GIT Department of Computer Engineering
CSE 222/505 - Spring 2023
Homework # Report

Hamza konaç
210104004202

1. SYSTEM REQUIREMENTS

Main Requirement

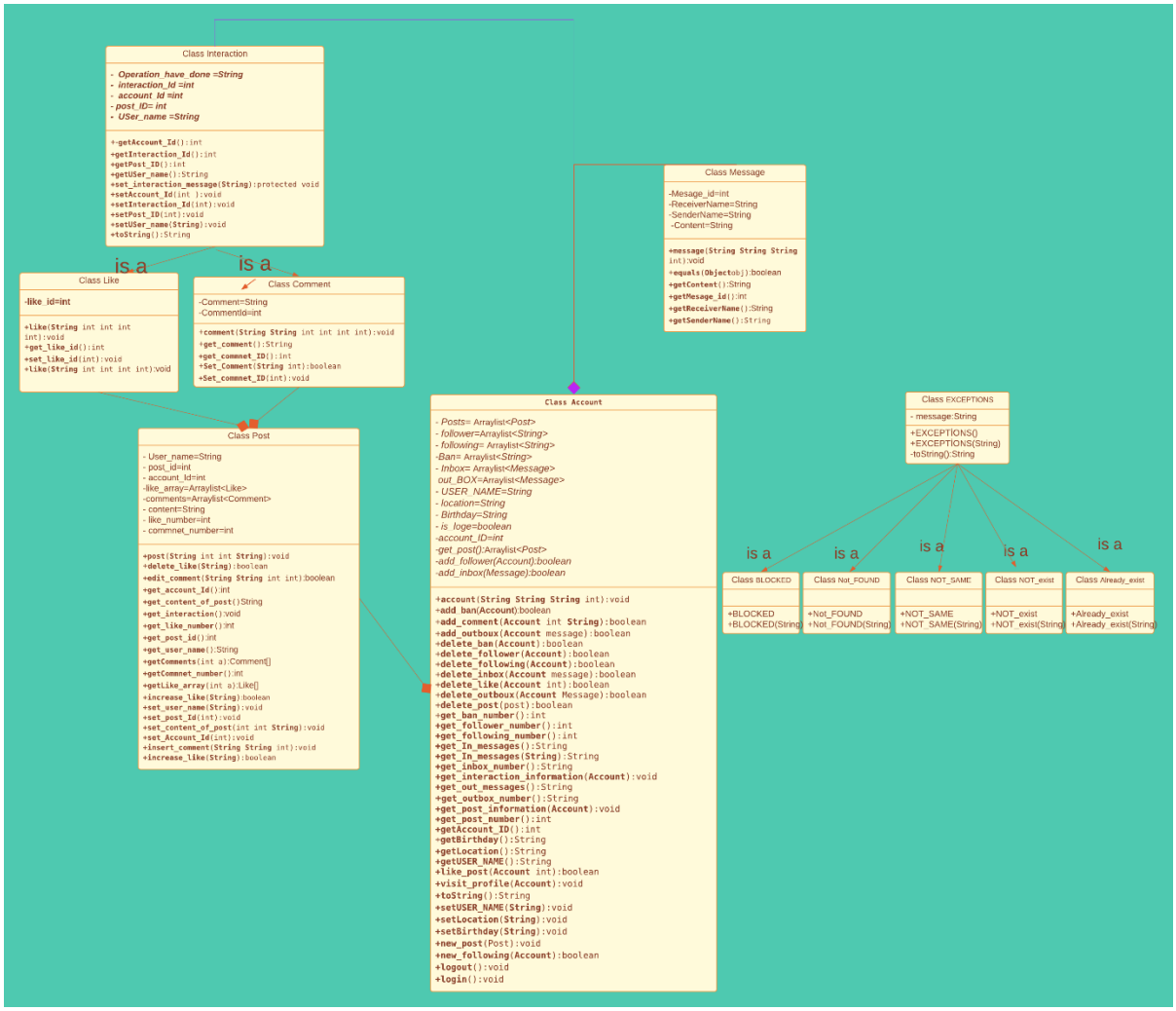
The system is basic social media application assignment. In this assignment user must create an account and he/her can follow another accounts , share post, view another post , like or commnet a post that post's user require to be found user following list.This likes and commnents are also interaction .Furthermore user can send message another account that must be found follower list.Finaly user can block an account that has already exist. System requirements for this assignment are class of like , comment , message, post ,interaction and account.Account class will be main class for launch all account together. This class can share post, send and receive message , like and comment a post and block an user.This class contain array of post, message and interaction classes with array of follower , following , block that has primitive type. Post class have array of like and commnet class.This arrays contains like and comment for each posts. For like ,comment and interaction classes system need a basic inheritance. Like and commnet classes derived from interaction class.each like and commnet will be an interacton at the same time . Therefore in basic class that is interaction class include an interaction message

Additional Requirements:

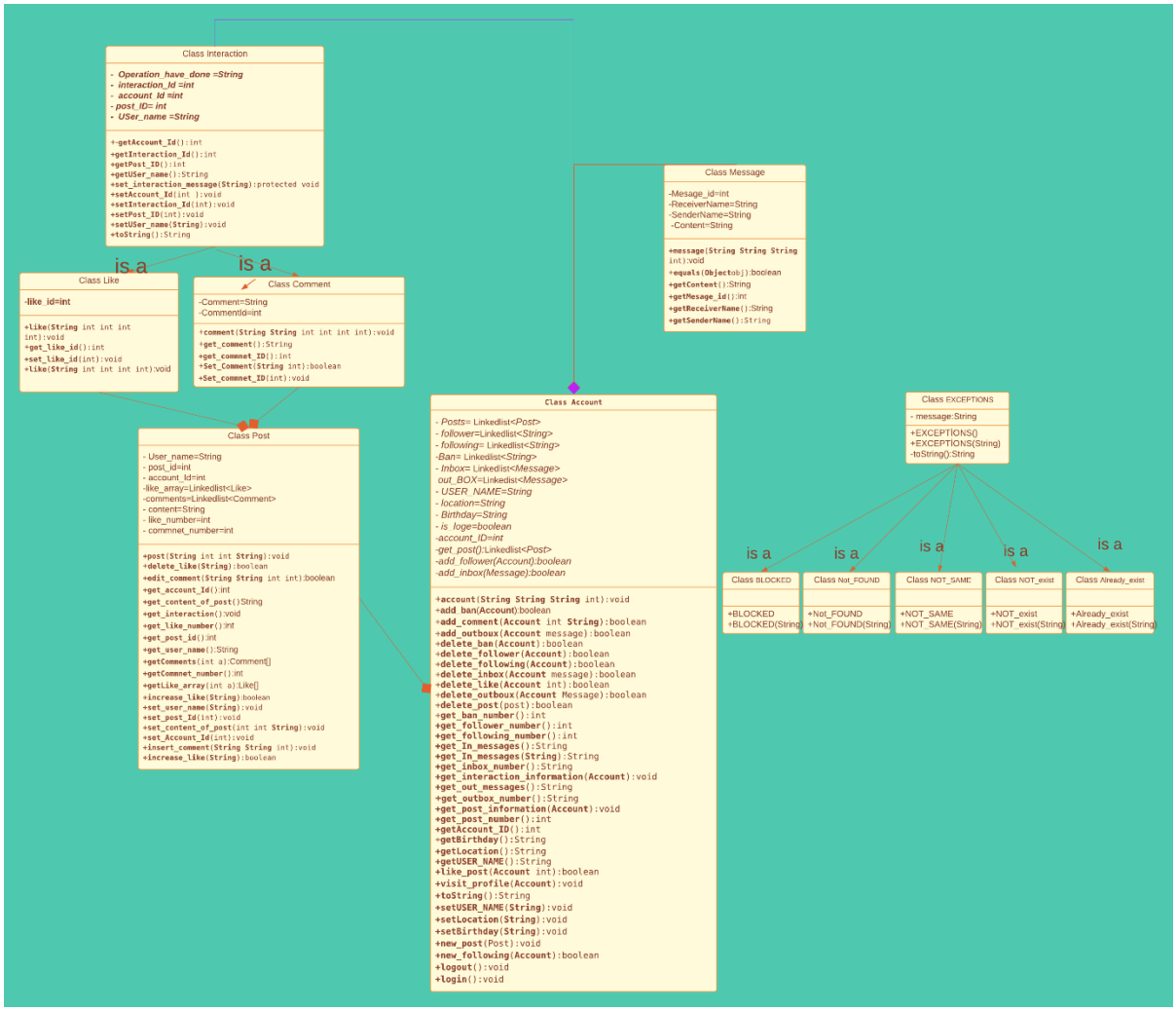
In this assignment want us to 3 more implemantation.For this implemantation we need use 3 data structure named linkedlist arraylist and Ldlinkedlist.Ld linked will implemented by me and it perform remove operation by laz deletion rules.It means It just marked node intstead of phsically remove.if marked node number will be 5 percent of the total number ,marked node will be delete.but in my implementation if marked node number will be 2 I have removed marked nodes.

2. USE CASE AND CLASS DIAGRAMS

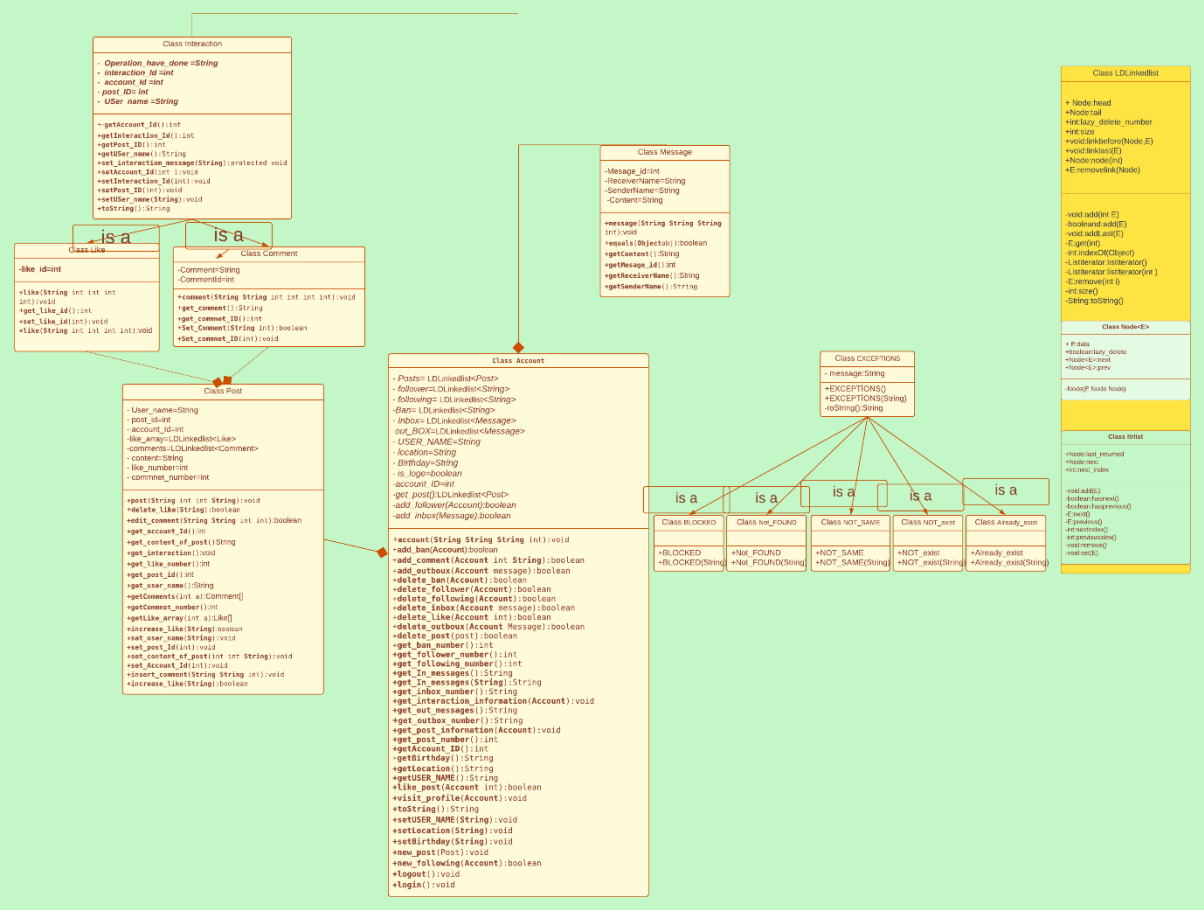
Arraylist Part:



Linkedlist Part



LDLinkedList Part



3. **PROBLEM SOLUTION APPROACH**

General Solution(HW1 Based):

For account I have created an account class.User can follow another users, send messages , share posts,like or comment post and block another accounts.I have looked step by step . for follow I have decleared arrays for all of this informations.if user follow someone it insert the following array's. At this point there is a point we need to care, If a account follow someone, user that has followed follower number must be increase.we need to do this for inbox and outbox.For perform this operation at the same time I take second user's object as an argument and pass it with method that perform operations.In post class there are array of like and commnet .this class contain Id and username. If user want to like or comment a post I take post ID and user's account as an argument for method. Firstly I find post for user that has this post. After I add like or comment . for this operation I have to create an like or comment object. This operation come true in post class . ofter decleration I add it to in like or comment array. At the this point there is one point that we need to care of, each like and coomment will be an interaction because of inheritance. We need to save this interaciton.

interaciton class include a message that contain usrname with done operation , id of occount and ıd of post. This interaction add in interaction array . for post class there is no more operation. Continue with acount class . user can send or receive message. So that we need to message class. Furthermore we need to 2 messages array .one of them will be inbox and another one will be outbox. If user want to send message an user there will be multiple conditions. User must not be blocked and user that want to send message must be in following list.after this condition message objects will be create. After creation , it insert to outbox. After insertion program insert to message receiver inbux.for view profile each acount has override method that return the infırmation of the account . if user A want to view user B profile , programe will call toString method for object of B class.To collect interaction I have 2 way.First one is use like and comment arrays.In Pdf sceanario also use this way. First take likes interaction.for do this just use superclass informations.But Have not performed this way .instead of this way I have decleared a String variable.Initiliazation of the this string will be by interaction have done .So if Account A will like account B's post , initialization will be like this " A have liked B's post.post Id is 0.account Id is 1".final operation is block an account .There are 2 array for this operation.one of them name is "Ban" another one name is "Banned". İf account A block account B, account B's will inserted account A's Ban list an Account A's name will be inserted in Account B's Bunned list. With this way program can grasp blocked user and give error message when a blocked user want to acces user that have block this user.

HW3 Based solution:

In hw3 dont contain new algoritm .It aim to compare and show Time complexity of data structure.to perform this I have use a few idea, compare them together and I want to see differences that ı have curied .

Actually In arraylist there is no question I want to learn.I have add program to arraylist method and code has finished.

But for Linkedlist class and LDLinked classes I Have tested iterator , lazy deletion and linkedlist class function.To do this comparements I have used only iterator in Linkedlist version.I have implemented all method again .For add follower , following,new message , new post ,new comment ,block account and remove all of this operation I have used iterator.

In LDLinkedlist version I have used linked list function that I implemented.for example add,remove,get or contain.in addition to I have performed lazydeletion in remove method.In lazy deletion I have marked node that want to remove.if another node select,I remove all node that marked.

Arraylist->Complexity of this no different from first version of the assignment.But it work faster than my code because of algorithms.

Linkedlist->In linked list if not used iterator and double linkedlist , complexity of the add remove and get function not be better than array list.If iterator used add remove and get complexity will be O(1).

LDLinkedlist-> LDLinked List is designed as double LinkedList. *Node<E>* inner class is used to hold data. The *Itrlist<E>* inner class has been implemented.In addition to classic linked list structure this structure has lazy deletion feature.To implement this class I extend to *AbstractSequentialList<E>* so that ıt contain classic linkedlist methode.Furthemore *Itrlis<E>* class implemented to *Listiterator<E>* interface and all function override again and checked.but didn't used in code because in this assignment I have compare time complexity of iterator and classic linkedlists structure that contain lazy deletion.for lazy deletion I have initialized a counter .if marked node number be 2 I have deleted all marked node.after deletion operation I have decreased the size.

THEORITICAL TIME COMPLEXITIES

***Scenario 1-2-3**

new_post() **: Add post into Post array**

Array:**O(n)** → I have used dynamic array and size of the array allway equal to size of the element

Arraylist:**O(1)** → used add(E item) function add it's complexity equal to O(1)

Linkedlist:**O(1)** → link new item to last of the list by using addLast(E item)

LDLinkedlist:**O(1**) → implemented double linked list so that used can add new element to end of the list

new_following() **: firs it check user exist in following list then add it following list**

Array:**O(n)**

Arraylist:**O(n)**

Linkedlist:**O(n)**

LDLinkedlist:**O(n)**

→Total operation for each structure is different actually.while Array has 4n operation , Linked list has 2n operation. But time complexity is equal for all of them.All of them lineer time complexity

visit_profile() **: in this methode There is a loop and it turn n time**

Array:**O(n)** → in this loop by index operator we can acces item so that T(n) will be O(n)

Arraylist:**O(n^2)** → in this loop by get merthod (O(n)) we can acces item so that T(n) will be O(n^2)

Linkedlist:**O(n)** → in this loop by iterator we can acces item so that T(n) will be O(n)

LDLinkedlist:**O(n^2)** → in this loop by get merthod (O(n)) we can acces item so that T(n) will be O(n^2)

Get_post_information() **: in this methode There is a loop and it turn n time**

Array:**O(n)** → in this loop by index operator we can acces item so that T(n) will be O(n)

Arraylist:**O(n^2)** → in this loop by get method (O(n)) we can acces item so that T(n) will be O(n^2)

Linkedlist:**O(n)** → in this loop by iterator we can acces item so that T(n) will be O(n)

LDLinkedlist:**O(n^2)** → in this loop by get merthod (O(n)) we can acces item so that T(n) will be O(n^2)

like_post():**beside of the check operation there is anarr add operation in this methode**

Array:**O(n)** → check operation has O(n) complexity,add operation has O(n) complexity

Arraylist :**O(n^2)**→ check operation has O(n^2) complexity because get function used in loop that run n time ,add operation has O(1) complexity

LinkedList :**O(n)** → check operation has O(n) complexity help of iterator ,add operation has O(1) complexity

LDLinkedlist :**O(n^2)** → check operation has O(n^2) complexity because get function used in loop that run n time ,add operation has O(1) complexity

add_comment():**it takes post from list and insert it a new like**

Array: **O(n)** → taking post complexity is O(1) and insert complexity is O(n)

Arraylist:**O(1)** → taking post complexity is O(1) and insert complexity is O(1)

Linkedlist : **O(n)** → taking post complexity is O(n) and insert complexity is O(1)

LDLinkedlist : **O(n)** → taking post complexity is O(n) and insert complexity is O(1)

add_outbox(): **it check user name by loop and add new message to outbox**

Array:**O(n)** → check operation has O(n) complexity,add operation has O(n) complexity

Arraylist :**O(n)**→ check operation has O(n) ,add operation has O(1) complexity

LinkedList :**O(n)** → check operation has O(n) complexity,add operation has O(1) complexity

LDLinkedlist :**O(n^2)** → check operation has O(n^2) complexity because get function used in loop that run n time ,add operation has O(1) complexity

Get_interaction_information():**print each post's like and comment information**

Array:**O(n^2)**

Arraylist:**O(n^2)**

Linkedlist:**O(n^3)**

LDLinkedlist:**O(n^3)**

→first loop in this methode using for accest to posts .second loop for accest to post's like and comment objects.But for Linkedlist and LDLinkedlist we have used to get methode(O(n)) for take like and commnets object

***Scenario 4**

delete_following():this method check user existing and remove it from list

Arraylist:**O(n^2)** → check operation complexity equals to O(n) but remove object from list by object equality has O(n^2) complexity

Linkedlist:**O(n)** → check operation performed with iterator and it like found remove methode has O(1) complexity

LDlinkedlist:**O(n^2)** → check operation complexity equals to O(n^2) but remove object from list by index equality has O(n) complexity

→if we compare linked list complaxites Linkedlist is more efficient.But we dont forget the lazy deletion affect.owing to lazy deletion removing operation has less operation.In this way we can obtain more efficient removing operation.But there are a few problem in here actually , beacuse of marke operation we must remove all node together.This cause O(n) complexity just remove operation.if we don't use iterator classic Linkedlist we see classic linked list has O(n) complexity.But if we use lazy deletion we decrease operation number.therefore run time will decrease.with this feature lazy deletion be more efficient. With lazy deletion usage of memory will increase . So that if we use lazy deletion we make sure number of item number in list not too much.if we dont care this lazy deletion create too marked node . In my opinion instead of use classic remove operation we should use lazy deletion. But Listiterator has best efficent.So that we should use Listiterator.It give us more efficient time and memory.

delete_comment():this method check user existing then check comment existing .finaly remove it from list

Arraylist:**O(n^2)** → check operation complexity equals to O(n) but remove object from list by object equality has O(n^2) complexity

Linkedlist:**O(n)** → check operation performed with iterator and if like found remove methode has O(1) complexity

LDlinkedlist:**O(n^2)** → check operation complexity equals to O(n^2) but remove object from list by index equality has O(n) complexity

delete_like():this method check user existing then check comment existing .finaly remove it from list

Arraylist:**O(n^2)** → check operation complexity equals to O(n) but remove object from list by object equality has O(n^2) complexity

Linkedlist:**O(n)** → check operation performed with iterator and if like found remove methode has O(1) complexity

LDlinkedlist:**O(n^2)** → check operation complexity equals to O(n^2) but remove object from list by index equality has O(n) complexity

add_ban():**this method check "ban" list then if not exist in the list add it into list**

Arraylist:**O(n)** → check operation complexity equals to O(n) and add object to list by object equality has O(1) complexity

Linkedlist:**O(n)** → check operation complexity equals to O(n) and add object to list by object equality has O(1) complexity

LDlinkedlist:**O(n^2)** → check operation complexity equals to O(n^2) because of get method and add object to list by object equality has O(1) complexity

delete_ban():**this method check "ban" list then if exist in the list remove from list**

Arraylist:**O(n^2)** → check operation complexity equals to O(n) but remove object from list by object equality has O(n^2) complexity

Linkedlist:**O(n)** → check operation performed with iterator and if like found remove methode has O(1) complexity

LDlinkedlist:**O(n^2)** → check operation complexity equals to O(n^2) but remove object from list by index equality has O(n) complexity

GENERAL COMPLEXITY OF THE PROGRAM EXCLUDING INTERACTIONS

Linked List implementation: O(n)

ArrayList implementation: O(n^2)

LDLinkedList implementation: O(n^2)

Hw1 implementation: O(n)

GENERAL COMPLEXITY OF THE PROGRAM INCLUDING INTERACTIONS

Linked List implementation: O(n^3)

ArrayList implementation: O(n^2)

LDLinkedList implementation: O(n^3)

Hw1 implementation: O(n^2)

4.TEST CASES

Firstly, we have to measure the run-time. To do this, I created data structures with 10,100,1000 and 10000 elements, respectively. After the insertion process, I removed all the elements again. (These operations were repeated in all 3 files and taken to the comment line in the Driver codes.)

After testing and measure run-time of the related code Scenario 1,2,3 and 4 run.Content of the scenario 4 is like this

***Scenarıo 4**

- **Create 7 new account that names are user1,user2 … user7.Total account number has been 10**

UNFOLLOW METHODE CHECKED

- **"gokhankaya" login**
- **View "gokhankaya" profile**
- **"gokhankaya" follow all this new account**
- **"gokhankaya" unfollow user1,user2,user3**

- **View "gokhankaya" profile**

UNCOMMNET AND UNLIKE METHODES CHECKED

- **Show "sibelgulmez" interaction**
- **Uncomment "sibelgulmez" post**

- **Unlike “sibelgulmez” post**
- **Show “sibelgulmez” interaction**
- **“gokhankaya” logout**

STORY METHODE CHECKED

- **“sibelgulmez” login**
- **Send message to “gizemsungu”**
- **Show history**
- **Visit “sibelgulmez profile”**
- **Unfollow “gizemsungu”**
- **Show history**
- **Logout**

BLOCK AND UNBLOCK METHODE CHECKED

- **“sibelgulmez” login**
- **Follow “gizemsungu”**
- **Visit herself profile**
- **Logout**
- **“gizemsungu” login**
- **Block “sibelgumez”**
- **Visit herself profile**
- **Visit sibelgulmez profile**
- **Logout**
- **“sibelgulmez” login**
- **Visit and send message to “gizemsungu”**
- **Logout**
- **“gizemsungu” login**
- **Unblock “sibelgulmez”**
- **Logout**
- **Visit profile “gizemsungu”**
- **Send message “gizemsungu”**
- **Logout**

_____

FINISH

_____

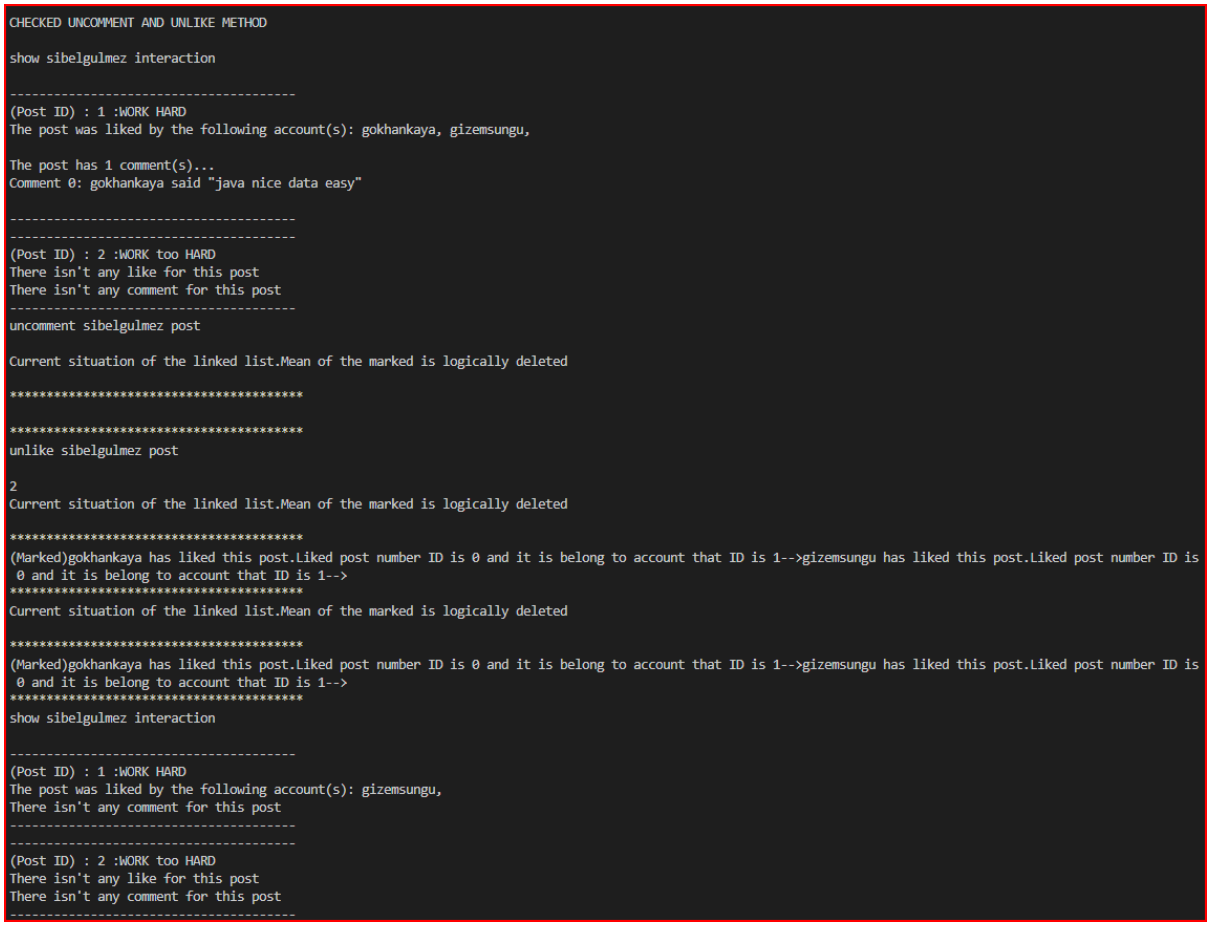
5. RUNNING AND RESULTS

UNFOLLOW

```
view gokhankaya profile
----------------------
User ID: 2
Username: gokhankaya
Location: adana
Birth Date: 10.03.2003
gokhankaya is following 9 Account(s) and has 1 follower
gokhankaya is following sibelgulmez gizemsungu user1 user2 user3 user4 user5 user6 user7
gokhankaya has followed by sibelgulmez
gokhankaya has 0 post

----------------------

FOR EVERY UNFOLOW OPERATION, SITUATION OF THE LINKED LIST WILL DISPLAYED.YOU CAN CHECK LAZY DELETION

Unfollow user1
Current situation of the linked list.Mean of the marked is logically deleted

**************************************
sibelgulmez-->gizemsungu-->(Marked)user1-->user2-->user3-->user4-->user5-->user6-->user7-->
**************************************
Unfollow user2
Current situation of the linked list.Mean of the marked is logically deleted

**************************************
sibelgulmez-->gizemsungu-->user3-->user4-->user5-->user6-->user7-->
**************************************
Unfollow user3
Current situation of the linked list.Mean of the marked is logically deleted

**************************************
sibelgulmez-->gizemsungu-->(Marked)user3-->user4-->user5-->user6-->user7-->
**************************************
view gokhankaya profile
----------------------
User ID: 2
Username: gokhankaya
Location: adana
Birth Date: 10.03.2003
gokhankaya is following 6 Account(s) and has 1 follower
gokhankaya is following sibelgulmez gizemsungu user4 user5 user6 user7
gokhankaya has followed by sibelgulmez
gokhankaya has 0 post

----------------------
```

→When we look this run result we can see , if user remove an account from folowing list it just marked.Not physically remove.situation of the linked list printed the screen between star lines.after user remove 2 account then linked list remove them from list.

UNCOMMNET AND UNLIKE



→ When we look sibelgulmez interaction post Id has 2 like and 1 comment.after display this interaction gokhankaya login and remove a like and a comment.when this operation performed we visit “sibelgulmez” interaction and check comment an like situation.

SHOW STORY

```
CHECKED STORY METHOD

send message gizemsungu

You don's follow this Account so that you can't send message to this profile
show story

You have shared new post
You have shared new post
You have followed gizemsungu
You have followed gokhankaya
gokhankaya has followed you
You have visited gizemsungu's profile
You has liked gizemsungu's post
You have unfollowed gizemsungu
You have visited gizemsungu's profile

Visit sibelgulmez profile

-----------------------
User ID: 1
Username: sibelgulmez
Location: bitlis
Birth Date: 10.03.2002
sibelgulmez is following 1 Account(s) and has 1 follower
sibelgulmez is following gokhankaya
sibelgulmez has followed by gokhankaya
sibelgulmez has 2 post
```

```
----------------------
unfollow gokhan

Current situation of the linked list.Mean of the marked is logically deleted

**************************************

**************************************
Visit sibelgulmez profile

----------------------
User ID: 1
Username: sibelgulmez
Location: bitlis
Birth Date: 10.03.2002
sibelgulmez is following 0 Account(s) and has 1 follower
sibelgulmez is following
sibelgulmez has followed by gokhankaya
sibelgulmez has 2 post

----------------------
show stroy

You have shared new post
You have shared new post
You have followed gizemsungu
You have followed gokhankaya
gokhankaya has followed you
You have visited gizemsungu's profile
You has liked gizemsungu's post
You have unfollowed gizemsungu
You have visited gizemsungu's profile
You have visited sibelgulmez's profile
You have unfollowed gokhankaya
You have visited sibelgulmez's profile

logout
```

UNBLOCK METHODE

```
1.sibelgulmez login

----------------------
User ID: 1
Username: sibelgulmez
Location: bitlis
Birth Date: 10.03.2002
sibelgulmez is following 0 Account(s) and has 1 follower
sibelgulmez is following
sibelgulmez has followed by gokhankaya
sibelgulmez has 2 post

----------------------
follow gizemsungu

visit herself profile

----------------------
User ID: 1
Username: sibelgulmez
Location: bitlis
Birth Date: 10.03.2002
sibelgulmez is following 1 Account(s) and has 1 follower
sibelgulmez is following gizemsungu
sibelgulmez has followed by gokhankaya
sibelgulmez has 2 post

----------------------
sibelgulmez logout
```

→ before the follow gizemsungu, I display the sibelgulmez profile.

```
2. "gizemsungu" logs in.

visit herself profile

----------------------
User ID: 0
Username: gizemsungu
Location: van
Birth Date: 10.03.2001
gizemsungu is following 0 Account(s) and has 2 follower
gizemsungu is following
gizemsungu has followed by gokhankaya sibelgulmez
gizemsungu has 2 post

----------------------
        a.blocks "sibelgulmez".

Current situation of the linked list.Mean of the marked is logically deleted

****************************************
gokhankaya-->
****************************************
Current situation of the linked list.Mean of the marked is logically deleted

****************************************

****************************************
visit herself profile

----------------------
User ID: 0
Username: gizemsungu
Location: van
Birth Date: 10.03.2001
gizemsungu is following 0 Account(s) and has 1 follower
gizemsungu is following
gizemsungu has followed by gokhankaya
gizemsungu has 2 post
```

→when we look “gizemsungu” profile we see she followed by “sibelgulmez”.but after deletion there is no user in her follower such this name

```
----------------------
visit sibelgulmez profile

----------------------
User ID: 1
Username: sibelgulmez
Location: bitlis
Birth Date: 10.03.2002
sibelgulmez is following 0 Account(s) and has 1 follower
sibelgulmez is following
sibelgulmez has followed by gokhankaya
sibelgulmez has 2 post

----------------------
        b. logs out.

3. "sibelgulmez" logs in.

        a. tries to view the profile of "gizemsungu".

THIS ACCOUNT HAS NOT ALREADY EXIST!!
        b. tries to send a message to "gizemsungu".

THIS ACCOUNT HAS NOT ALREADY EXIST!!
        c. logs out.
```

→ when we look sibelgulmez profile she dont follow gizemsungu now

→furthermore if “subelgulmez” want to send message to “gizemsungu” she couldn’t find user has this name

```
        a. "sibelgulmez" remove "gizemsungu" from blocked account list

Current situation of the linked list.Mean of the marked is logically deleted

**************************************

**************************************
        b. logs out.

5. "sibelgulmez" logs in.

        a. tries to view the profile of "gizemsungu".

----------------------
User ID: 0
Username: gizemsungu
Location: van
Birth Date: 10.03.2001
gizemsungu is following 0 Account(s) and has 1 follower
gizemsungu is following
gizemsungu has followed by gokhankaya
gizemsungu has 2 post

----------------------
        b. tries to send a message to "gizemsungu".(CONTENT:HI!!)

You don's follow this Account so that you can't send message to this profile




gizemsungu's inbox


Message ID: 0
From: gokhankaya
To : gizemsungu
Message: slm

Message ID: 1
From: gokhankaya
To : gizemsungu
Message: Hello!
```

→then “gizemsungu” remove “sibelgulmez” from blocked list.When sibel gulmez want to view “gizemsungu” profile, she can see her profile.But if she want to send message to her , she couldn’t do this.Because she dont follow gizem sungu

EXPERIMENTAL RUNING TIME

Implementation Type	Scenario 1	Scenario 2	Scenario 3	Scenario 4	
Basic Array Structure (HW1)	0,1380624	0,011687	0.0103912	Not available	
Array List Structure (a)	0.0939936	0.145258	0.136808	0.1130134	
Linked List Structure (b)	0.092785	0.0165621	0.0230096	0.0514189	
LD Linked List Structure (c)	0.1128341	0.0114451	0.0134973	0.1018799	

TESTING COMPLEXITIES

```
Account[] all_account=new Account[10];
int counter=0;
for(int i=0;i<10;i++)
{
    all_account[i]=new Account(Integer.toString(++counter), location:"null", Birthday:null, counter);
}
all_account[0].login();
long lazy_start=System.nanoTime();
for(int i=1;i<10;i++)
    all_account[0].new_following(all_account[i]);

for(int i=1;i<10;i++)
    all_account[0].delete_following(all_account[i]);

// long start_time1=System.nanoTime();
long lazy_finish=System.nanoTime();
long totalTime_lazy = lazy_finish - lazy_start;
System.out.println( "\n\ntested element number equals 10:"+"\u001B[32m"+totalTime_lazy/1e9+"\u001B[0m\n\n");
all_account[0].logout();
```

→By using this code ;complexity of the three data structured checked and their running time has been measured.After the testing this code it has taked comment line and TestClass has runned.

RUNNING TIME OF THE STRUCTURES

For testing I have created array that contain 10,100,1000,10000 element.after that I have add them to relating to data structure.after that removed from the structure .During this operations I have measured run-time.Output are like this:

Linkedlist:

```
tested element number equals 10:0.0132386
```

```
tested element number equals 100:0.0154774
```

```
tested element number equals 1000:0.0338803
```

```
tested element number equals 10000:1.1124713
```

Arraylist:

```
tested element number equals 10:0.0130871
```

```
tested element number equals 100:0.0142859
```

```
tested element number equals 1000:0.0370334
```

```
tested element number equals 10000:0.8981249
```

LDLinkedlist:

```
tested element number equals 10:0.0142153
```

```
tested element number equals 100:0.0161668
```

```
tested element number equals 1000:0.039993
```

```
tested element number equals 10000:1.0599714
```