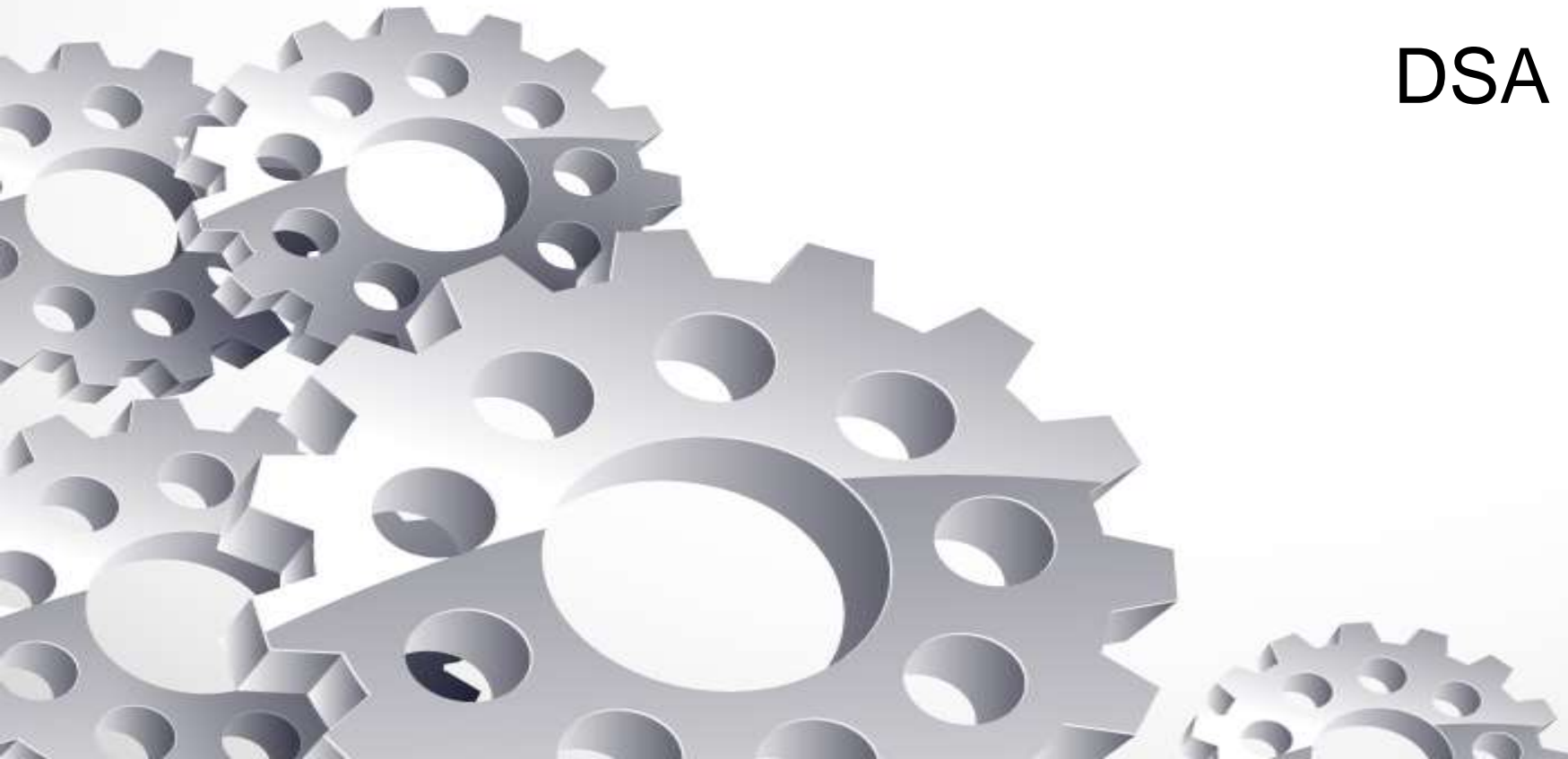


BANK MANAGEMENT SYSTEM

DSA MINI PROJECT





INTRODUCTION

This C++ bank management system leverages AVL trees for organized account data storage and linked lists for efficient transaction handling. The system offers admin and staff logins for account management and includes ATM services for customer transactions.

Functionalities:

Admin Access: Account creation, deletion, and display.

Staff Privileges: Account checks, cash deposits, and withdrawals.

ATM Services: Withdrawals, deposits, and balance inquiries for customers.

This system ensures secure, organized, and swift banking operations through AVL tree-based account structuring and linked lists for transaction handling, enabling a seamless banking experience for users.

OBJECTIVE



- 1. ***Efficient Account Management:*** Implement a system that efficiently manages customer accounts using AVL trees, ensuring quick access, insertion, deletion, and modification of account information.
- 2. ***Secure Data Handling:*** Employ data structures and algorithms that maintain the security and integrity of customer information, reducing the risk of unauthorized access or data breaches.
- 3. ***Admin and Staff Functionality:*** Provide distinct functionalities for administrators to handle account management tasks (creation, deletion, and display), and for staff to perform account checks, cash deposits, and withdrawals while ensuring proper security measures for each role.
- 5. ***ATM Services Integration:*** Develop an interface for ATM services, allowing customers to carry out essential transactions such as withdrawals, deposits, and balance inquiries securely and conveniently.
- 6. ***Scalability and Performance:*** Design the system with scalability in mind, ensuring it can handle a growing number of accounts and transactions without compromising performance.
- 7. ***User-Friendly Interface:*** Create an intuitive user interface for easy navigation, providing a seamless experience for both bank staff and customers.
- These objectives aim to create a robust, efficient, and secure bank management system that fulfills the needs of administrators, staff, and customers while ensuring data integrity and system reliability.

PROJECT OVERVIEW

FUNCTIONALITIES

1. ***Admin Operations:*** - Account Creation: Admins can create new customer accounts with essential details. - Account Deletion: Admins can remove customer accounts from the system. - Account Display: Ability for admins to view all stored customer accounts.
2. ***Staff Operations:*** - Account Checks: Staff can view customer account details, including balances and transaction history. - Cash Deposit: Staff members can process cash deposits into customer accounts. - Cash Withdrawal: Staff can facilitate cash withdrawals from customer accounts.
3. ***Account Management using AVL Trees:*** - Efficient storage and retrieval of customer account information using AVL trees. - Account Insertion and Deletion: Balancing operations to add or remove accounts.
4. ***Transaction Handling using Linked Lists:*** - Recording and managing individual account transactions via linked lists. - Maintaining a detailed transaction history for each customer account.
5. ***ATM Services:*** - Withdrawals: Allowing customers to withdraw cash securely from their accounts through ATMs. - Deposits: Enabling customers to deposit funds into their accounts using ATMs. - Balance Inquiry: Customers can check their account balance via ATM services.





- FLOW

The program starts with the main menu and navigates to different panels based on the user's input, allowing different functionalities related to banking operations.

This program follows a command-line interface structure and uses conditional statements to handle user choices. Each panel or section within the program provides specific functionalities related to account management and banking operations.

- UTILITY

provide the user a sublime interface to the user to manage banking operations. This project provides a foundation for a basic banking system in a simulated environment. It is designed to ensure an organized and systemic approach to implement a banking system

OVERVIEW OF DATA STRUCTURES



Data Structure used: AVL Tree

AVL trees in a banking system are fundamental for their ability to efficiently manage and process a significant number of account records while maintaining a balanced and consistent performance in search, insertion, and deletion operations. This structural efficiency is critical for ensuring a robust and responsive banking system, enabling swift and reliable access to account information.

By consistently providing a balanced structure, AVL trees secure predictable performance, making them suitable for applications, like banking systems, dealing with high volumes of dynamic data. This characteristic is crucial in preventing worst-case scenarios where unbalanced trees could result in significantly slower search times (with a time complexity of $O(n)$).

Data structures used

LINKED LIST

- In a banking system, a linked list is used to efficiently manage transaction logs for individual accounts. Each account has its own linked list to store transaction histories, allowing for easy and quick access to this chronological data. Linked lists enable straightforward addition and removal of transaction logs, providing a flexible structure for recording account activities



Operation used in the linked list:

TRAVERSAL

DESCRIPTION: Navigate through each node of the linked list from the starting point (head) to the end.

DELETION

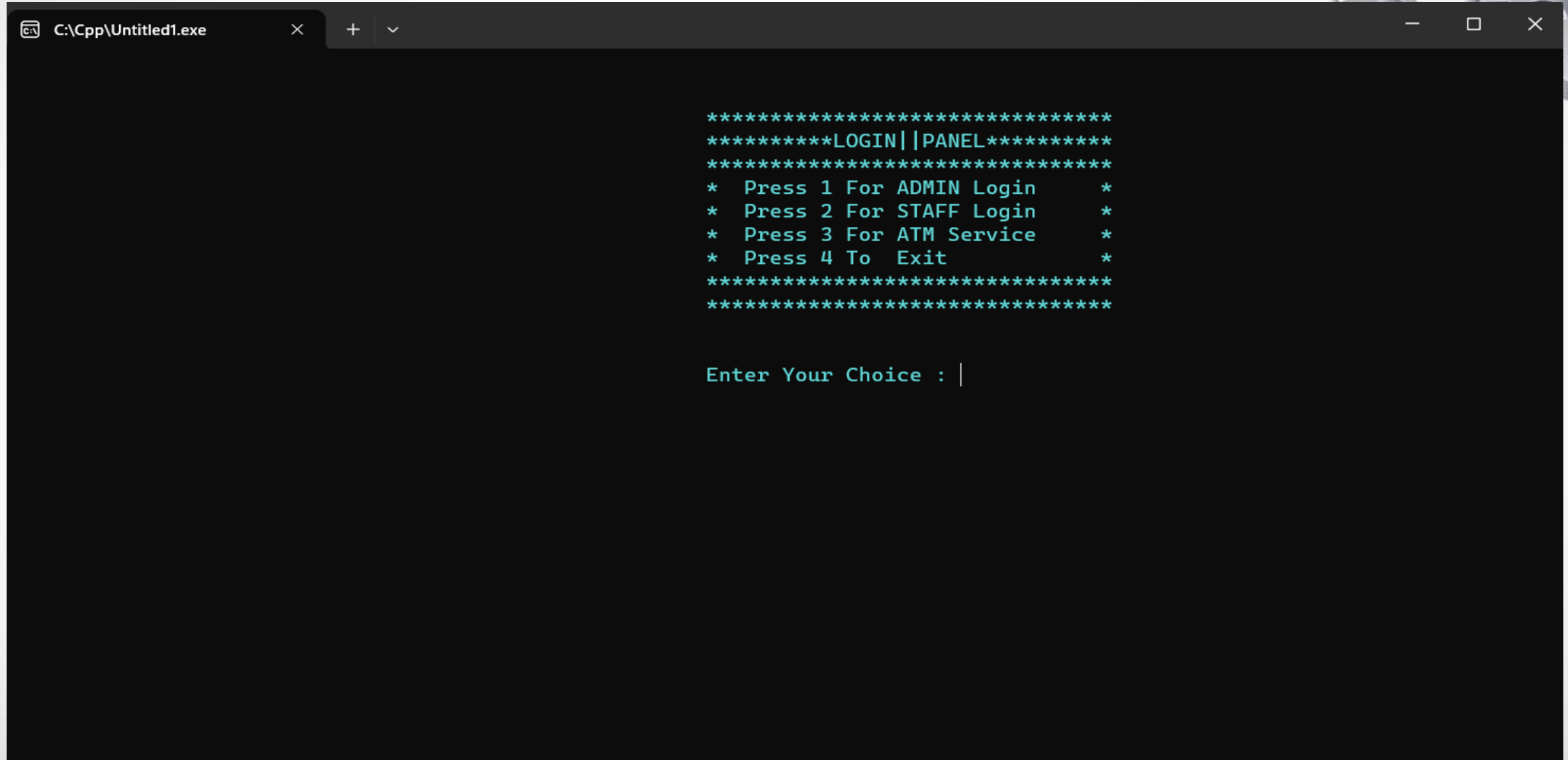
DESCRIPTION: Remove a node from the linked list. The node can be removed based on a specific condition or value.

INSERTION

DESCRIPTION: Add a new node to the linked list. Depending on the requirement, the node can be added at the beginning, end, or any specific position in the list.



OUTPUT:

A screenshot of a Windows-style application window titled "C:\Cpp\Untitled1.exe". The window has a dark background and displays a menu-driven program in a light green monospace font. The menu is enclosed in a rectangular border made of asterisks. The options are: "Press 1 For ADMIN Login", "Press 2 For STAFF Login", "Press 3 For ATM Service", and "Press 4 To Exit". Below the menu, the text "Enter Your Choice : " is followed by a vertical cursor bar, indicating the program is waiting for user input. The window's title bar includes standard minimize, maximize, and close buttons.

```
C:\Cpp\Untitled1.exe × + ∨  
  
*****  
*****LOGIN| |PANEL*****  
*****  
*   Press 1 For ADMIN Login   *  
*   Press 2 For STAFF Login   *  
*   Press 3 For ATM Service   *  
*   Press 4 To  Exit         *  
*****  
*****  
  
Enter Your Choice : |
```

C:\Cpp\Untitled1.exe



```
*****  
*****ADMIN||PANEL*****  
*****  
*   Press 1 To Create Account       *  
*   Press 2 To Delete Account       *  
*   Press 3 To Search Account       *  
*   Press 4 To Show List of All Acc *  
*   Press 5 To Edit Information      *  
*   Press 6 To Show Logs of Deleted Acc*  
*   Press 7 To Go Back to Main Menu *  
*****  
Enter an Option : |
```

```
*****  
*****STAFF || PANEL*****  
*****  
*   Press 1 To Check Account Info   *  
*   Press 2 To Deposit Cash         *  
*   Press 3 To With_Draw Cash       *  
*   Press 4 To Check Logs of User   *  
*   Press 5 To Go Back to Main Menu *  
*****  
Enter an Option : |
```

C:\Cpp\Untitled1.exe



```
*****  
*****ATM*****  
*****  
*   Press 1 To Deposit Cash           *  
*   Press 2 To Go Back to Main Menu   *  
*****  
Enter an Option : |
```

CONCLUSION

The bank management system designed using AVL trees and linked lists in C++ offers a robust and secure platform for efficient account management and transaction handling. Through the implementation of AVL trees for organized account storage and linked lists for detailed transaction recording, the system ensures quick access, balanced data storage, and accurate transaction history for each customer. With distinct functionalities for administrators and staff, encompassing account creation, deletion, checks, deposits, and withdrawals, the system prioritizes security and user-friendliness. Furthermore, the integration of ATM services provides customers with convenient and secure access to common banking operations. Overall, this system successfully streamlines banking activities, ensuring reliability, security, and ease of use for administrators, staff, and customers.





TEAM MEMBERS

Vishwajith V

RA2211003010081

Pranesh S

RA2211003010097

A S Arunththamizh

RA2211003010099

THANK YOU!

