**Research Report: Bank Management System Development**

**--- By Taksham Sharma**

**Jims Sec 3**

**Aim**

The development of a robust Bank Management System (BMS) is essential for managing banking operations efficiently. This report delves into the conceptualization, design, and implementation of a BMS using C++. The system supports functionalities such as account creation, deposit, withdrawal, and balance display. Additionally, it incorporates features for both admin and user access, ensuring secure and streamlined banking processes.

**Introduction**

A Bank Management System (BMS) is a software application that automates the core banking functions. This project aims to build a simplified BMS using C++ programming, focusing on fundamental banking operations. The primary goal is to understand the underlying principles of bank management systems, including data handling, transaction management, and security measures.

**Objectives**

1. To design a system capable of creating, managing, and deleting bank accounts.
2. To implement transaction functionalities including deposits, withdrawals, and balance inquiries.
3. To ensure secure access through admin and user authentication.
4. To facilitate data persistence through file handling.

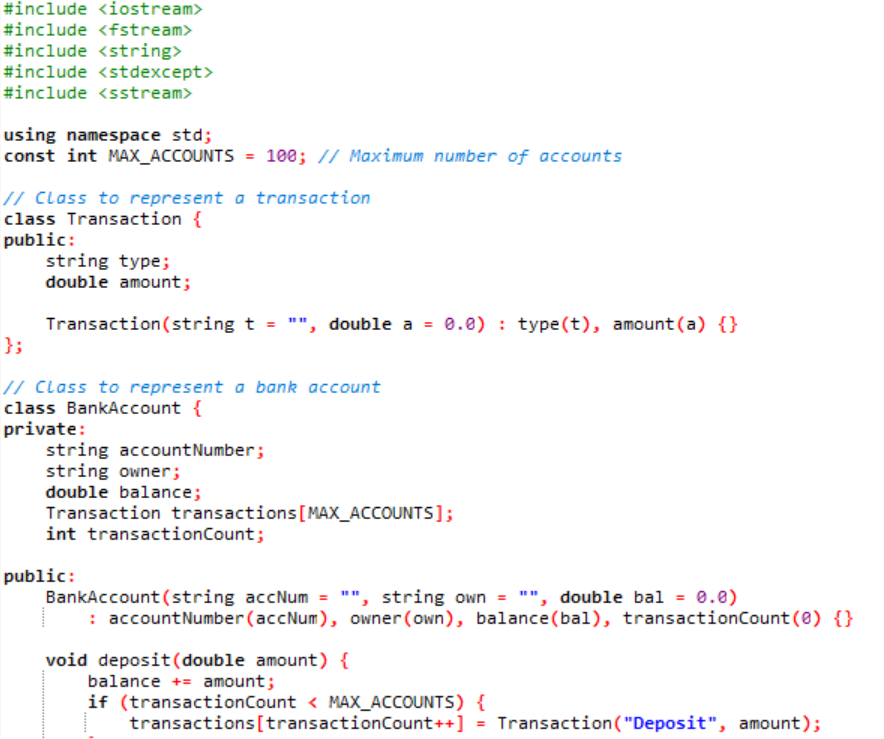
**System Design**

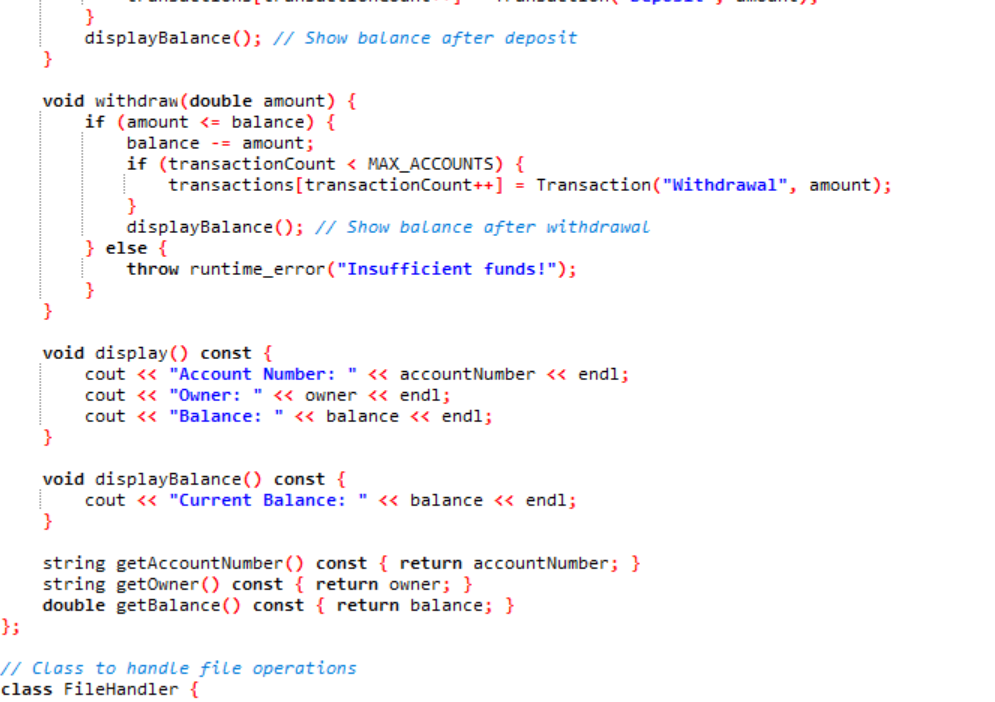
1. **Class Structure**:
   1. **Transaction Class**: Represents individual transactions with attributes for type and amount.
   2. **BankAccount Class**: Manages account details, including account number, owner, balance, and transactions.
   3. **FileHandler Class**: Handles file operations for saving and loading account data.
   4. **Admin and User Classes**: Provide interfaces for admin and user interactions.
2. **Functional Requirements**:
   1. Account creation with validation.
   2. Secure deposit and withdrawal operations.
   3. Balance display with real-time updates.
   4. Admin login with secure password authentication.
   5. Persistent storage of account data.

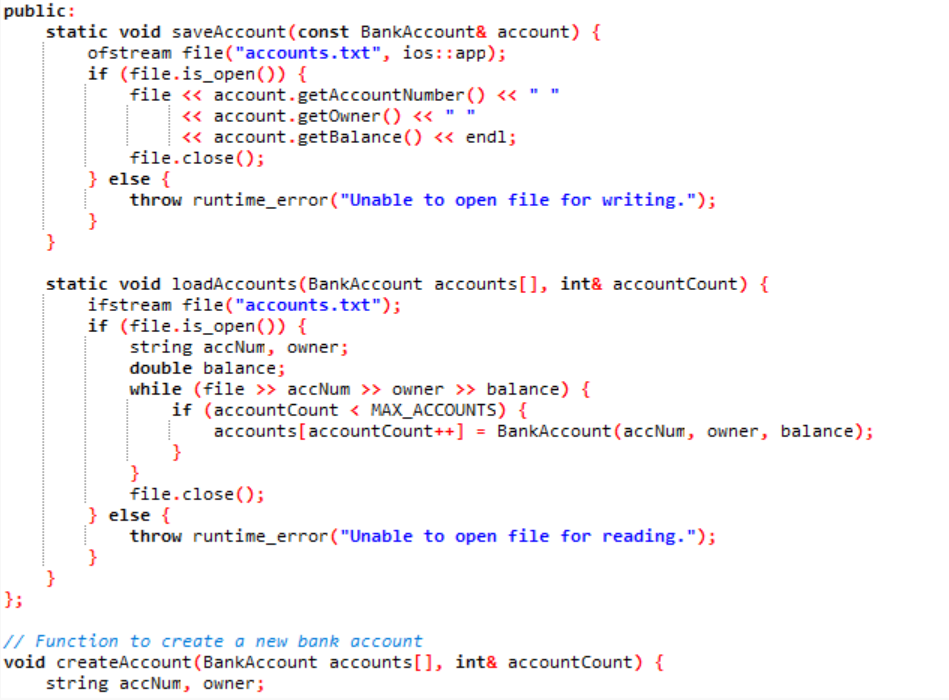
**Implementation**

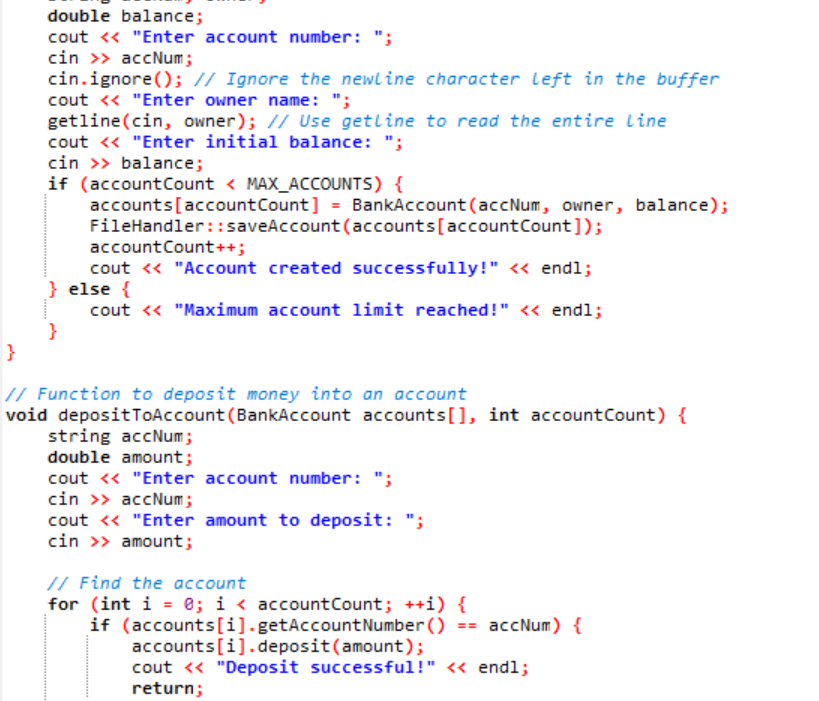
The implementation involves writing C++ code to define the classes and their methods. Key functionalities include:

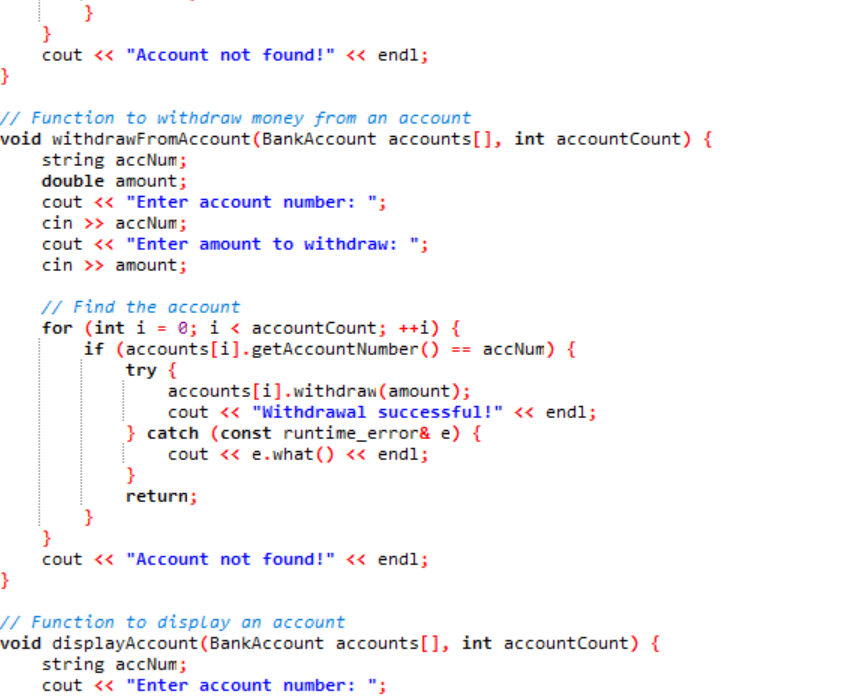
1. **Account Management**: Creating, displaying, and managing bank accounts.
2. **Transaction Processing**: Handling deposits and withdrawals securely.
3. **File Handling**: Storing and retrieving account data from text files.
4. **User Authentication**: Ensuring secure admin access with password protection.

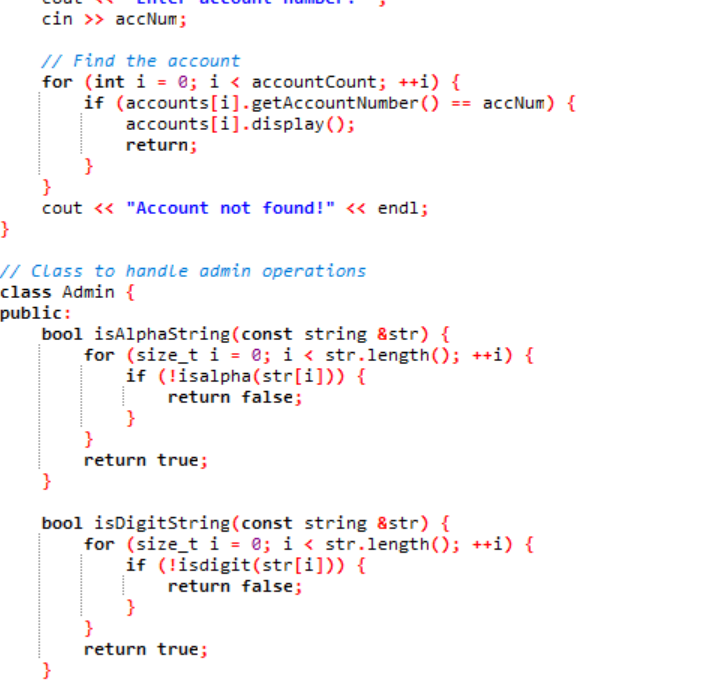
**Code**

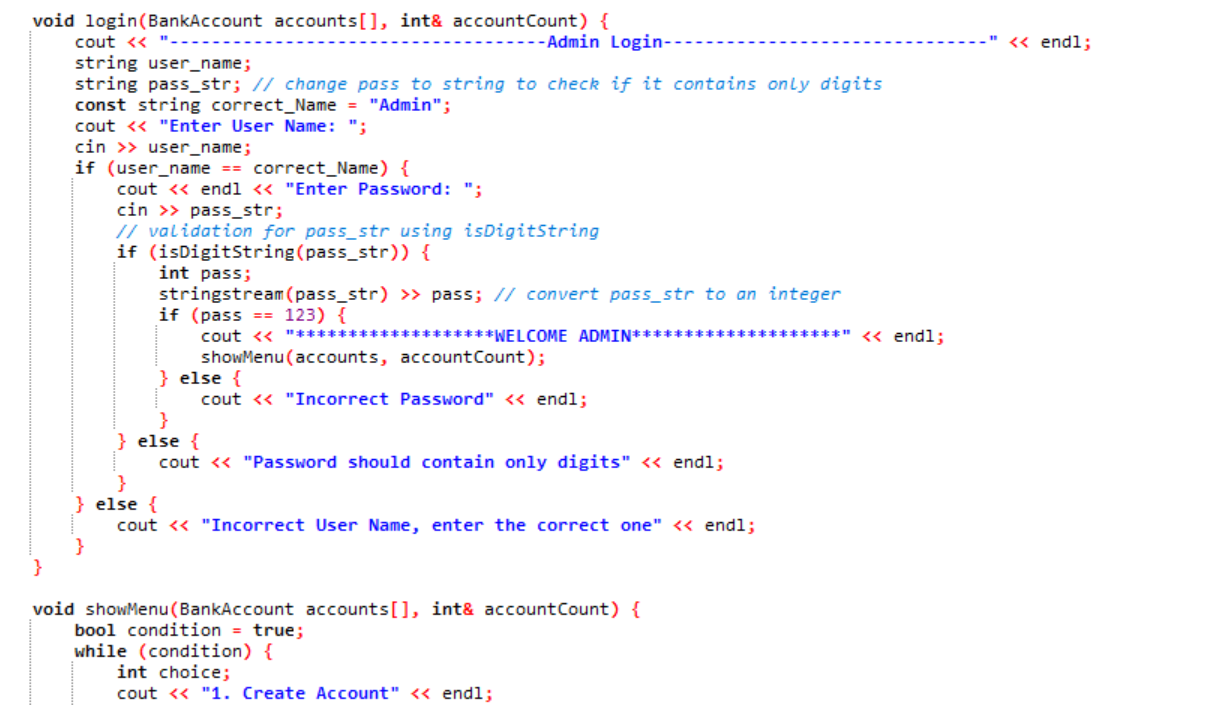


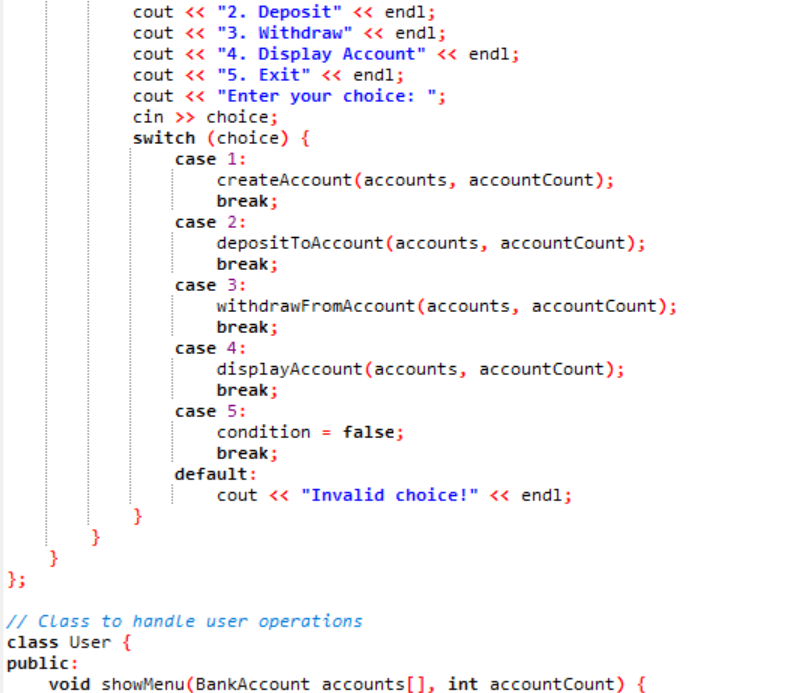


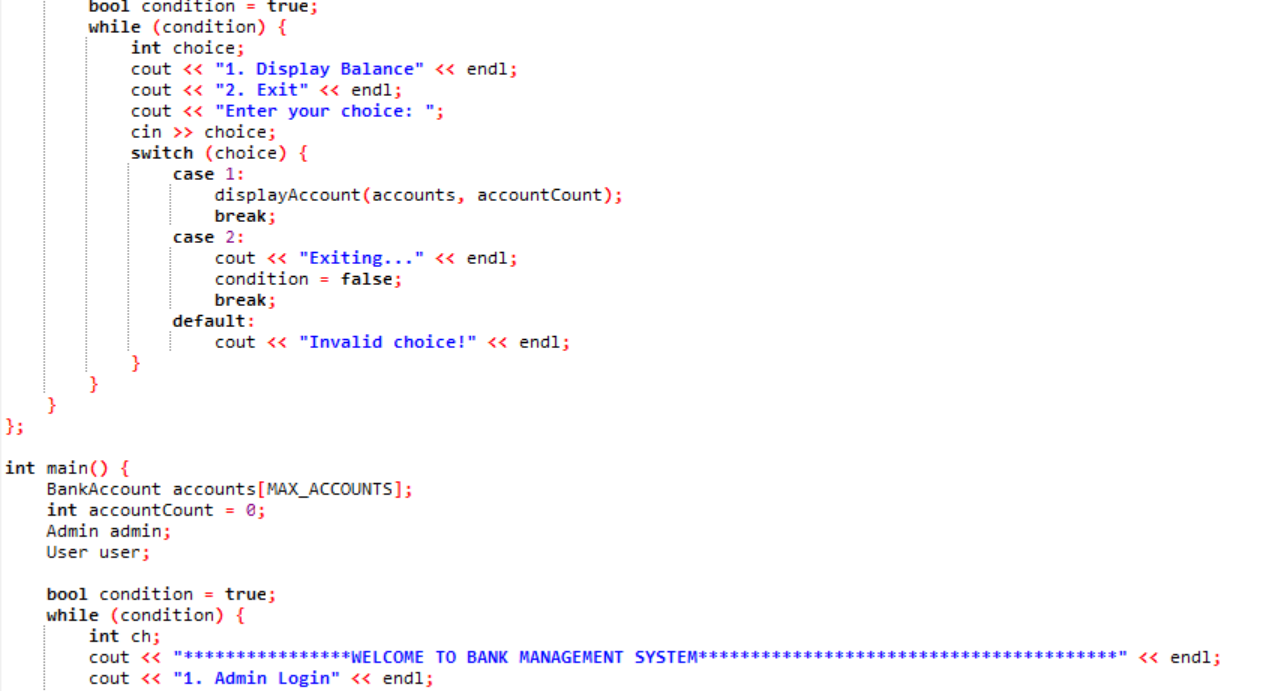


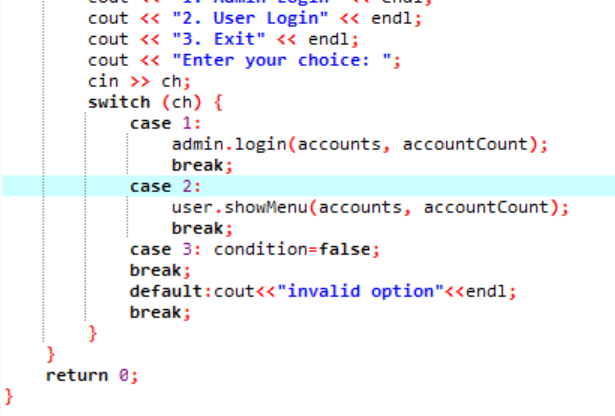


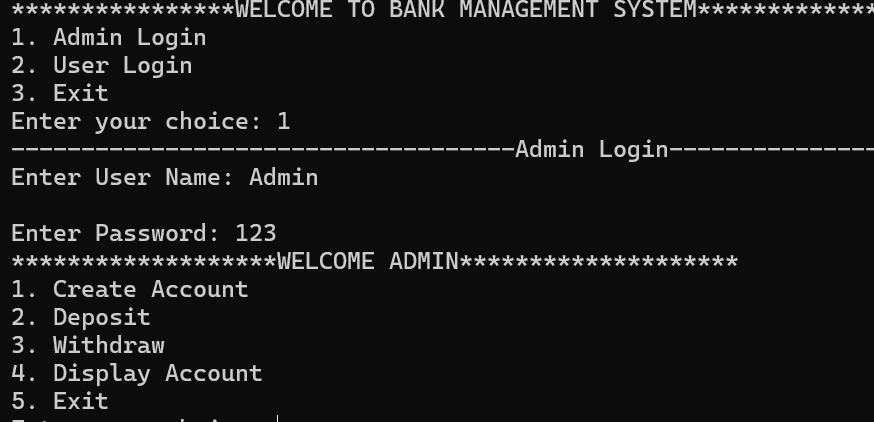


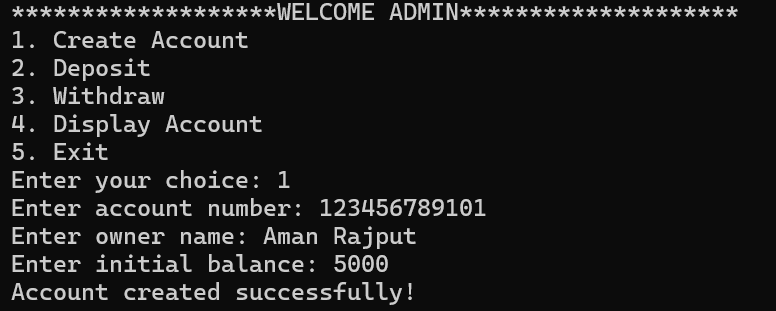


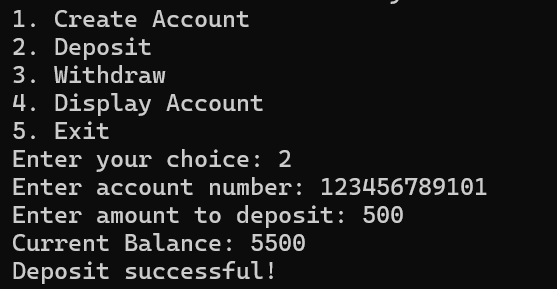




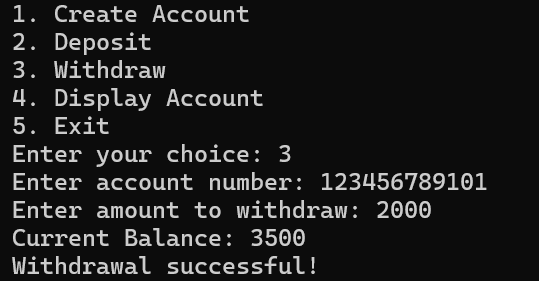


**Testing and validations**

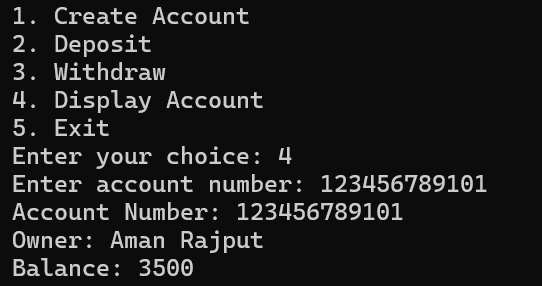
1.Admin menu with functions (creation of new Account)

2.Deposition of Amount

3.Withdrawal of Amount

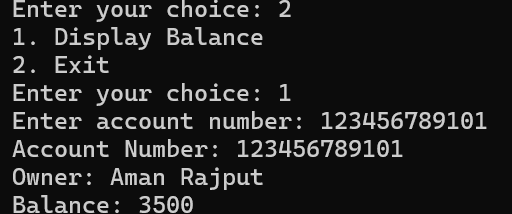


4.Displaying Account

nvjbjb

5. User Menu

6.Displaying User’s Balance



**Conclusion**

The project achieved its primary objectives of creating a functional Bank Management System using C++. The system handles account management, transactions, and secure admin access efficiently. Future enhancements could focus on improving security, scalability, and user interface. This project provides a foundational understanding of BMS development, which can be extended to more complex and feature-rich systems.

**References**

1. **C++ Programming Language**: Bjarne Stroustrup, Addison-Wesley Professional.
2. **Bank Management System Concepts**: Various online resources and tutorials on banking software development.
3. **File Handling in C++**: Documentation and examples from C++ reference sites.