



Student Name: _____

Reg. No.: _____

Due Date: Dec 28th, 2023

Instructor: Ali Imran Sandhu

- * This project will be assessed based on your technical report writing skills as well as a viva.
- * The grading will contribute 10% towards your overall course grade.
- * Students are encouraged to collaborate for understanding the project and course material, but it is not permissible for one student to help or be helped by another student in working through this course project.
- * Copied work, and/or unsatisfactory viva, will be penalized.!!!

For Instructor's Use Only

[illegible]

Implementation Guidelines

1. Maintain a modular approach in your analysis, ensuring that each component is well-documented for clarity and ease of understanding.
2. Include code snippets detailing the process, where necessary.
3. Ensure consistent formatting throughout the report, including font size, style, and line spacing.

Simulate The Banking System

Create a simulated banking system that utilizes various data structures and algorithms for efficient and secure operations. Provide clear documentation explaining the:

- Design choices
- Used data structures
- Implemented algorithms
- Appendix (Code file)

Also, include instructions for running the program and any dependencies. The various components and features that shall reflect in your implementation are:

1. Account Management System (Linked List, Hash Table)

1. Create a program to implement a linked list for managing customer information.
2. Define each node in the linked list to represent a customer, storing essential details like account number, name, address, and contact information. Create a menu for adding new customers and opening new accounts.
3. Utilize a hash table to efficiently manage customer accounts.
4. Define a mapping mechanism within the hash table to link account numbers to corresponding customer information, ensuring quick and efficient retrieval.

2. Transaction History (Queue)

1. Implement a queue for maintaining a transaction history for each account, offering deposit, withdrawal, and fund transfer functionalities.
2. Implement functionality to Enqueue new transactions and Dequeue older transactions, ensuring the transaction history is limited and efficiently managed.

3. Account Verification (Binary Search Tree)

1. Create a binary search tree for efficient verification of account details during login or transactions.
2. Explain how this binary search tree structure can be employed to quickly determine if an account exists and retrieve its associated information.

5. Testing

1. Thoroughly test the system to ensure the correct functioning of data structures and algorithms.
2. Conduct tests for edge cases and unexpected scenarios.