Short Report on Banking System Code

Overview:

The program implements a console-based Banking System using C++. It supports:

- Account creation, viewing, deletion
- Deposits, withdrawals, and fund transfers
- Admin functionalities like listing users and checking logs

Files are used to persist user data across sessions.

Core Data Structure:

struct Account:

- accountNumber: Unique identifier for the account
- name: Name of the account holder
- email: Email ID
- contact: Phone number
- balance: Floating-point balance
- password: Used for login verification

Function-wise Breakdown:

main():

- Initializes the program and loads existing account data.
- Presents a menu to perform various banking operations.

createAccount():

- Registers a new account and stores user information in text and binary files.

- Adds the account number to the database and logs the event.

viewAccount():

- Verifies credentials and displays user profile and balance.
- Helps users verify their data and current financial status.

deposit():

- Allows users to add money to their account.
- Updates balance securely and logs the operation.

withdraw():

- Enables users to withdraw money.
- Ensures sufficient funds before processing.

transaction():

- Transfers money between accounts.
- Verifies both sender and receiver accounts and updates balances.

deleteAccount():

- Deletes the account after confirmation.
- Removes all related files and updates the database.

listAccounts():

- Admin feature to display basic details of all existing accounts.
- Does not show sensitive information like balance or password.

showLog():

- Admin feature to display all recorded logs of user activity.

Utility Functions:

- dynamicFileName(): Creates unique file names for user data.
- sensitiveContent(): Creates file names for sensitive data.
- validityCheck(): Validates account credentials before allowing access.

File Design Summary:

database.bin - Stores all account numbers (binary)

account_<id>.txt - Stores public user info (text)

sensitive_<id>.bin - Stores balance and password (binary)

log.txt - Records all activities (text)

Functionality of Key Features:

1. Account Creation:

- Ensures uniqueness of account number.
- Splits data into public (.txt) and sensitive (.bin) files.
- Initializes balance and logs the creation event.

2. Deposit and Withdraw:

- Ensures secure reading and updating of balance in binary file.
- Maintains transaction integrity using file rewinding and updating.

3. Transactions:

- Validates sender and receiver before transferring money.
- Handles insufficient fund checks and provides feedback.

- Logs every transaction with timestamps.

4. Data Security:

- Separates sensitive (password, balance) and public data.

- Requires correct password to access any account.

5. Admin Controls:

- Protects sensitive features (logs and user listing) using an admin password.

- Ensures only authorized personnel can view all user data or logs.

Conclusion:

This system demonstrates basic banking operations using file-based storage and user authentication.

It effectively applies modular programming and data validation.

While not suitable for real-world deployment due to lack of encryption and database, it is an excellent learning project.