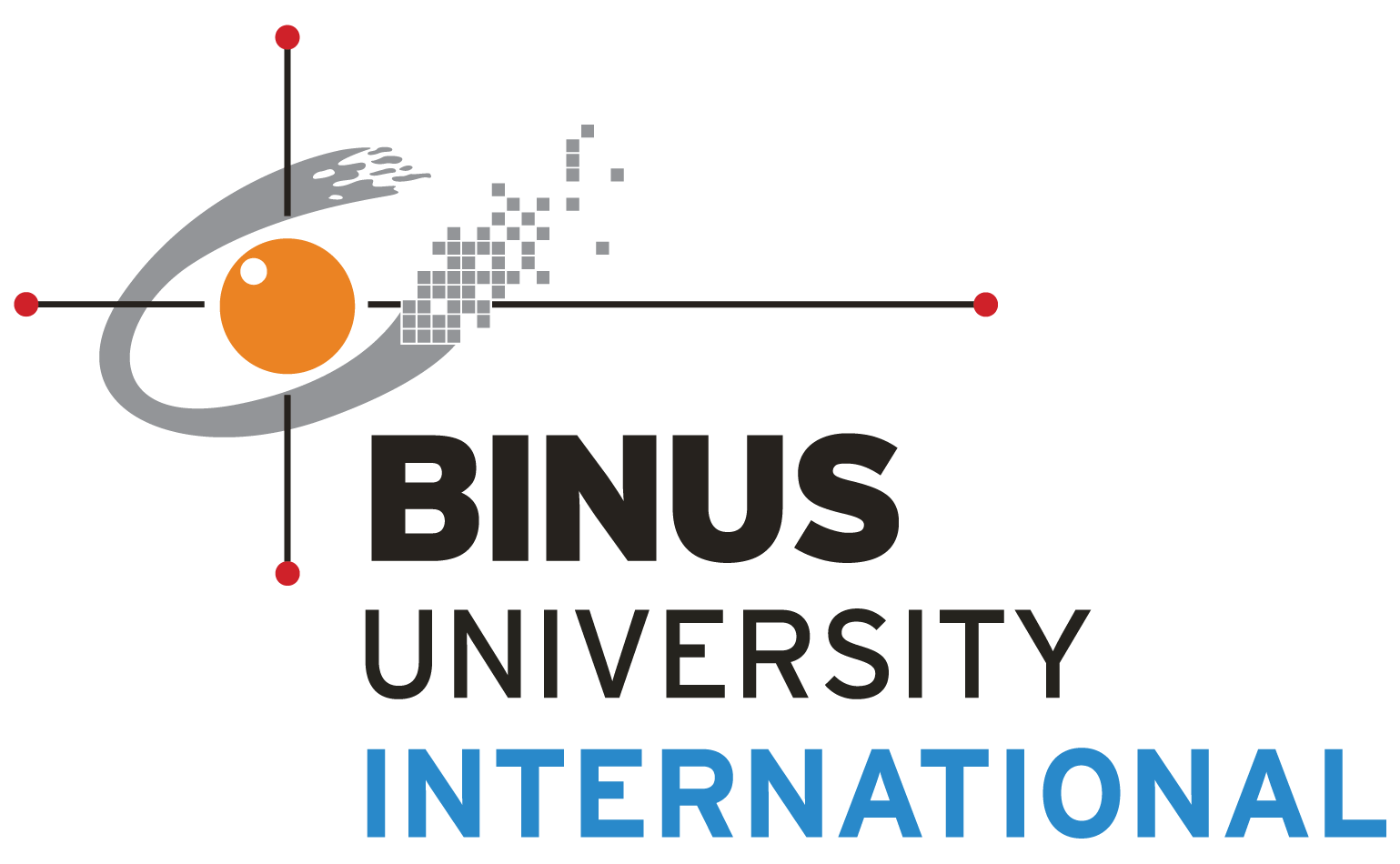
**Data Structures**

**COMP6048001 / Nunung Nurul Qomariyah**



**Final Project**

L2AC

Christopher Alexander Tjiandra / 2502019230

Christopher Owen / 2502019180

Benedictus Filbert Federico / 2502005263

**Table of Contents**

1. Cover1
2. Table of Contents2
3. Mini Library3
   1. Program Description3
   2. Data Structures Implemented3
   3. Application Flow4
   4. Features4
   5. Code Explanation5
   6. Project Link11
4. References11

**Project Report: Mini Library**

1. **Program Description**

Our project is about creating a mini library using linked list as the core of its data structure implementation. Sometimes, a library can be troublesome to manage. Not only it can happen in real life, but also possible to happen in the digital world. A library has books containing many details that identify themselves, such as the title, author, publisher, or even the year the book is released. It is very difficult to have every detail of a book while also managing it at the same time. In addition, there will be someone who comes to the library to read or borrow a book. A librarian needs to store the person’s data, such as an ID number, in order to track all the books, preventing unnecessary incidents. Our application can help store elements of the book and the borrower, so everything can be arranged well.

1. **Data Structures Implemented**

There are a variety of data structures to choose from, but what distinguishes them are the varying purposes each data structure possesses. Looking at the library management system, the most prominent activities are inserting, deleting, and searching. With these activities in mind, it is up to one to decide which data structure does all the above and utilizes the most resources (space and time). After critically thinking through things like this, this project utilized linked list as its data structure.

We can say that singly linked list is the best for this case. To insert an element at head is O(1) after manipulating a couple of pointers. Removing the head element is O(1) and also requires manipulating just a couple of pointers. With inlining in C++ the compiler usually completes that with just a couple of assembly instructions. Any other data structure is more complicated and takes more time even if it is O(1), there are more CPU cycles involved in insertion/deletion.

Singly linked list is also very accessible because it only moves forward and requires less memory compared to other types of linked lists. The implementation is fairly simple but very efficient when running it. Therefore, singly linked list is the most suitable and efficient data structure for this application.

1. **Text

   Description automatically generatedApplication Flow**

Text

Description automatically generated

(1) Add book to the list

Text

Description automatically generated

(4) Search book based ID

Text

Description automatically generated

(5) List all books

(2) Delete book based on title

Text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated

1. **Features**
2. Add a book to the list

Add a book (node) using single linked list containing title, author, publisher, year, and binusian ID.

1. Delete a book based on the title

Search and delete a book (node) based on the title within the linked list. The user required to input a title and the program will delete the corresponding book that matches the title.

1. Delete a book based on binusian ID

Search and delete a book (node) based on the binusian ID within the linked list. The user required to input a binusian ID and the program will delete the corresponding book that matches the binusian ID.

1. Search a book by binusian ID

Search a book (node) based on the binusian ID within the linked list. The user required to input a binusian ID and the program will delete the corresponding book that matches the binusian ID.

1. List all books

Print all books within the saved file (linked list).

1. List all books by an author

Print all books within the saved file (linked list) according to the author.

1. Exit

Exit program.

1. **Code Explanation**

Text

Description automatically generated

These are the libraries that are used.

Text

Description automatically generated

class book is defined, and several variables and pointers are declared within.

Text

Description automatically generated

Inherit from book class

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Declare all class functions

Text

Description automatically generated

void readFile will read all information contained in the books.txt file.

Text

Description automatically generated

void insert function will request the user for a new book (node).

Text

Description automatically generated

void delTitle is a function for deleting a book in the list based on the title of the book. If the title of the book is not in the list, deletion cannot be done.

A screenshot of a computer

Description automatically generated with medium confidence

For searching book (node) according to title

Text

Description automatically generated

void delBinusid is a function for deleting a book in the list based on the binusian ID of the user. If the binusian ID does not exist in the list, deletion cannot be done.

A screenshot of a computer

Description automatically generated with medium confidence

For searching book (node) according to binusian ID

A screenshot of a computer

Description automatically generated with medium confidence

void searchBinusid is a function to search the book based on the user binusian ID, so it will list the book based on the binusian ID that the user input.

A screenshot of a computer

Description automatically generated with medium confidence

void printList is a function to list all the books that have already been inputted.

A screenshot of a computer

Description automatically generated with medium confidence

void printAuthor is a function to print all the books by searching the author's name.

Text

Description automatically generated

void saveFile is a function to save the data from the user and store it in a file named saved.txt

A screenshot of a computer

Description automatically generated with medium confidence

For counting total books (nodes).

Text

Description automatically generated

Text

Description automatically generated

This is the main function to run the code. It will display all the menus and run them using switch case based on the user’s input.

1. **Project Link**

<https://github.com/JugBones/DS-FinalProject>

**References**

<https://stackoverflow.com/>

<https://rrtutors.com/>

<http://www.cppforschool.com/>

<https://youtu.be/Z0egL-qx2-Y>