Report for TokTik App

Introduction

TikTok is a social media app that allows users to create and share short videos with various effects and filters. In this report we will discuss the development of TokTik in a Linux environment which is going to be a clone of TikTok and will be used in place of TikTok should TikTok get banned in the US. The main focus in this case is based on storage and retrieval of data using Binary Search algorithms for effective retrieval.

Binary search Tree algorithm

A binary search tree is a hierarchical data structure in which every node has a key and a value, and the keys are arranged so that the keys in each node's left and right subtrees are more and less than the node's key, respectively. While the algorithm for finding a node with a given key involves recursively traversing the tree until the key is found or it is determined that the key is not in the tree, the algorithm for adding a new node to a binary search tree involves traversing the tree until the appropriate place for the new node is found. The average time complexity of the binary search tree technique for both insertion and searching is O(log n), however in the worst case, it can be O (n).

Design of the App

TokTik is an app that effectively stores and retrieves user accounts and postings using the concepts of object-oriented programming. The software hierarchically arranges the accounts and posts using a binary search tree data structure. A key-value pair is kept in each node of the tree, with the key being the username for account, and the value being details about the account or post.

The application employs a recursive binary search technique to retrieve data, starting at the root node and comparing the search key to the key of the current node. The algorithm recursively searches the left subtree if the search key is less than the current node's key, and the right subtree if the search key is greater. The associated value is returned if the key matches.

Overall, My TokTik uses data structures including binary search trees and object-oriented design concepts to effectively store and retrieve user accounts and post information, offering a quick and dependable user experience.

In this App we I have multiple classes acting together to achieve al the above mentioned goals.

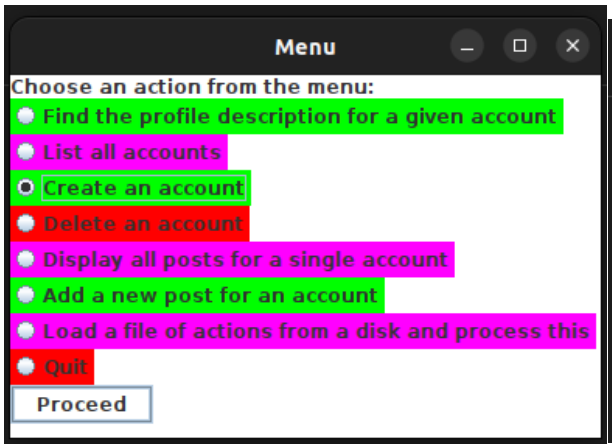
There's BinaryTree class and Node class which work together to provide us all the basic functionalities of a Binary Search Tree including searching , insertion and deleting. There's also a Posts class which is for posts where each post contains a vide, title, and likes information. The account class stores the name of the account and description, it also has a stack to store each specific user's posts, a stack is used since we want to retrieve the posts in order of newest to oldest. There's then a Validating class which is used to validate all the information that needs to be stored (Title, video, likes, Name, description and file to be loaded), this class have methods for each variable to be stored and also uses basic while loops for validating the inputs. We then have a TokTikTree which is used for creation of a binary Tree that we use as our storage data structure for accounts. There's then a Menu class which is for the GUI menu, all inputs from the user are taken via the Gui that is found in this menu and all sorts of communications happen through the GUI. There's then TokTik class which has the main method and it's only task is to create an instance of the menu then everything else happens from there.

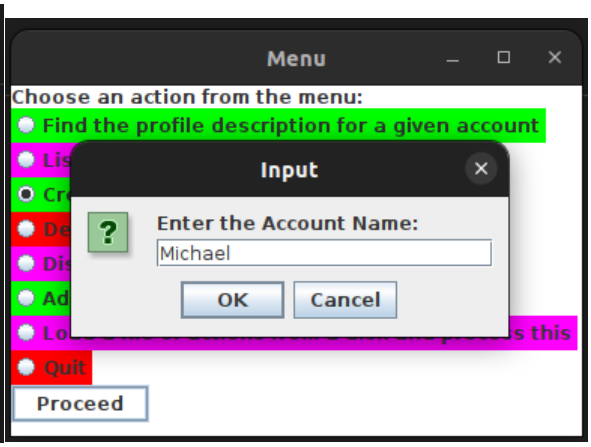
Benefits and challenges of using BST

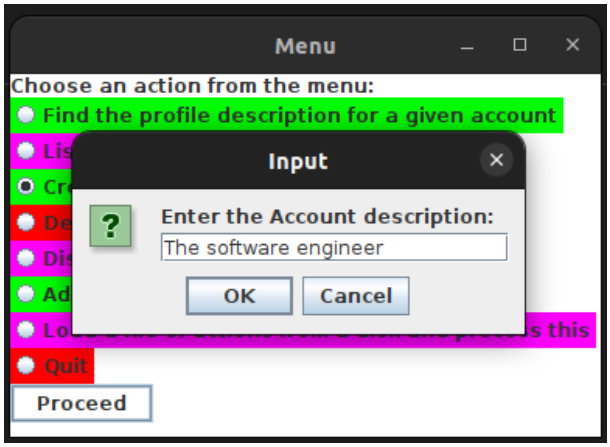
The binary search technique is a quick and effective way to look for a specific element in an array or sorted list. Given that it has an O(log n) time complexity, it is extremely scalable for huge data sets. An app's performance and speed can be greatly enhanced by the usage of binary search, creating a better user experience. Faster search times can be achieved by using binary search to reduce the number of comparisons needed to locate an element. Also, because fewer variables need to be saved, it might lower the memory use of the app. Overall, by offering a quick and scalable search solution, binary search algorithms can improve an app's productivity and user experience.

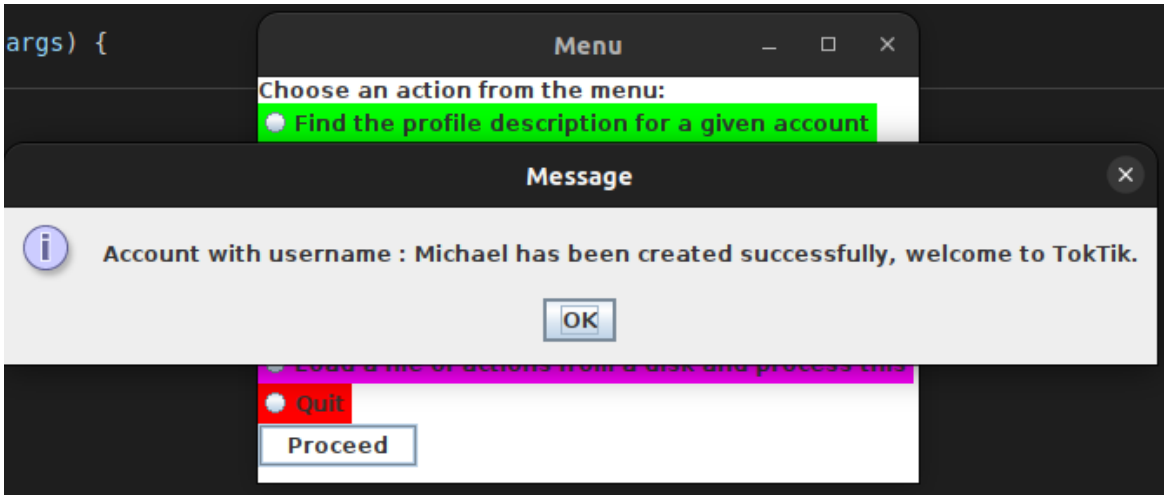
The challenges that comes with it is that it requires sorting after every node creation or update or deletion. This can be time consuming and can have a toll on the ram and processor of the machine.

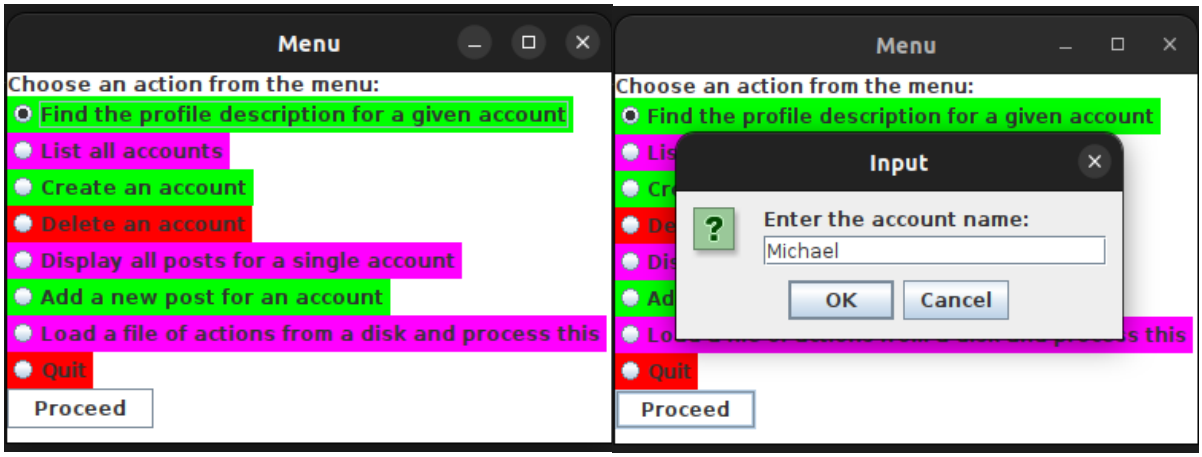
Testing

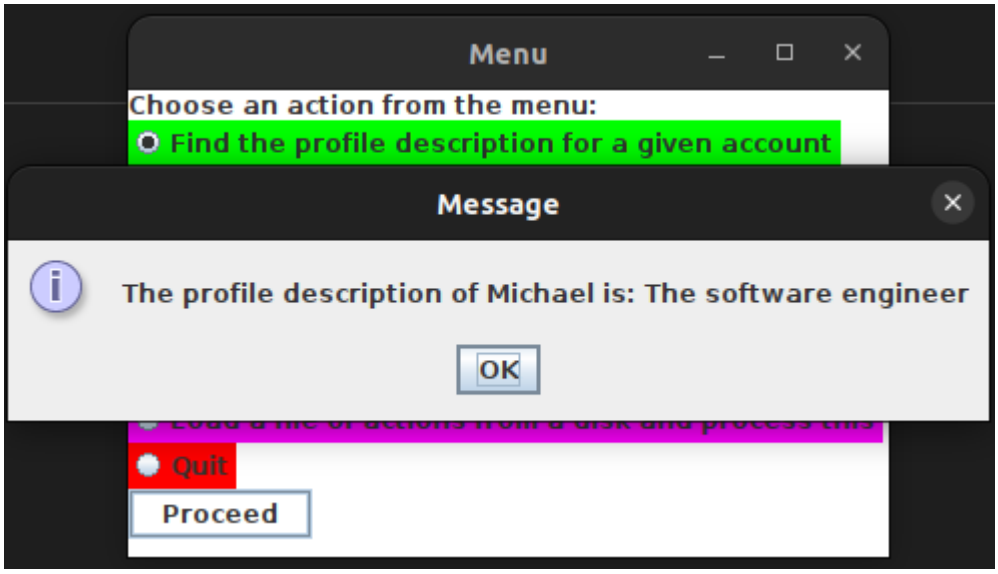

Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed

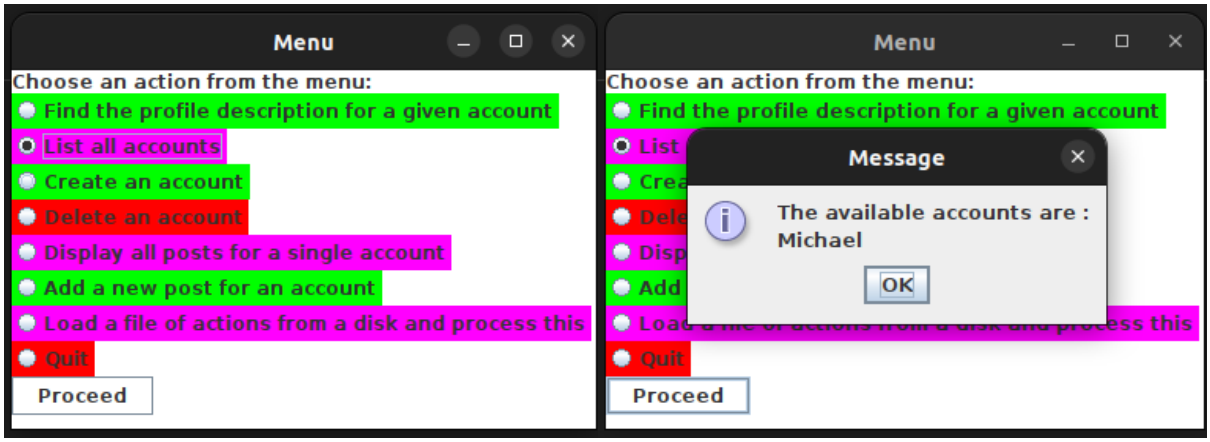

Menu
Choose an action from the menu:
Find the profile description for a given account
Input
Enter the Account Name:
Michael
OK
Cancel
Quit
Proceed

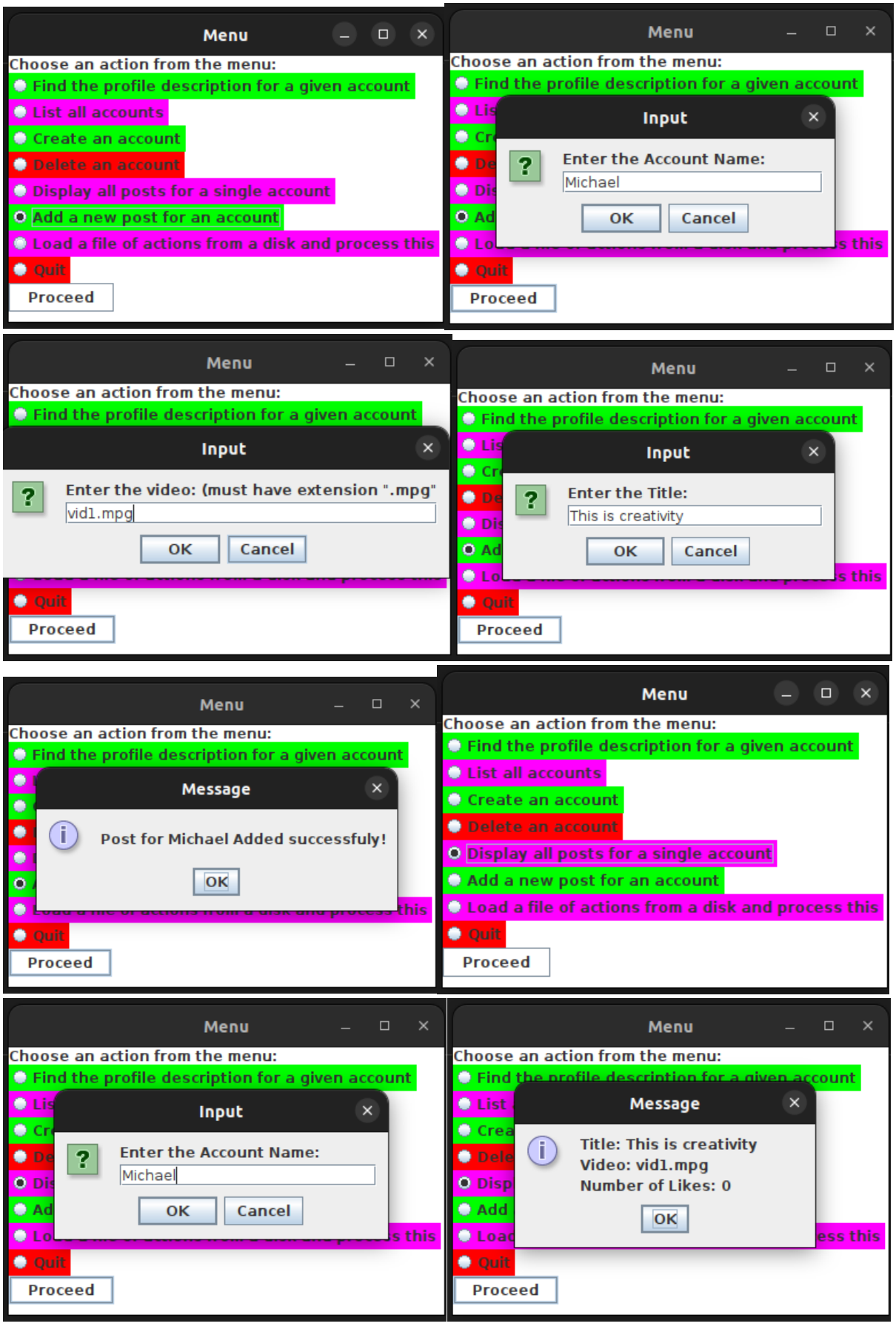

Menu
Choose an action from the menu:
Find the profile description for a given account
Input
Enter the Account description:
The software engineer
OK
Cancel
Quit
Proceed

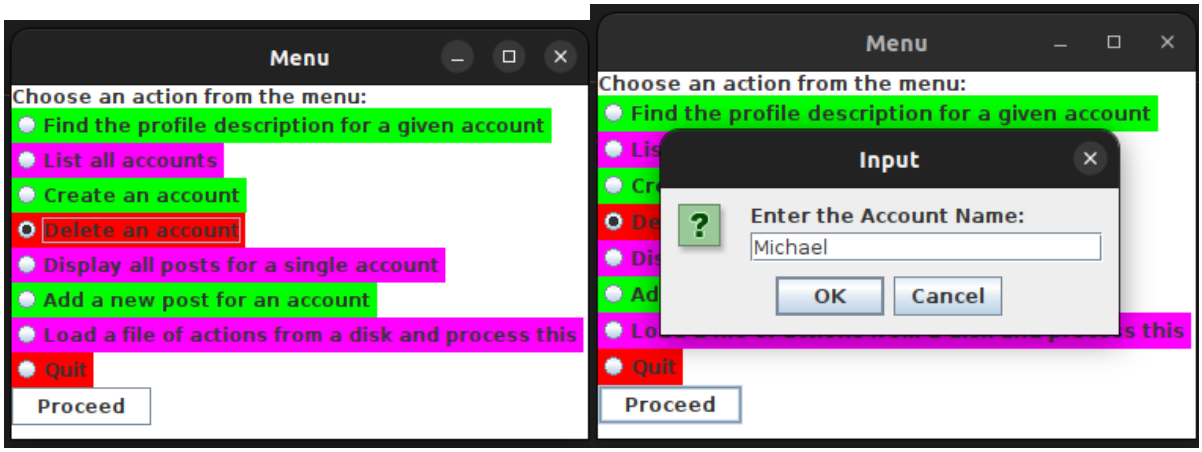

args) {
Menu
Choose an action from the menu:
Find the profile description for a given account
Message
Account with username : Michael has been created successfully, welcome to TokTik.
OK
Quit
Proceed

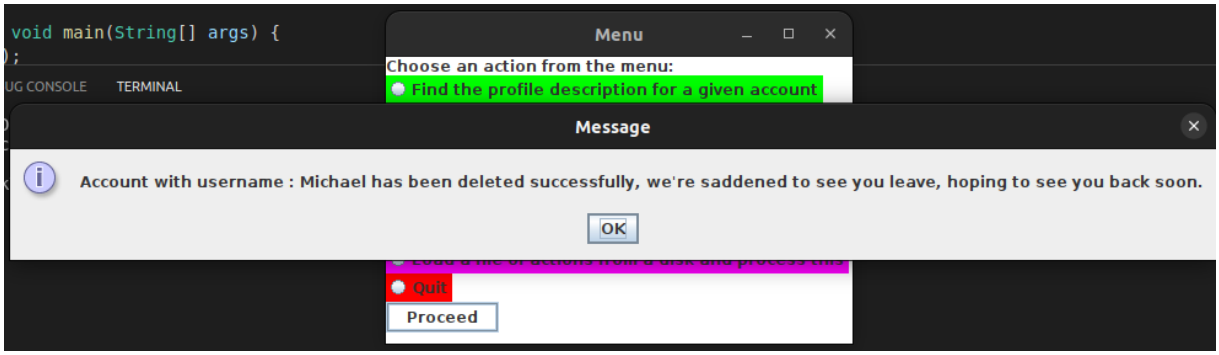

Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed
Input
Enter the account name:
Michael
OK
Cancel

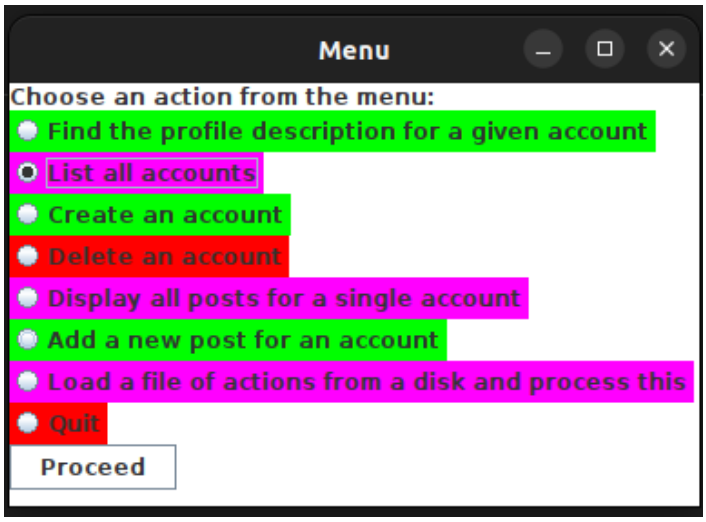

Menu
Choose an action from the menu:
Find the profile description for a given account
Message
The profile description of Michael is: The software engineer
OK
Quit
Proceed

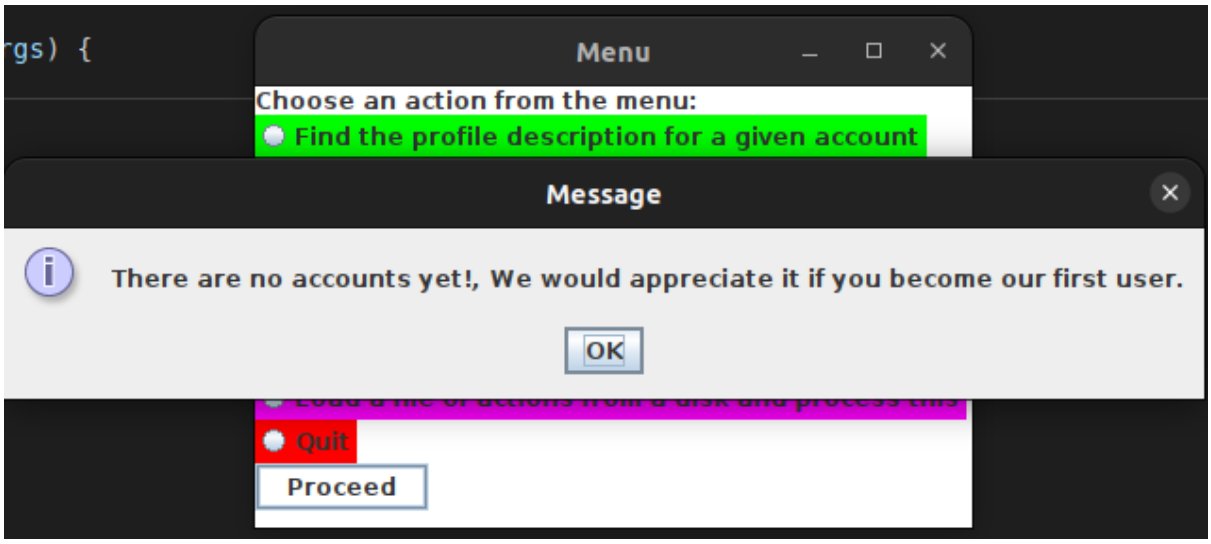

Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed
Message
The available accounts are :
Michael
OK

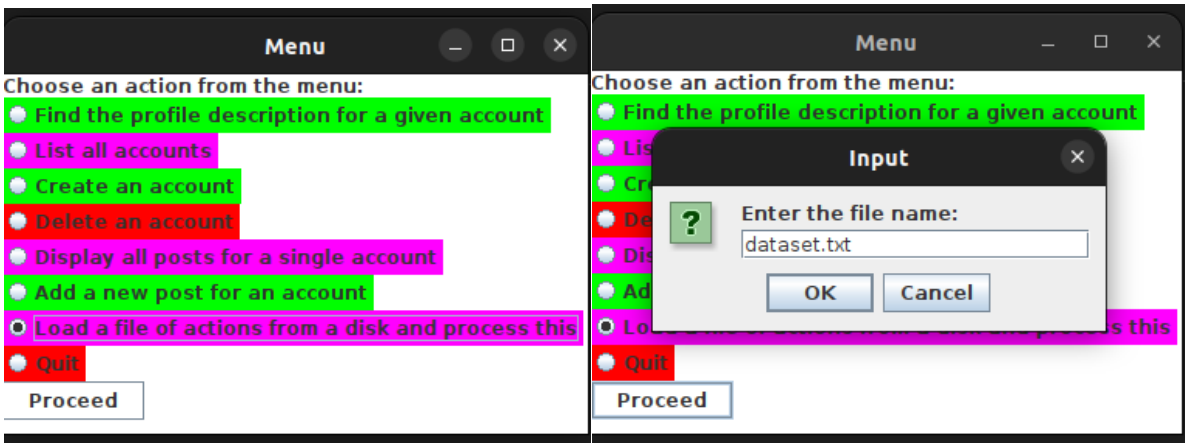

Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed
Input
Enter the Account Name:
Michael
OK
Cancel
Input
Enter the video: (must have extension ".mpg"
vid1.mpg
OK
Cancel
Input
Enter the Title:
This is creativity
OK
Cancel
Message
Post for Michael Added successfuly!
OK
Message
Title: This is creativity
Video: vid1.mpg
Number of Likes: 0
OK


Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed
Input
Enter the Account Name:
Michael
OK
Cancel

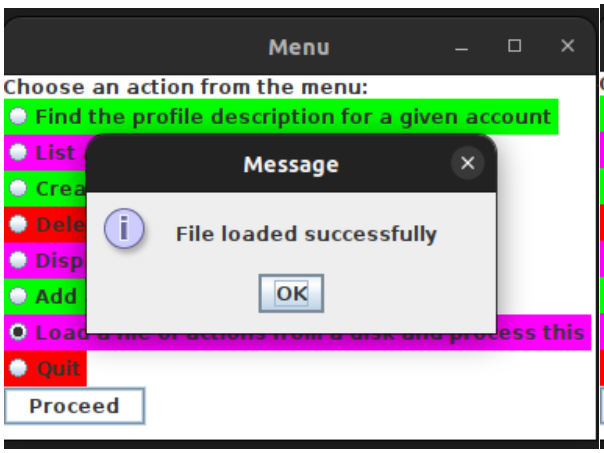

void main(String[] args) {
TERMINAL
Menu
Choose an action from the menu:
Find the profile description for a given account
Message
Account with username : Michael has been deleted successfully, we're saddened to see you leave, hoping to see you back soon.
OK
Quit
Proceed

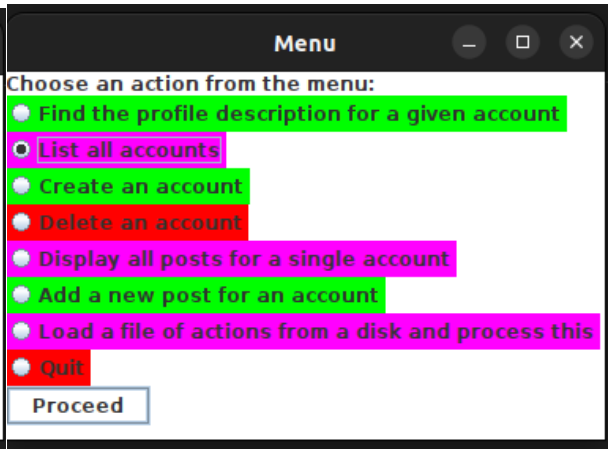

Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed

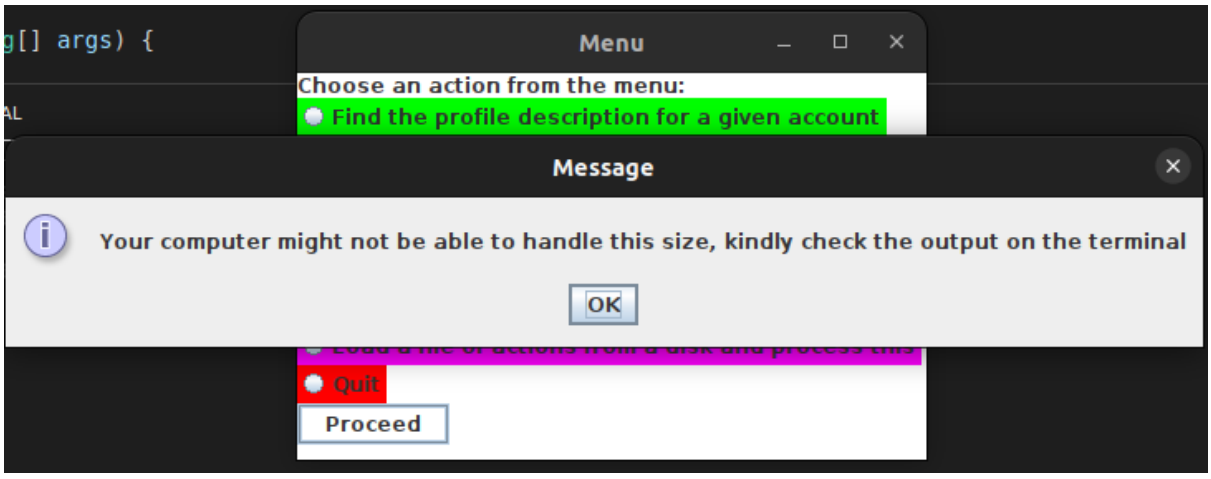

Menu
Choose an action from the menu:
Find the profile description for a given account
Message
There are no accounts yet!, We would appreciate it if you become our first user.
OK
Quit
Proceed

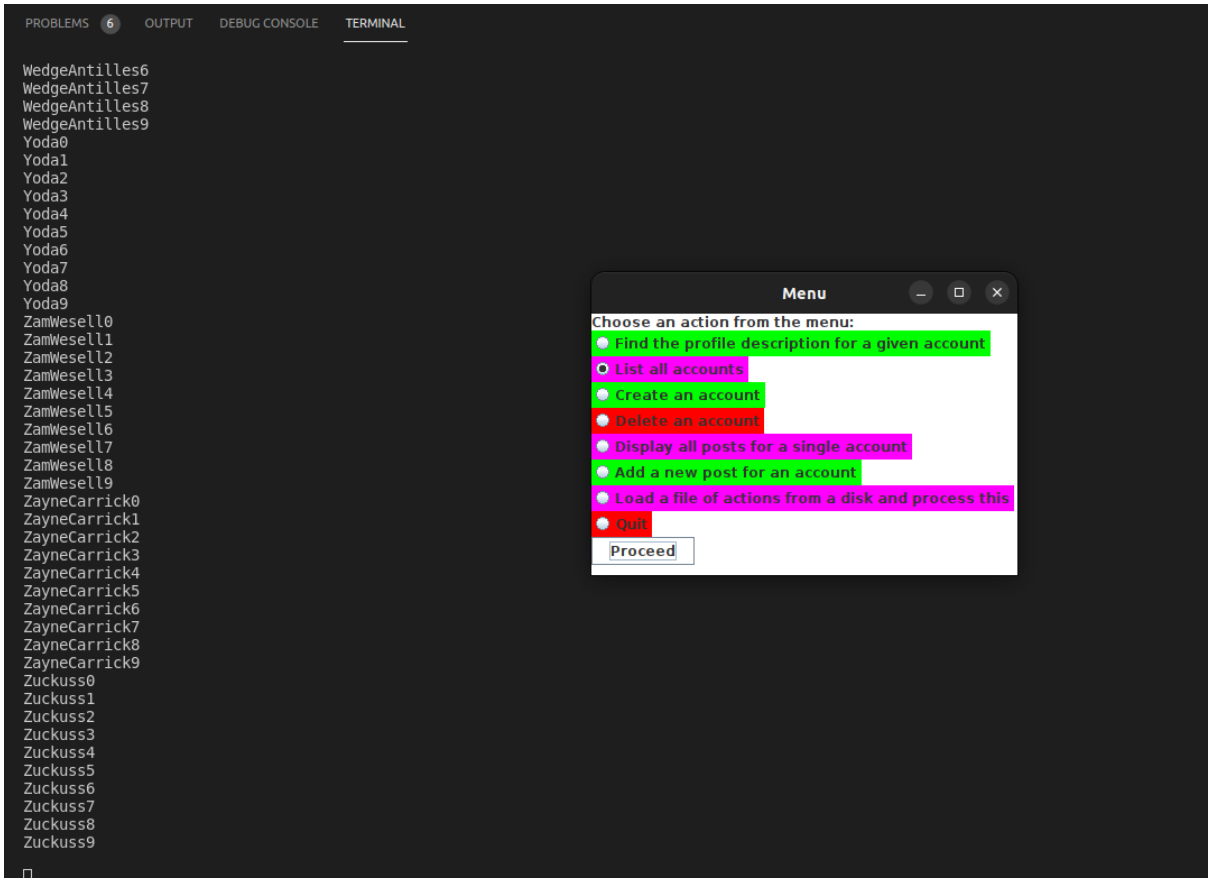

Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed

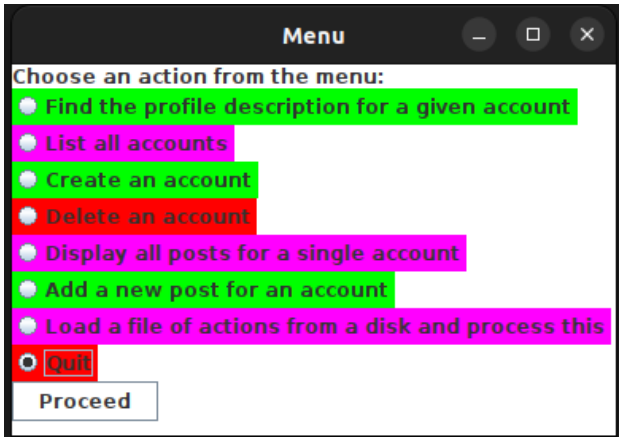
Menu
Choose an action from the menu:
Find the profile description for a given account
Input
Enter the file name:
dataset.txt
OK
Cancel
Quit
Proceed

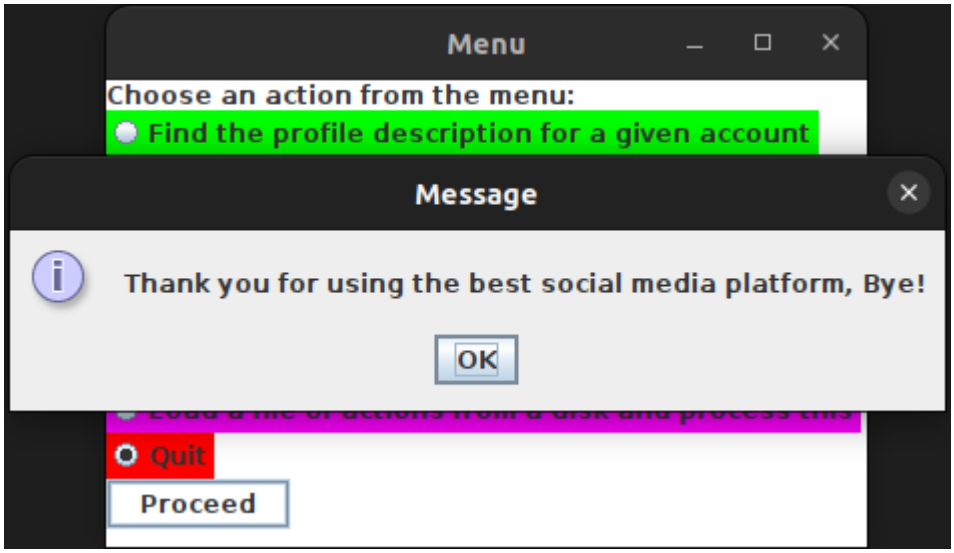

Menu
Choose an action from the menu:
Find the profile description for a given account
Message
File loaded successfully
OK
Quit
Proceed


Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed


Menu
Choose an action from the menu:
Find the profile description for a given account
Message
Your computer might not be able to handle this size, kindly check the output on the terminal
OK
Quit
Proceed


PROBLEMS 6
OUTPUT
DEBUG CONSOLE
TERMINAL
WedgeAntilles6
WedgeAntilles7
WedgeAntilles8
WedgeAntilles9
Yoda0
Yoda1
Yoda2
Yoda3
Yoda4
Yoda5
Yoda6
Yoda7
Yoda8
Yoda9
ZamWesell0
ZamWesell1
ZamWesell2
ZamWesell3
ZamWesell4
ZamWesell5
ZamWesell6
ZamWesell7
ZamWesell8
ZamWesell9
ZayneCarrick0
ZayneCarrick1
ZayneCarrick2
ZayneCarrick3
ZayneCarrick4
ZayneCarrick5
ZayneCarrick6
ZayneCarrick7
ZayneCarrick8
ZayneCarrick9
Zuckuss0
Zuckuss1
Zuckuss2
Zuckuss3
Zuckuss4
Zuckuss5
Zuckuss6
Zuckuss7
Zuckuss8
Zuckuss9
Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed


Menu
Choose an action from the menu:
Find the profile description for a given account
List all accounts
Create an account
Delete an account
Display all posts for a single account
Add a new post for an account
Load a file of actions from a disk and process this
Quit
Proceed


Menu
Choose an action from the menu:
Find the profile description for a given account
Message
Thank you for using the best social media platform, Bye!
OK
Quit
Proceed

Errors like empty inputs, spaces inputs are all handled, the program asks for a valid input should it get such as an input, it displays a message to notify the user and asks for a valid one, until the user enters a valid one or selects cancel. Should the user enter any option other than create account and quit while the tree is empty, It tells the user that there are no accounts. It also doesn't allow users to share the same username.

Creativity:

For creativity I added a GUI , everything operates on GUI, inputs and messages are all displayed on the GUI, I also added a feature for the program to suggest a username should a user enter an existing username. E.g for Michael it can suggest Michael123.

Git Log:

```
mskmic017@sl-dual-300:~/Desktop/Assignment4$ git log | (ln=0; while read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
0: commit c1ac42bd9b76ef6a5df905af14d0f47834499854
1: Author: Michael Maseko <mskmic017@sl-dual-300.cs.uct.ac.za>
2: Date: Wed Apr 19 22:59:08 2023 +0200
3:
4: Final version
5:
6: commit 1522bb62b85cc545d0b85e77b306215b3eeff102
7: Author: Michael Maseko <mskmic017@sl-dual-300.cs.uct.ac.za>
8: Date: Wed Apr 19 10:35:45 2023 +0200
9:
...
121: Author: Michael Maseko <mskmic017@sl-dual-265.cs.uct.ac.za>
122: Date: Wed Apr 12 21:55:40 2023 +0200
123:
124: 2nd commitment with Accounts modified to implement Comparable
125:
126: commit 71997a0e6d729b532c3a99e19a8c627a2d0b8a58
127: Author: Michael Maseko <mskmic017@sl-dual-265.cs.uct.ac.za>
128: Date: Wed Apr 12 21:49:30 2023 +0200
129:
130: 1st commitment with Account class
mskmic017@sl-dual-300:~/Desktop/Assignment4$
```

Summary:

Binary Search trees are one of the best algorithms to use for this app since they allow fast searches and they allow us to retrieve and manipulate data easily in a quicker way. They make things easier as no user wants to wait for a long time for an action to be done after clicking a certain button on the app. Recommendations will be to use a balanced tree like AVL trees to make it much more efficient when it comes to searching though it has more work of balancing the tree every time the user makes a change to the tree