

Accounting System Documentation

This document provides documentation for the Accounting System application. The application is organized into several modules, each responsible for specific functionalities. The main components of the application include users, accounts, and transactions.

Table of Contents

1. Introduction
 2. Main Module (main.c)
 3. User Module (users.c)
 4. Transaction Module (transactions.c)
 5. Account Module (accounts.c)
-

1. Introduction

The Accounting System is designed to provide users with the ability to manage their accounts, record transactions, and generate financial reports. The application is structured into different modules to ensure modularity and ease of maintenance.

2. Main Module (main.c)

2.1 main()

The `main` function is the entry point of the application. It handles user sign-in, sign-up, and system exit. The user is presented with a menu offering options such as signing in, changing credentials, or exiting the system.

2.2 mainMenu()

The `mainMenu` function displays the main menu options for the user, including signing in, changing credentials, and exiting the system.

2.3 reportsMenu()

The `reportsMenu` function displays the financial reporting menu options, including account statements, transaction registers, and expense reports.

2.4 printTransactionsToFile(User user)

This function prints transactions of a user to a file named “transaction_register.txt.”

2.5 printAccountStatements(User user)

This function prints account statements to a file named “statements.txt.”

2.6 financialReporting(User user)

Handles financial reporting options based on user input.

2.7 bank(User user)

Manages user interactions within the banking system, including account management and transactions.

3. User Module (users.c)

3.1 userMenu()

Displays the user menu options for managing accounts, recording transactions, generating reports, and logging out.

3.2 signUp(User user)

Signs up a new user by appending their username and password to the USERS-FILE.

3.3 signIn(User user)

Signs in a user by comparing the provided username and password with those in the USERSFILE.

4. Transaction Module (transactions.c)

4.1 isValidDate(const char *dateStr)

Checks the validity of a date.

4.2 recordTransaction(Account *account, Transaction transaction)

Records a transaction for the specified account.

4.3 createTransaction(User *user)

Creates a new transaction for a user's account.

4.4 writeTransactions(User user)

Writes user transactions to a file named "transactions.txt."

5. Account Module (accounts.c)

5.1 manageAccounts(User user)

Manages user accounts by providing options to create, update, and delete accounts.

5.2 accountsMenu()

Displays the accounts menu with options for account management.

5.3 saveAccountsToFile(User user)

Saves user accounts to a file named “accounts.txt.”

5.4 createAccount(User *user)

Creates a new account for the user.

5.5 loadAccounts(User *user)

Loads user accounts from the “accounts.txt” file.

5.6 printAccounts(User user)

Prints the user’s accounts and returns the number of accounts.

5.7 deleteAccount(char *accountName, User *user)

Deletes an account with the given name.

5.8 updateAccount(char *accountName, User *user)

Updates an account with the given name.
