|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Practicum Final Exam – Example Case** | | | | | | |
| **Subject** | | | **COMP6048 – Data Structures**  **COMP6048001 – Data Structures**  **COMP6048016 – Data Structures**  **COMP6048049 – Data Structures** | | |  |
| **Class** | **:** | **-** | | **Start Date** | **: -** |
| **Lecturer** | **:** | **-** | | **Start Time** | **: -** |
| **End Date** | **: -** |
| **End Time** | **: -** |

**Soal**

*Case*

**Sally Salon**

**Sally Salon** is an application used to manage customer data using old technology. Recently, the owner of Sally Salon, Ms. Suki, decides to hire you to help her build a faster, more reliable application using **AVL Tree** as its algorithm. You are to create an application based on the following requirements.

* At the start of the program, there will be **4 menus**, which consists of:

1. **View Available Treatment**
2. **Insert New Treatment**
3. **Delete Treatment**
4. **Exit**

A picture containing graphical user interface

Description automatically generated

***Figure 1. Main Menu***

* If user choose **menu 1** (“**View Available Treatment**”), the program will:
* Check data from AVL Tree. If there are **no data**, **display** following message.



***Figure 2. No Data Message***

* Otherwise:
* Prompt user to input **view order**. Validate the input must be **between** “**pre**”, “**in**”, and “**post**” (**case sensitive**).



***Figure 3. View Input Prompt***

* If **view order** is “**pre**”, **display** all data in **pre-order** format.

Table

Description automatically generated

***Figure 4. View Data in Pre-Order Format***

* If **view order** is “**in**”, **display** all data in **in-order** format.

Table

Description automatically generated

***Figure 5. View Data in In-Order Format***

* If **view order** is “**post**”, **display** all data in **post-order** format.

Table

Description automatically generated

***Figure 6. View Data in Post-Order Format***

* If user choose **menu 2** (“**Insert New Treatment**”), the program will:
* Prompt user to input **treatment name**. Validate the input based on the following conditions:
* Input must **starts with** “**Sally**”.
* Input must be **between 5** and **20 characters** (**inclusive**).
* Input must contains **2 words** at minimum.
* Prompt user to input **treatment price**. Validate the input must be **between 50000** and **1000000**.
* Prompt user to input **treatment category**. Validate the input must be **between** “**Hair Care**”, “**Nail Care**”, and “**Body Care**” (**case sensitive**).
* Prompt user to input **treatment availability**. Validate the input must be **between** “**Available**” and “**Unavailable**” (**case sensitive**).
* Finally, **push** all data unto the **AVL Tree** with **treatment name** as its **key.**

Text

Description automatically generated

***Figure 7. Insert Treatment Data Input***

* If user choose **menu 3** (“**Delete Treatment**”), the program will :
* Check data from **AVL Tree**. If there are **no data**, **display** following message.



***Figure 8. No Data Message***

* Otherwise:
* **Display** all data in **in-order** format.
* Prompt user to input **treatment name**.



***Figure 9. Delete Input***

* Check data from **AVL Tree** based on input. If data **not exists**, **display** following message.

A picture containing graphical user interface

Description automatically generated

***Figure 10. Searched Data Not Exists***

* Otherwise, **remove** corresponding data from **AVL Tree** and **display** following message.

A picture containing graphical user interface

Description automatically generated

***Figure 11. Successfully Delete Data Message***

* If user choose **menu 4** (“**Exit**”), the program will **close**.