Assignment 4
Binary Search Trees

BY WTBROC002

Roche Witbooi | CSC2001F | 17 April 16, 2023

Contents

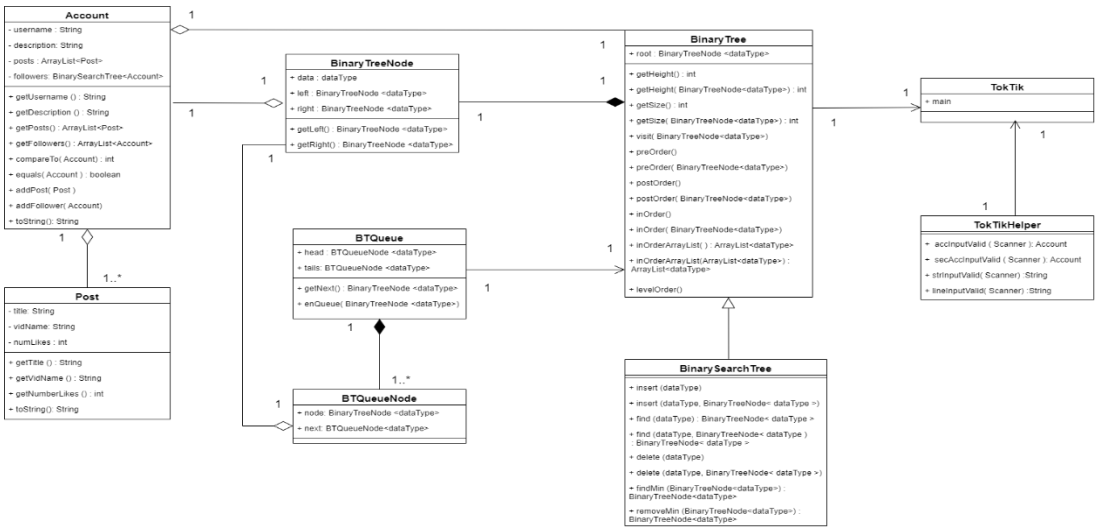
Object Oriented Design:

TokTik contains the main program that the user interacts with. The TokTikHelper class contains code helper methods to avoid retyping code in the TokTik program.

A Binary Tree is used as the main data structure to store accounts used in the TokTik program. It is an efficient structure for storing many accounts as it allows the user to access and delete accounts faster than a linked list or array. Furthermore, there is no bound to the number of accounts that can be stored. The Account class represents the TokTik users. Each account is unique; however, all accounts have a unique username description, posts, and followers. By creating an Account class account objects can be created with unique values for each user. Since a user can have multiple posts the Post class is created to store the unique attributes associated with each post. A post has a title, video name, and several likes and is stored in a user account using an Array List. Similarly, a user also may have followers, these are accounts stored as a binary search tree which is an attribute of the Account class.

The Binary Search Tree is implemented using the main BinaryTree class and its functionality is extended using a child class Binary SearchTree which implements the insert, find, and delete operations. The Binary Tree is made up of nodes that store the data object in each node, together with the left and right child nodes to support the Binary Tree Structure. The BTQueue class is used to implement the Queue data structure for level Order traversal. The Queue is made up of BTQueueNodes which store the Binary Tree Nodes during level order Traversal.The Binary Tree, BinarySearchTree, BinaryTreeNode, Queue, and BTQueueNode are implemented as generics to allow for the code to be repurposed in various scenarios by inserting different classes into the datatype variable.

The TokTik program uses the generic classes that help implement the BinarySearchTree to store Account objects. The Account class is inserted as “dataType” parameter in the generic classes this allow the Binary Tree to store accounts.



Testing values:

CHOICE SELECTION

```
Choose an action from the menu:
1. Find the profile description for a given account
2. List all accounts
3. Create an account
4. Delete an account
5. Display all posts for a single account
6. Add a new post for an account
7. Load a file of action from disk and process this
8. Quit

9. Follow an account
10. Recommend accounts to follow
11. Account summary

Enter your choice (1-11): 1
Enter the account name:
```

The user is required to enter a numeric value between 1 and 11 which corresponds to an action.

When the value entered falls outside boundaries 1 and 11, the user is prompted to select an option in the valid range.

The values **12** and **0** have been provided.

The program provides the message: **"Select an option (1-11)"**

```
Enter your choice (1-11): 0

Select an option (1-11)
```

```
Enter your choice (1-11): find the description

Please enter a number from 1-11
```

In the case where the user enters text instead of an integer, the program prompts the user to enter a number.

The text "**find the description**" is entered by the user. The program responds with the message "**Please enter a number from 1-11**".

CHOICE 1:

```
Enter your choice (1-11): 1
Enter the account name: User1

The profile description is : is the first TokTik user!
```

The program is required to find the description of an account that corresponds to the account username entered.

The account with the username "**User1**" is provided by the user. The program finds the account details and displays the account description **" is the first TokTik user!**"

```
Enter your choice (1-11): 1
Enter the account name:

No account - found.
Enter your choice (1-11): 1
Enter the account name: User1

Enter your choice (1-11): 1
Enter the account name: User 1

No account User found.
```

Where the user enters a space instead of the account name the program responds with the message "**No account - found**."

If an account that does not exist is entered, the program informs the user the account has not been found.

Where the user has entered a space in the account name, the program takes the first word as the account name. In this case, “User 1” is provided. Since there are no accounts with this username, the program displays the “No account <entry> found” message

CHOICE 2:

The program is required to list all the users of TokTik. The program does this by first stating the account username followed by the username.

```
Enter your choice (1-11): 2

Account username followed by description
User1 is the first TokTik user
User2 is the second TokTik user

Enter your choice (1-11): 2

Account username followed by description
4-LOM0 miniscule's Hamlin's Lorenz
4-LOM1 sale contingent Gielgud's
4-LOM2 prod's tawdry imperfect
4-LOM3 defensiveness's Banks's subletting

Enter your choice (1-11): 2

There are no accounts listed
```

In this caseTokTik only has two users “**User1**” and “**User2**”.

Once TokTik has more accounts this selection will list all the accounts. To the left are the first four accounts listed after processing the **dataset.txt** file, the remaining accounts are also listed below these four lines.

If TokTik has no account users the program displays the message” **There are no accounts listed**”

CHOICE 3:

Selection 3 allows the user to create new accounts by entering a Username and description. The username should only be one word. But the description can be longer.

```
Enter your choice (1-11): 3
Enter username of the new user: NewUser
Enter the profile description: new TokTik user
NewUser account successfully created
```

The user has entered the username “**NewUser**” and the description “**new TokTik user**”. The program displays the message the <new account username> in this case “NewUser” account has been **successfully created**. This informs the user that the information they have entered is valid and a new account with their provided information is now stored in the TokTik system.

```
Enter your choice (1-11): 3
Enter username of the new user: New  User
Enter the profile description: is a new user
New account successfully created
```

Where the user has entered two words for the account the program will attempt to use the first word as the account username. Since the user has entered “New User” as the account username the program has created a new account using the first word “New” as the username.

```
Enter your choice (1-11): 3
Enter username of the new user: NewUser

An account with this username already exists. Please try again.

Choose an action from the menu:
1. Find the profile description for a given account
2. List all accounts
3. Create an account
4. Delete an account
5. Display all posts for a single account
```

The user has provided "**NewUser**" as the account username, but since this username already exists, the account cannot be created since all account usernames should be unique. The program, therefore, tells the user "An account with this username already exists. Please try again". The user will then be taken back to the main selection options.

CHOICE 4:

Selection 4 allows the user to enter the username of an account they wish to delete. The program then requests that the user confirm the account deletion and based on the confirmation deletes the

```
Enter your choice (1-11): 4
Enter name of the user you wish to delete: NewUser

Are you sure you would like to delete this account (y/n)? :y

The account of user: NewUser has been deleted.
```

account if it is present in the TokTik system or not.

The user has entered the username "**NewUser**" and the user wishes to delete this account. The user confirms the deletion by responding with "**y**" to the prompt "**Are you sure you would like to delete this account (y/n)?**". The program then displays a deletion message "**The account of the user: <entered username> has been deleted** ".

```
Enter your choice (1-11): 4
Enter name of the user you wish to delete: NewUser

Are you sure you would like to delete this account (y/n)? :n

The account deletion was unsuccesful
```

The user has entered the username "**NewUser**" and the user wishes to delete this account. The user changes their decisions and does not confirm the deletion by responding with "**n**" to the prompt "**Are you sure you would like to delete this account (y/n)?**". The program then displays a message indicating that the account has not been deleted by displaying "**The account deletion was unsuccessful**"

```
Enter your choice (1-11): 4
Enter name of the user you wish to delete: User1

The account you wish to delete does not exist
```

However, in the case where the account username entered does not belong to an existing account

the program displays "**The account you wish to delete does not exist**". Since the account entered "**User1**" is not an existing account the message was produced.

CHOICE 5:

```
Enter your choice (1-11): 5
Enter account name: NewUser

Title: My third Post!
Video: third.mp4
Number of likes: 100

Title: My second Post!
Video: second.mp4
Number of likes: 50

Title: My first Post!
Video: new.mp4
Number of likes: 10
Enter your choice (1-11): 5
Enter account name: NewUser

NewUser does not have any posts

Enter your choice (1-11): 5
Enter account name: User

The account does not exist
```

By selecting this choice the user enters an account and all the accounts posts are listed from latest to oldest.

The user has entered the account username "**NewUser**". The program, therefore, lists all the user's posts beginning with the most recent posts.

In the case where the user entered does not have any posts the program displays the message **"<account name> does not have any posts**." In this case "**NewUse**r" has not made any posts.

Where the account entered "**User**" is not an existing account the program displays the message "The account does not **exist**"

CHOICE 6:

```
Enter your choice (1-11): 6
Enter account name: NewUser
Enter post name: My first Post
Enter video name: first.mp4
Enter number of likes: 12
Enter your choice (1-11): 6
Enter account name: NewUser
Enter post name: my second post
Enter video name: second mp4
Enter number of likes: 123

Title: my second post
Video: second
Number of likes: 123

Title: My first Post
Video: first.mp4
Number of likes: 12
```

Selection six adds a post to an account. The user is prompted to enter an account username, the post name, the video file name, and the number of likes. The post is created and added to the corresponding user account entered.

The user has entered the username "**NewUser**" and wishes to create a post "**My first Post**" with the video file name "**first.mp4**" and the post has "**12**" likes. The post is then added to NewUser's account.

Where the user has entered the video name "**second mp4**" the program uses the first word as the video name.

The account "NewUser" now has two posts the first with all the information as entered and the second with the adjusted video name, corresponding to the first word entered.

```
Enter your choice (1-11): 6
Enter account name: NewUser
Enter post name: My last post
Enter video name: last.mp4
Enter number of likes: last

The number of likes must be a an integer greater than 0

Choose an action from the menu:
1. Find the profile description for a given account
2. List all accounts
```

Where the user has not entered a value for the number of likes. For example, "**last**". The program displays a message, "**The number of likes must be an integer greater than 0**". The post is not added and the user is directed back to the main selection options.

```
Enter your choice (1-11): 6
Enter account name: Account1
The account does not exist
```

Where the user enters an account user that does not exist like "Account1" the program displays the message "The account does not exist".

CHOICE 7:

The selection prompts the user for a file name and loads actions from the text entered in the file.

```
Enter your choice (1-11): 7
Enter the file name: dataset.txt
```

The user enters the file name "**dataset.txt**". The file is processed and based on the command in the file the program creates users and adds posts.

```
Enter your choice (1-11): 2

Account username followed by description
4-LOM0 miniscule's Hamlin's Lorenz
4-LOM1 sale contingent Gielgud's
4-LOM2 prod's tawdry imperfect
4-LOM3 defensiveness's Banks's subletting
Zuckuss4 Osceola Calexico Hawthorne
Zuckuss5 too crofts rustiest
Zuckuss6 elicit overbite clarioning
Zuckuss7 Mahavira theocratic gadget
Zuckuss8 Kim's Sheridan's puzzle
Zuckuss9 goldsmith's grape's locomotive's
```

The result of processing the file creates the accounts listed, some of the listed accounts have been omitted from the screenshots as the file is quite large. However, the first four and last six have been provided.

```
Enter your choice (1-11): 7
Enter the file name: tester.txt

Delete Greedo8 video6283.mpg 8251 Kasai Aquitaine's Helvetius diagrammatic empa
thetic : this action is invalid
Commands must begin with Add/Create
```

Where the file contains a command that does not begin with Add or Create like the provided "**Delete Greedo8 …"** line the program displays the line that does not conform to the standard and explains **"<the command line>: this action is invalid"** and that "**Commands must begin with Add/Create**."

```
Enter your choice (1-11): 7
Enter the file name: nofile

The file was not found
```

Where the file name entered is not present in the program working directory the program produces the message "**The file was not found**"

CHOICE 8:

```
Enter your choice (1-11): 8

Bye!
```

The user selects 8 to terminate the program and the message "**Bye !**" is produced.

Creativity:

The TokTik account management program additionally allows users to follow other accounts, recommends accounts for the user to follow, and displays a summary of the user's information.

CHOICE 9:

```
Enter your choice (1-11): 9
Enter your account name: NewUser
Enter the account you wish to follow: User1
You now follow User1
```

This selection enables the user to follow another user by entering their account name and the account they wish to follow. A Binary Tree is created for each account to store the account's followers. This option checks that both the user's account and the account they wish to follow exist before adding the user as a follower to the account entered."

CHOICE 10:

```
Enter your choice (1-11): 10
Enter your account name: NewUser
Enter a topic of interest (e.g cat): dog

You may want to follow these accounts:
User2 has a dog
User3 watches dog videos

Would you like to follow any of these accounts (y/n)? y

Enter the account name: User2

You now follow User2
```

The user is recommended accounts based on an interest that they enter. The program then searches all the descriptions of all the accounts in the system for the word of interest. The program then lists all the accounts that are description contain the word of interest and asks the user if they would like to follow any of these accounts. The user then enters the username of the account they wish to follow. The user is then added as a follower of the account which requested to follow.

CHOICE 11:

```
Enter your choice (1-11): 11

Enter your account name:
User1
USER1
Welcome to your account Summary:
Description: is new
Total Posts: 2
Your post popular post is:

Title: my second video
Video: video4079.mpg
Number of likes: 8116

Number of followers is : 1
Number of accounts following: 0
```

This selection allows the user to enter their username and obtain a summary of their account details.

The summary includes the account username and description. Furthermore, the user's total number of posts and the details of the most popular post are displayed. The most popular post is the post with the highest number of likes. Lastly, the summary states the number of accounts following the user account and the number of accounts the user follows.

Git log

```
roche@roche-VirtualBox:~/Desktop/CSC2001F/Assignment_4$ git log | (ln=0; while
read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
0: commit fec512f0e7847a102de0a918eb146b1beb70cff5
1: Author: Roche Witbooi <roche.witboooi@outlook.com>
2: Date: Sun Apr 16 11:33:49 2023 +0200
3:
4: version 8
5:
6: commit 6a6de4b69ea00b2ae0872b2ce45d8b81e229435b
7: Author: Roche Witbooi <roche.witboooi@outlook.com>
8: Date: Sun Apr 16 10:38:42 2023 +0200
9:
...
37: Author: Roche Witbooi <roche.witboooi@outlook.com>
38: Date: Sun Apr 9 15:29:36 2023 +0200
39:
40: version 2
41:
42: commit c437be07c59d9c20594bd72da4124afe6b9410b6
43: Author: roche <roche.witboooi@outlook.com>
44: Date: Sat Apr 8 13:23:30 2023 +0200
45:
46: version 1
roche@roche-VirtualBox:~/Desktop/CSC2001F/Assignment_4$
```