FACULTY OF COMPUTER SCIENCE

UNIVERSITI TEKNOLOGI MALAYSIA

DATA STRUCTURE & ALGORITHM - SECJ 2013 Section 02

PROJECT REPORT

TITLE : BANKING TRANSACTION SYSTEM

PREPARED BY : DACCrew

NAME	MATRIC NO
CHAI YU TONG	**A22EC0145**
NUR FARAH ADIBAH BINTI IDRIS	**A22EC0245**
WONG QIAO YING	**A22EC0118**

PREPARED FOR:

DR. LIZAWATI BINTI MI YUSUF

Table of Contents

Part 1: Introduction

Objective
- To manage transactions such as withdrawals, deposits, and transfers
- To implement the stack data structure
- To improve knowledge retention through applying

Synopsis

Introducing the DACCrew Banking Management System for Transaction Management
Our banking system is designed to streamline and manage basic transactions such as deposits, withdrawals, and transfers. By utilizing stack data structures, our project ensures efficient transaction handling. The system is user-friendly and boasts five main functions: "Check Balance," "Display Transaction Limit," "Perform Transaction," "Search for Transactions," and "Exit."

Features and Functions:

1. **Welcome Message and Menu Display:**
   Upon initiation, the system greets the user with a welcome message and presents a main menu with multiple functions available.

2. **Check Balance:**
   This feature allows users to check their account's current balance, and the system displays will display the current balance.

3. **Display Transaction Limit:**
   Users can review their transaction history with this feature, which lists all previously performed transactions.

4. **Perform Transaction:**
   When users select this option, the system prompts them to provide transaction details, including the date in "DD-MM-YYYY" format, transaction type (W for withdrawal, D for deposit, and T for transfer), and transaction amount.

5. **Search for Transactions:**
   This feature enables users to search for transactions using two categories: transaction type and date.

6. **File Saving:**
   The system can save the transaction history to a file named "transaction_history.txt," ensuring the persistence of transaction data for future reference.

7. **Exit:**
    Allows the user to exit the banking system.

Part 2: System Design

System algorithm design (pseudocode)
The pseudocode below will show the flow of our system:

1. Start
2. do
3. Display menu
4. User input choice
5. if (choice == 1)
   5.1 System display balance
6. else if (choice == 2)
   6.1 Print transaction list
7. Else if (choice == 3)
   7.1 System performs the transaction
   7.2 User enter transaction type
   7.3 if(type == 'D')
         7.3.1 User input amount
         7.3.2 Update balance and push transaction into the stack
   7.4 else if(type == 'W')
         7.4.1 User input amount
         7.4.2 if(amount<=balance)
               7.4.2.1 Update balance and push transaction into the stack
         7.4.3 else
               7.4.3.1 Print "insufficient amount for withdrawal"
   7.5 else if (type == 'T')
         7.5.1 User input amount
         7.5.2 if (amount<=balance)
               7.4.2.1 Update balance and push transaction into the stack
         7.5.3 else
               7.4.3.1 Print "insufficient amount for withdrawal"
   7.6 else
         7.6.1 Print "Invalid transaction type"
8. Else if (choice == 5)
   8.1 Exit program
9. Else
   9.1 Print "Invalid choice. Try again."
10. while(choice!=5)
11. End

Data Structure Operation and Implementation

For our project is about the banking system, we adopted the use of stack in our program as stack offers several advantages, mainly on its principle that is LIFO(Last In, First Out) principle that is suitable for the user to view the latest transaction and the balance over time as the latest transaction will be at the top of the list. We use a linked list stack as it will be more flexible in scenarios where the number of transactions is uncertain.

Stack Implementation:
- nodeStack class
  - represents a node in the stack, in which each node will contain information on a single transaction, including the date, amount, balance and type.
  - Have getter functions that are getDate(), getBalance(), getAmount() and getType() that return the date, balance, amount and type of transaction.

- Stack class
  - The banking transaction will be pushed onto the stack whenever a user performs a transaction.

  In the public:
  - push(string d, string t, double a, double b): push the transaction into the stack
  - searchByDate(string searchDate): search the transaction by date
  - searchByType(string searchType): search the transaction by transaction type

To conclude, stacks are used in the banking system to manage and track transactions in Last In, First Out order. Stack implementation will show the order of transactions where the latest one will be at the top of the list.

Part 3: User guide

The following figures show the complete interface of the DACCrew Banking Management System.

```
<<<<< WELCOME TO DACCrew BANKING MANAGEMENT SYSTEM >>>>>

1. Check Balance
2. Display Transaction List
3. Perform Transaction
4. Search for Transaction
5. Exit
Your choice:
```

Interface 1: Main Menu

In this main menu, the user needs to insert an integer value in between 1 and 5 to perform particular operations.

```
Current Balance: RM0

1. Check Balance
2. Display Transaction List
3. Perform Transaction
4. Search for Transaction
5. Exit
Your choice:
```

Interface 2: Check Balance Interface

Within this interface, the system displays the current account balance. As there have been no transactions recorded, the balance currently stands at zero. Additionally, if no exit option is selected, the system will continue to prompt the user for their next action.

```
----------------------------------------------------------
| Date          | Type        | Amount   | Balance  |
----------------------------------------------------------
1. Check Balance
2. Display Transaction List
3. Perform Transaction
4. Search for Transaction
5. Exit
Your choice: |
```

Interface 3: Display Transaction List
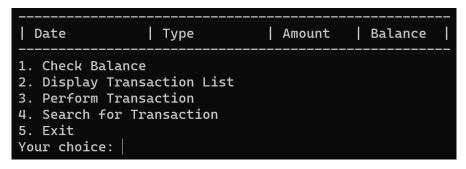
In this interface, the system shows the history of the transaction. Since there is no transactions done, only the header is being displayed.

```
Enter date (DD-MM-YYYY): 17-01-2024
Enter transaction type (D: Deposit/W: Withdraw/T: Transfer): D
Enter deposit amount: 200.00

Transaction history saved to 'transaction_history.txt'.

Transaction completed successfully.
```

Interface 4: Perform Transaction

In this interface, the system will prompt the user to enter the transaction details. Transactions can be initiated in the form of withdrawals, deposits, or transfers. Upon completion, all transaction results will be saved in a file titled "transaction_history.txt".

Date	Type	Amount	Balance
19-01-2024	Deposit	33	577
18-01-2024	Transfer	288	544
17-01-2024	Withdrawal	44	832
16-01-2024	Deposit	350	876
15-01-2024	Withdrawal	77	526
15-01-2024	Transfer	60	603
14-01-2024	Withdrawal	88	663
13-01-2024	Deposit	400	751
12-01-2024	Transfer	50	351

Interface 5: Transaction List

This interface displays the transaction history of the user. The transactions are saved using the stack concept, with the most recent transaction always appearing at the top of the list.

```
1. Check Balance
2. Display Transaction List
3. Perform Transaction
4. Search for Transaction
5. Exit
Your choice: 4

[1] Search by Type
[2] Search by Date
[3] Back to Main Menu
```

Interface 6: Search for Transaction

This interface provides the option to search for transactions based on either type or date.

```
Enter transaction type (D: Deposit/W: Withdraw/T: Transfer): D
-------------------------------------------------------------
| Date         | Type       | Amount    | Balance   |
-------------------------------------------------------------
| 19-01-2024   | Deposit    | 33        | 577       |
-------------------------------------------------------------

| 16-01-2024   | Deposit    | 350       | 876       |
-------------------------------------------------------------

| 13-01-2024   | Deposit    | 400       | 751       |
-------------------------------------------------------------

| 10-01-2024   | Deposit    | 500       | 500       |
-------------------------------------------------------------
```

Interface 7: Search for Transaction by Type (D: Deposit)

```
------------------------------------------------------------------
| Date          | Type          | Amount    | Balance    |
------------------------------------------------------------------
| 17-01-2024    | Withdrawal    | 44        | 832        |
------------------------------------------------------------------

| 15-01-2024    | Withdrawal    | 77        | 526        |
------------------------------------------------------------------

| 14-01-2024    | Withdrawal    | 88        | 663        |
------------------------------------------------------------------

| 11-01-2024    | Withdrawal    | 99        | 401        |
------------------------------------------------------------------
```

Interface 8: Search for Transaction by Type (W: Withdrawal)

```
------------------------------------------------------------------
| Date          | Type          | Amount    | Balance    |
------------------------------------------------------------------
| 18-01-2024    | Transfer      | 288       | 544        |
------------------------------------------------------------------

| 15-01-2024    | Transfer      | 60        | 603        |
------------------------------------------------------------------

| 12-01-2024    | Transfer      | 50        | 351        |
------------------------------------------------------------------
```

Interface 9: Search for Transaction by Type (T: Transfer)

```
------------------------------------------------------------------
| Date          | Type          | Amount    | Balance    |
------------------------------------------------------------------
| 15-01-2024    | Withdrawal    | 77        | 526        |
------------------------------------------------------------------

| 15-01-2024    | Transfer      | 60        | 603        |
------------------------------------------------------------------
```

Interface 10: Search for Transaction by Date (15-01-2024)

Interface 11: Check Balance

The system will automatically consider all transactions made and calculate the current balance.

Interface 12: Exit