**SACCO MANAGEMENT SYSTEM.**

I am creating a savings application for farmers in rural areas that is a SACCO management system Using entirely JAVA language. This application, one can save their money by making deposits, one can withdraw their savings and also receive financial statements for their transactions either deposits or withdrawals and even reports sent to their email. Because this application is going to be used by many people, one is required to signup and login to be able to create their own savings account, and access their data seperate from any other person saving their data on the applications. Traditional SACCO management systems rely on databases, but an optimized in-memory approach using data structures can improve transaction speed and efficiency. The challenge is to determine the most suitable data structure(s) to store and process farmers' transactions, ensuring fast deposits, withdrawals, and statement retrievals while maintaining data integrity and security.

**APPLICATION DESIGN.**

There is going to be a login page, then a signup page. For the login page, the farmers will be able to enter their accounts after they have created their accounts by signing up. When they login, they are taken to a dashboard which will display their current balance in their savings account There should be a side navigation bar that should enable the user navigate between pages. In this navigation bar, there should be the user profile button which when clicked, takes you to the user page whre is shows the users Bio data and account details. Then there should also be an instructions button which when clicked, takes the user to the instruction page that contains instructions of how the user can maneuver through the application. There should be the withdraw button also which when clicked, takes the suer to the withdraw page where the withdraws happen. There should be a deposit button which when clicked also takes the user to the deposit page. There should be a Trasactions button which when clicked, takes the user to the transactions page which contains financial statements of both deposits and withdrawals. Here, farmers can filter the statements by date, and this page should also be paginated.

**SYSTEM DESIGN AND IMPLEMENTATION.**

**1) Deposit Money (Efficient Storage and Retrieval)**

● Farmers should be able to deposit money into their SACCO accounts.

● Each deposit should store farmer ID, amount, and timestamp.

● The data structure should allow efficient insertion and retrieval of deposits.

● Recommended Data Structures: ○ Linked List (for maintaining an ordered list of deposits) ○ Hash Map (for quick lookup of a farmer’s deposit history)

**2). Withdraw Money (Efficient Balance Management)**

● Farmers can withdraw money if they have sufficient balance.

● The system should check account balance before withdrawal.

● Each withdrawal should store farmer ID, amount, and timestamp.

● Recommended Data Structures: ○ Hash Maps (for storing and updating farmer balances in constant time) ○ Stack (for retrieving the most recent transactions first)

**3. Retrieve Account Statement (Reverse Order Retrieval)**

● Farmers should be able to request a statement of their last N transactions.

● The system should return transactions in reverse chronological order (most recent first).

● Farmers can filter statements by date range.

● Recommended Data Structures: ○ Stack (Last-In-First-Out for quick retrieval of recent transactions) ○ Queue (First-In-First-Out if statements need to be processed in order)

**4. Check Balance**

● Retrieve the farmer's balance from the hash map and display it.

**PROJECT OBJECTIVES**  
Analyze different data structures (linked lists, stacks, queues, hash maps, trees) to determine their efficiency in managing SACCO transactions.

2. Implement a data structure-based transaction system that supports deposits, withdrawals, and statement retrieval efficiently.

3. Optimize retrieval speed for recent transactions using appropriate in-memory data structures.

4. Compare and contrast the performance of different data structures in storing and retrieving SACCO transactions.