



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTER SCIENCE

UNIVERSITI TEKNOLOGI MALAYSIA

DATA STRUCTURE & ALGORITHM - SECJ 2013 Section 02

ASSIGNMENT 2 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

Table of Contents	2
Part 1: Introduction	3
1.1 Objective	3
1.2 Synopsis	3
Part 2: System Design	4-5
2.1 System Algorithm Design	4-5
2.3 Data Structure Operation	5

PART 1: INTRODUCTION

1.1 Objectives

- Develop a simple bank account management system by applying the linked list concept.
- Perform some techniques in manipulating the data such as insertion, deletion, searching, and sorting within the system.
- Apply the concepts of encapsulation and data hiding when deploying a class in our system.

1.2 Synopsis

In this assignment, a simple bank account management system will be designed by applying the linked list concept as our main focus. There will be four functions available in this particular system. Firstly, the user is able to enter the account information into this system as a node regardless of the position in the linked list. To do so, the attributes, which are account name, account number, IC number, and account balance must be provided by the user during each insertion. Next, the accounts within the list can be removed when the user inputs the position of the node(account) that will be deleted into the system. Thirdly, the search function is designed to identify the existing account in the list. The system will return and display the account if it does exist and return an error message otherwise. Finally, the system can sort the accounts in the list in ascending order based on the users' choice of attributes.

PART 2: SYSTEM DESIGN

2.1 System Algorithm Design

1. Start.
2. Display menu.
3. User input choice
4. While(choice!=12)
 - 4.1 If choice == 1
 - 4.1.1 User key in the data
 - 4.1.2 Insert node at the beginning
 - 4.2 Else if choice == 2
 - 4.2.1 User key in the data
 - 4.2.2 Insert at the middle
 - 4.3 Else if choice == 3
 - 4.3.1 User key in the data
 - 4.3.2 Insert node at the end
 - 4.4 Else if choice == 4
 - 4.4.1 Search node by name
 - 4.5 Else if choice == 5
 - 4.5.1 Search node by account number
 - 4.6 Else if choice == 6
 - 4.6.1 Search node by ic
 - 4.7 Else if choice == 7
 - 4.7.1 Delete the first node
 - 4.8 Else if choice == 8
 - 4.8.1 Delete the middle node
 - 4.9 Else if choice == 9
 - 4.9.1 Delete the node at the end
 - 4.10 Else if choice == 10
 - 4.10.1 Sort the data
 - 4.10.1.1 User input sort option
 - 4.10.1.2 if sort option == 1
 - 4.10.1.2.1 Sort by name
 - 4.10.1.3 else if sort option == 2

```

                                4.10.1.3.1    Sort by account number
4.10.1.4    else if sort option == 3
                                4.10.1.4.1    Sort by ic
4.10.1.5    else if sort option == 4
                                4.10.1.5.1    Sort by balance
4.10.1.5    endif
4.11    endif
5.    Endwhile
6.    End

```

2.2 Data Structure Operation

Linked List

The linked list in this program stores the list of bank objects with the attributes account holder's name, account number, IC number and balance. There is no contiguous storage of elements in singly linked lists, and all elements are connected to each other only by a pointer. The index of the linked list starts from 1 instead of 0, like a normal array. Also, there is a head pointer pointing to the first node of the list.

- `insert (Node n) :` insert a node at the front of the list
- `insertMiddle (int pos, Bank a) :` insert node at position pos of the list
- `insertEnd (Bank b) :` insert node at the end of the list
- `removeFront :` remove the first node of the list
- `removeMiddle (int pos) :` remove the node of the position pos of the list
- `removeEnd :` remove the last node of the list
- `findNodeKey(keyType item) :` the item is the key to search like accountNumber, name, and IC, the result is displaying the information of the element found. It will return to 0 once the key is not found in the list.
- `operator << :` cout << list to display menu and all the nodes in list
- `displayAll() :` display details of all nodes in the list