
- Allow admins to create, view, and delete accounts.
- Allow users to deposit, withdraw, transfer funds, and view transaction histories.
- Store all account details in CSV files for persistence.
- Provide automated statements and full transaction logs.
- Be lightweight, secure, and easily extendable.

Methodology

- Requirement Analysis: Identify user/admin needs and basic banking operations.
- System Design: Define data structures (Account, Transactions) and storage format (CSV).
- Implementation: Develop in C++ with modular functions for account handling, transactions, and history.
- Testing: Verify correctness of deposits, withdrawals, transfers, and file handling.
- Evaluation: Ensure usability and robustness.

Development Tools

- Language: C++
- Compiler/IDE: GCC / MinGW, Visual Studio Code
- File Handling: CSV-based storage for persistence
- Libraries: Standard C++ libraries (iostream, fstream, iomanip, etc.)

Development Process

1. Planning & Requirements Gathering
2. Design of Data Structures (Account, Transaction)
3. Implementation of Core Functions (Create, Deposit, Withdraw, Transfer, Logs)
4. Integration (Admin + User modules)
5. Testing and Debugging
6. Final Deployment & Documentation

Project Scope

- Suitable for small-scale banking operations or academic projects.
- Covers essential banking functionalities without requiring external databases.
- Can be expanded with features like interest calculation, loan management, or GUI in the future.

Project Schedule

Phase | Duration

Requirement Gathering | 2 days
System Design | 2 days
Implementation | 5 days
Testing & Debugging | 3 days
Documentation | 2 days

Timeline

- Week 1: Requirement analysis and system design
- Week 2: Core module implementation (Accounts, Transactions, Admin/User)
- Week 3: Testing, debugging, and refinement
- Week 4: Documentation and project submission