Reflection from each members

Group K2-7

- Gan Shu Xian (24004577): Student Records Storage
- Lee Yu Xuan (23098783): Search EnginesChatGPT Reflections
- Teoh En Xi (24004582): Sorting
- Yeong Hui Ni (24004529): Recursion
- Tham Wing Shan (24059824): User Interface and Integration
- Kwek Chee Ling (23080328): User Interface and Integration

Module 1- Gan Shu Xian

- What prompts did you use?

Since I need to combine it with other codes, I thought it is highly related to module 5. How will it work?

My method has an error and at the same time give a simple code.

- How did ChatGPT assist you? Include code snippets and/or screenshots.

ChatGPT explained the relation between modules to me. The delete method now is corrected. In addition, the delete code in a more simple version is shorter and clearer.

```
private StudentNode deleteRecursive(StudentNode node, Stri  Copy  Edit
    if (node == null) return null;

    if (id.compareTo(node.matricID) < 0) {
        node.left = deleteRecursive(node.left, id);
    } else if (id.compareTo(node.matricID) > 0) {
        node.right = deleteRecursive(node.right, id);
    } else {
        if (node.left == null) return node.right;
        if (node.right == null) return node.left;

        StudentNode minNode = findMin(node.right);
        node.matricID = minNod matricID;
        node.name = minNode.na
```

- What changes or corrections did you apply?

At first, i wrongly typed the “<” in “>”, I change it to the correct one, thus the code can run.

- Compare the algorithms before and after implementing the corrections. What changed and why?

These changes made the module both functionally correct and integration-friendly, which was essential for working as part of a team.

- What did you learn from the process?

I learn how to debug and improve logic using clear feedback from ChatGPT.
Writing clean code is not just about getting it to work, it's also about making it easy for others to use, test, and extend.

Module 2-Lee Yu Xuan

- What prompts did you use?

-"Can you explain the search using recursion in a simple way?"
-"Can you give a simpler version without recursion?"

- How did ChatGPT assist you? Include code snippets and/or screenshots.

ChatGPT helped me understand tree traversal to search for different student information. It explained that I should start from the root, go left and right using recursion, and check the condition (like name or CGPA) at each node. Later, I asked for a simpler version without recursion.

```
public void searchByName(StudentNode root, String name) {
    int count = 0;
    Stack<StudentNode> stack = new Stack<>();
    StudentNode current = root;

    while (current != null || !stack.isEmpty()) {
        while (current != null) {
            stack.push(current);
            current = current.left;
        }

        current = stack.pop();
        if (current.name.equalsIgnoreCase(name)) {
            printStudent(current);
            count++;
        }

        current = current.right;
    }

    if (count == 0) {
        System.out.println("*Student not found");
    }
}
```

- What changes or corrections did you apply?

The first solution from ChatGPT was too complex. So I asked for a simpler version. After understanding the idea, I rewrote the functions myself using iteration instead of recursion.

- Compare the algorithms before and after implementing the corrections. What changed and why?

The algorithms before implementing the corrections, more complex and harder to understand, however, after implementing the corrections, easy to test and explain step by step.

- What did you learn from the process?

I learned how to build multiple types of search (by name, CGPA, programme, etc.) using in-order traversal. I also learned how to simplify the code using loops instead of recursion to make it easier to understand.

Module 3- Teoh En Xi

- What prompts did you use?

-"How to let users choose which fields to sort by and the order?"
-"Can you show a simple way to sort a list of students by multiple fields?"

- How did ChatGPT assist you? Include code snippets and/or screenshots.

ChatGPT helped me create a menu that allows users to choose which fields to sort (e.g. Matric ID, name, state, programme, CGPA) and whether to sort each one in ascending or descending order.

- What changes or corrections did you apply?

-At first, I tried sorting using Collections.sort with a single comparator, but it only allowed sorting by one field.
-After asking ChatGPT, I learned how to sort by multiple fields, and how to build comparators dynamically usingComparator.thenComparing().

-I also added input validation (e.g. checking if choices were between 1–5) and defaulted to ascending if the user chose an invalid sort order.

- Compare the algorithms before and after implementing the corrections. What changed and why?

The algorithms before implementing the corrections, no option for ascending/descending, however, after implementing the corrections,user can now choose sort direction for each field.

- What did you learn from the process?

-I learned how to collect multiple field inputs and sorting orders using lists.

-This module helped me improve both my Java logic and user experience design skills.

Module 4-Yeong Hui Ni

- What prompts did you use?

“How do I use recursion to count students by CGPA in a binary search tree?”
“Can you write a recursive function to count students in Java?”

- How did ChatGPT assist you? Include code snippets and/or screenshots.

ChatGPT helped me understand how to use recursive tree traversal to calculate statistics like:

-The number of students within a CGPA range
-The number of students from a specific programme
-Support for multiple programme types (AI, SE, MM)
-The height of the tree and balance factor

- What changes or corrections did you apply?

At first, I used a loop approach to count nodes, but it didn’t work properly in a tree structure.After using ChatGPT’s recursive method, I rewrote all the functions using recursion to visit every node.I also added a version to handle multiple programme selections (AI, SE, MM) using boolean flags.I added simple Math.max() to correctly get the tree height.

- Compare the algorithms before and after implementing the corrections. What changed and why?

The algorithms before implementing the corrections, couldn’t calculate balance factor easily, however, after implementing the corrections,can now calculate height and balance factor easily

- What did you learn from the process?

-I learned how to calculate statistics like total students by programme or CGPA.
-I now understand how to get tree height and compute balance factors for nodes.

Module 5-Tham Wing Shan & Kwek Chee Ling

- What prompts did you use?

We asked ChatGPT how to build a menu system in Java, load/save CSV files, handle duplicates, integrate modules, and implement an undo feature using a stack.

- How did ChatGPT assist you? Include code snippets and/or screenshots.
ChatGPT helped us design the menu structure, read and write CSV files, check for duplicate Matric IDs, and use a stack to implement the undo function. It also showed us how to handle input errors and integrate all modules.

- What changes or corrections did you apply?
We fixed input issues by adding `scanner.nextLine()`, prevented crashes by checking if the undo stack was empty, and handled malformed CSV lines using length checks. These changes improved the system's stability.

- Compare the algorithms before and after implementing the corrections. What changed and why?
Before using ChatGPT, the system crashed with invalid input and the undo function was unstable. After corrections, the menu was smooth, the undo feature worked reliably, and file operations were safe and accurate.

- What did you learn from the process?
We learned how to create a user-friendly Java interface, integrate multiple modules, manage CSV files, and implement stack-based undo. ChatGPT helped us understand how to build a complete and robust Java application.