

1 Account.java

```
1 package socialmedia;
2 import java.util.ArrayList;
3 import java.io.*;
4
5 public class Account implements java.io.Serializable{
6     /* The Account class that is serializable so that i can be saved into a file later.
7     * Has ID, username, and description as requested. Has an arraylist of posts(original), ednorsements and
8     * comments
9     * to more easily keep track of their count and simplify the way they will be shown. It has a description
10    * field which is never used because the methods never had the the arguments for it but I added it because
11    * it is in the specs. They it has been given a get and set method if needed and there is a constructor
12    * method for it.
13    */
14    private int ID;
15    private String username;
16    private String description;
17    private ArrayList<Original> posts = new ArrayList<>();
18    private ArrayList<Endorsement> endorsements = new ArrayList<>();
19    private ArrayList<Comment> comments = new ArrayList<>();
20
21    Account(int id, String uname){
22        this.ID = id;
23        this.username = uname;
24        this.description = "";
25    }
26    Account(int id, String uname, String desc){
27        this.ID = id;
28        this.username = uname;
29        this.description = desc;
30    }
31
32    public void setID(int id){
33        this.ID = id;
34    }
35    public int getID(){
36        return this.ID;
37    }
38    public void setUsername(String uname){
39        this.username = uname;
40    }
41    public String getUsername(){
42        return this.username;
43    }
44    public void setDescription(String desc){
45        this.description = desc;
46    }
47    public String getDescription(){
48        return this.description;
49    }
50    public int getPostCount(){
51        return posts.size() + endorsements.size() + comments.size();
52    }
53 }
```

```

51 public void addOriginal(Original post){
52     posts.add(post);
53 }
54 public void addEndorsement(Endorsement post){
55     endorsements.add(post);
56 }
57 public void addComment(Comment post){
58     comments.add(post);
59 }
60 public ArrayList<Original> getPosts(){
61     return this.posts;
62 }
63 public ArrayList<Endorsement> getEndorsements(){
64     return this.endorsements;
65 }
66 public ArrayList<Comment> getComments(){
67     return this.comments;
68 }
69
70 public int getRootIDPost(int id){
71     for (Comment i: comments){
72         if (i.getID() == id){
73             return i.getRootID();
74         }
75     }
76     for (Endorsement i: endorsements){
77         if (i.getID() == id){
78             return i.getRootID();
79         }
80     }return -1;
81 }
82 public Original getPost(int pid){
83     for (Original i : posts){
84         if (i.getID() == pid){
85             return i;
86         }
87     }
88     return null;
89 }
90 public Comment getComment(int cid){
91     for (Comment i : comments){
92         if (i.getID() == cid){
93             return i;
94         }
95     }
96     return null;
97 }
98 public Endorsement getEndorsement(int eid){
99     for (Endorsement i : endorsements){
100         if (i.getID() == eid){
101             return i;
102         }
103     }
104     return null;
105 }

```

```

106 public int getFirstPostID(){
107     return this.posts.get(0).getID();
108 }
109 public int getFirstEndorsementID(){
110     return this.endorsements.get(0).getID();
111 }
112 public int getFirstCommentID(){
113     return this.comments.get(0).getID();
114 }
115 public int getEndorsementCount(){
116     int count=0;
117     for (Original i : posts){
118         count += i.getLenEndorsements();
119     }
120     for (Comment x : comments){
121         count += x.getLenEndorsements();
122     }
123     return count;
124 }
125 public int [] getMostEndorsed(){
126     int [] temp = {-1, -1};
127     for (Original i : posts){
128         if(i.getLenEndorsements() > temp[0]){
129             temp[0] = i.getLenEndorsements();
130             temp[1] = i.getID();
131         }
132     }
133     for (Comment x : comments){
134         if(x.getLenEndorsements() > temp[0]){
135             temp[0] = x.getLenEndorsements();
136             temp[1] = x.getID();
137         }
138     }
139     return temp;
140 }
141
142 public void endorsePost(Endorsement en, int id){
143     for (Original i : posts){
144         if (i.getID() == id){
145             i.addEndorsement(en);
146         }
147     }for (Comment x : comments){
148         if(x.getID() == id){
149             x.addEndorsement(en);
150         }
151     }
152 }
153 public void commentPost(Comment com, int id){
154     for (Original i:posts){
155         if(i.getID() == id){
156             i.addComment(com);
157         }
158     }
159 }
160 public void removeEndorsement(int id){

```

```

161     int temp = 0;
162     for (int i=0; i<endorsements.size(); i++){
163         if(id == endorsements.get(i).getID()){
164             temp = i;
165             break;
166         }
167     }
168     endorsements.remove(temp);
169 }
170 public void removeEndorsementFromPosts(int id, int eid){
171     for (Original i : posts){
172         if (i.getID() == id){
173             i.removeEndorsement(eid);
174         }
175     }
176 }
177 public void removeComment(int id){
178     int temp = 0;
179     for (int i=0; i<comments.size(); i++){
180         if(id == comments.get(i).getID()){
181             temp = i;
182             break;
183         }
184     }
185     comments.remove(temp);
186 }
187
188 public void removeCommentFromPosts(int id, int cid){
189     for (Original i : posts){
190         if (i.getID() == id){
191             i.removeComment(cid);
192         }
193     }
194 }
195 public void removeOriginal(int id){
196     for (Original i : posts){
197         if (i.getID() == id){
198             i.setBody("The original content was removed from the system and is no longer available.");
199             i.setAuthorID(0);
200         }
201     }
202     int temp = 0;
203     for (int i=0; i<posts.size(); i++){
204         if(id == posts.get(i).getID()){
205             temp = i;
206             break;
207         }
208     }
209     posts.remove(temp);
210 }
211
212 }

```

2 Post.java

```
1 package socialmedia;
2
3 import java.io.*;
4
5 public class Post implements java.io.Serializable{
6     /* This is the parent class Post for the 3 types of post, Original, Endorsement and Comment. It is also
7     * serializable and therefore the children will be too. It has attributes ID, authorID, and body.
8     */
9     private int ID;
10    private int authorID;
11    private String body;
12
13    Post(int id, int aid, String text){
14        this.ID = id;
15        this.authorID = aid;
16        this.body = text;
17    }
18
19    public void setID(int id){
20        this.ID = id;
21    }
22    public int getID(){
23        return this.ID;
24    }
25    public void setAuthorID(int aid){
26        this.authorID = aid;
27    }
28    public int getAuthorID(){
29        return this.authorID;
30    }
31    public void setBody(String text){
32        this.body = text;
33    }
34    public String getBody(){
35        return this.body;
36    }
37    public int getRootID(){
38        return -1; //This method is here just so that all Post children inherit it and it can be called from an
39        //arraylist of posts in the MiniSocialMedia
40        //if it returns -1 it has to be the original post ID
41    }
42 }
```

3 Original.java

```
1 package socialmedia;
2
3 import java.util.ArrayList;
4 public class Original extends Post{
5     /* This is the Original class which extends the Post class. It has added attributes Endorsements and
6     Comments
7     * to keep track of the number of times this post has been endoresed or commented. In addition to make
```

```

import junit.framework.TestCase;
7  * easier to recreate when needed.
8  */
9  private ArrayList <Endorsement> endorsements = new ArrayList<>();
10 private ArrayList <Comment> comments = new ArrayList<>();
11
12
13 Original(int id, int aid, String text){
14     super(id, aid, text);
15 }
16 public void setEndorsements(ArrayList<Endorsement> endors){
17     this.endorsements = endors;
18 }
19 public void addEndorsement(Endorsement endor){
20     this.endorsements.add(endor);
21 }
22 public ArrayList <Endorsement> getEndorsements(){
23     return this.endorsements;
24 }
25 public int getLenEndorsements(){
26     return this.endorsements.size();
27 }
28 public void removeEndorsement(int id){
29     int temp = 0;
30     for (int i=0; i<endorsements.size(); i++){
31         if(id == endorsements.get(i).getID() ){
32             temp = i;
33             break;
34         }
35     }
36     endorsements.remove(temp);
37 }
38 public Boolean checkDoubleEndorsements(int id){
39     for (Endorsement i: endorsements){
40         if(i.getAuthorID() == id){
41             return false;
42         }
43     }return true;
44 }
45 public void setComments(ArrayList<Comment> comms){
46     this.comments = comms;
47 }
48 public void addComment(Comment comms){
49     this.comments.add(comms);
50 }
51 public ArrayList<Comment> getComments(){
52     return this.comments;
53 }
54 public int getLenComments(){
55     return this.comments.size();
56 }
57 public void removeComment(int id){
58     for (Comment i : comments){
59         if(i.getID() == id){
60             i.setBody("The original content was removed from the system and is no longer available.");

```

```

61         i.setAuthorID(0);
62     }
63 }
64 }
65 public Boolean isLastComment(int id){
66     if (comments.get(this.comments.size()-1).getID() == id){
67         return true;
68     }else{
69         return false;
70     }
71 }
72 }
73 }

```

4 Endorsement.java

```

1 package socialmedia;
2
3 public class Endorsement extends Post{
4     /* The Endorsement extends post adding a rootID attribute to know which post it is endorsing.
5     */
6     private int rootID;
7
8     Endorsement(int id, int aid, String text, int rid){
9         super(id, aid, text);
10        this.rootID = rid;
11    }
12    public void setRootID(int rid){
13        this.rootID = rid;
14    }
15    @Override
16    public int getRootID(){
17        return this.rootID;
18    }
19 }

```

5 Comment.java

```

1 package socialmedia;
2
3 import java.util.ArrayList;
4
5 public class Comment extends Post{
6     /* This is the Comment class which extends the Post class. There is a rootID attribute and an arraylist of
7     *endorsements added to know to which Original Post a comment points too so that it can later be recreated
8     * As comments can be endorsed there is an arraylist of them.
9     */
10    private int rootID;
11    private ArrayList <Endorsement> endorsements = new ArrayList<>();
12
13    Comment(int id, int aid, String text, int rid){
14        super(id, aid, text);
15        this.rootID = rid;
16    }

```

```

17     public void setRootID(int rid){
18         this.rootID = rid;
19     }
20     @Override
21     public int getRootID(){
22         return this.rootID;
23     }
24     public void setEndorsements(ArrayList<Endorsement> endors){
25         this.endorsements = endors;
26     }
27     public void addEndorsement(Endorsement endor){
28         this.endorsements.add(endor);
29     }
30     public ArrayList <Endorsement> getEndorsements(){
31         return this.endorsements;
32     }
33     public int getLenEndorsements(){
34         return this.endorsements.size();
35     }
36     public void removeEndorsement(int id){
37         for (Endorsement i : endorsements){
38             if(i.getID() == id){
39                 endorsements.remove(i);
40             }
41         }
42     }
43 }

```

6 MiniSocialMedia.java

```

1  package socialmedia;
2
3  import java.io.IOException;
4  import java.util.ArrayList;
5  import java.util.Arrays;
6  import java.io.*;
7
8  /**
9   * MiniSocialMedia compiles by implementing the MiniSocialMediaPlatform interface
10  *
11  * @author Jakub Davison
12  * @version 1.0
13  */
14  public class MiniSocialMedia implements MiniSocialMediaPlatform {
15      static Account delAcc = new Account(0, "admin");
16      static ArrayList <Account> accounts = new ArrayList<>(Arrays.asList(delAcc));
17      static ArrayList <Post> allPosts = new ArrayList<>();
18
19      public int getNumberOfAccounts(){
20          //returns the number of accounts
21          return accounts.size();
22      }
23      public int getTotalOriginalPosts(){
24          //returns the total number of original Posts created by all users

```



```

25     int counter = 0;
26     for (Post i : allPosts){
27         if (i instanceof Original){
28             counter ++;
29         }
30     }return counter;
31 }
32 public int getTotalEndorsmentPosts(){
33     //returns the total number of Endorsement Posts created by all users
34     int counter = 0;
35     for (Post i : allPosts){
36         if (i instanceof Endorsement){
37             counter ++;
38         }
39     }return counter;
40 }
41 public int getTotalCommentPosts(){
42     //returns the total number of Comments created by all users
43     int counter = 0;
44     for (Post i : allPosts){
45         if (i instanceof Comment){
46             counter ++;
47         }
48     }return counter;
49 }
50
51 @Override
52 public int createAccount(String handle) throws IllegalHandleException, InvalidHandleException {
53     int x=handle.length();
54     if (x > 30){
55         throw new InvalidHandleException ("The handle you have entered is longer than 30 characters. Pelase
56             try again.");
57     }else if (x == 0){
58         throw new InvalidHandleException ("The handle you have entered is empty. Pelase try again.");
59     }else if(handle.indexOf(" ") != -1){
60         throw new InvalidHandleException ("The handle you have entered contains a white space. Pelase try
61             again.");
62     }
63     if(accounts.size()>0){
64         for (Account i : accounts){
65             if (i.getUsername()==handle){
66                 throw new IllegalHandleException("The handle you have entered is already in use. Pelase try
67                     again.");
68             }
69         }
70         int id = this.accounts.get(accounts.size()-1).getID() +1;
71         Account acc = new Account(id, handle);
72         accounts.add(acc);
73         return id;
74     }
75     else{
76         Account acc = new Account(1, handle);
77         accounts.add(acc);
78         return 1;
79     }
80 }

```

```

77     }
78
79     @Override
80     public void removeAccount(int id) throws AccountIDNotRecognisedException {
81         if(id==0){
82             throw new AccountIDNotRecognisedException ("You tried to delete the admin account. Please try again");
83         }
84         for (Account i : accounts){
85             if (i.getID() == id){
86                 int temp=-1;
87                 try{
88                     for (int x = 0; x<i.getPosts().size(); x++){
89                         temp = i.getFirstPostID();
90                         deletePost(temp);
91                     }
92                     for (int x = 0; x<i.getEndorsements().size(); x++){
93                         temp = i.getFirstEndorsementID();
94                         deletePost(temp);
95                     }
96                     for (int x = 0; x<i.getComments().size(); x++){
97                         temp = i.getFirstCommentID();
98                         deletePost(temp);
99                     }
100                 } catch(PostIDNotRecognisedException e){
101                     System.out.print(e);
102                 }
103                 accounts.remove(i);
104                 return;
105             }
106         }
107         throw new AccountIDNotRecognisedException ("The id isnt recognized " + id);
108     }
109
110     @Override
111     public void changeAccountHandle(String oldHandle, String newHandle)
112         throws HandleNotRecognisedException, IllegalHandleException, InvalidHandleException {
113         int x=newHandle.length();
114         if (x > 30){
115             throw new InvalidHandleException ("The new handle you have entered is longer than 30 characters.
116                 Pelase try again.");
117         }else if (x == 0){
118             throw new InvalidHandleException ("The new handle you have entered is empty. Pelase try again.");
119         }else if(newHandle.indexOf(" ") != -1){
120             throw new InvalidHandleException ("The new handle you have entered contains a white space. Pelase
121                 try again.");
122         }
123         if(accounts.size(>0){
124             for (Account i : accounts){
125                 if (i.getUsername()==newHandle){
126                     throw new IllegalHandleException("The new handle you have entered is already in use. Pelase
127                         try again.");
128                 }
129             }
130             Account acc=null;
131             for (Account i : accounts){

```

```

129         if (i.getUsername() == oldHandle){
130             acc=i;
131         }
132     }if(acc==null){throw new HandleNotRecognisedException("The handle you have entered does not match
133         any in our records. Please try again.");}
134     acc.setUsername(newHandle);
135 }
136
137 @Override
138 public String showAccount(String handle) throws HandleNotRecognisedException {
139     String output;
140     Account acc=null;
141     for (Account i : accounts){
142         if (i.getUsername() == handle){
143             acc=i;
144         }
145     }if(acc==null){throw new HandleNotRecognisedException("The handle you have entered does not match any
146         in our records. Please try again.");}
147     output = "ID: " + acc.getID() + "\nHandle: " + acc.getUsername() + "\nDescription: " +
148         acc.getDescription() + "\nPost count:" + acc.getPostCount() + "\nEndorsement count: " +
149         acc.getEndorsementCount();
150     return output;
151 }
152
153 @Override
154 public int createPost(String handle, String message) throws HandleNotRecognisedException,
155     InvalidPostException {
156     int x=message.length();
157     if (x > 100){
158         throw new InvalidPostException ("The message you have entered is longer than 100 characters. Pelase
159             try again.");
160     }else if (x == 0){
161         throw new InvalidPostException ("The message you have entered is empty. Pelase try again.");
162     }
163     Account acc=null;
164     for (Account i : accounts){
165         if (i.getUsername() == handle){
166             acc=i;
167         }
168     }if(acc==null){throw new HandleNotRecognisedException("The handle you have entered does not match any
169         in our records. Please try again.");}
170     if(allPosts.size() == 0){
171         Original og = new Original(1, acc.getID(), message);
172         acc.addOriginal(og);
173         allPosts.add(og);
174         return og.getID();
175     }
176     Original og = new Original(allPosts.get(allPosts.size()-1).getID()+1, acc.getID(), message);
177     acc.addOriginal(og);
178     allPosts.add(og);
179     return og.getID();
180 }

```

```

177 @Override
178 public int endorsePost(String handle, int id)
179     throws HandleNotRecognisedException, PostIDNotRecognisedException, NotActionablePostException {
180     for (Post i: allPosts){
181         if(i.getID() == id){
182             if(i instanceof Endorsement){
183                 throw new NotActionablePostException ("The post you are trying to endorse is already an
184                     endorsement. Please try again.");
185             }
186             Account acc=null;
187             for (Account x : accounts){
188                 if (x.getUsername() == handle){
189                     acc=x;
190                 }
191             }if(acc==null){throw new HandleNotRecognisedException("The handle you have entered does not
192                 match any in our records. Please try again.");}
193
194             int uid =acc.getID();
195             String text = "EP@" + i.getAuthorID() + ": " + i.getBody();
196             Endorsement en = new Endorsement (allPosts.get(allPosts.size()-1).getID()+1, acc.getID(), text,
197                 id);
198             for (Account x : accounts){
199                 if (x.getID() == i.getAuthorID()){
200                     if (x.getPost(id).checkDoubleEndorsements(uid) == false){
201                         return -1;//if this user already endorsed the post then reutnr -1 so no new endorsement is
202                             created and the platform is unchanged
203                     }
204                     allPosts.add(en);
205                     acc.addEndorsement(en);
206                     x.endorsePost(en, id);
207                 }
208             }
209             return en.getID();
210         }
211     }throw new PostIDNotRecognisedException ("The ID isnt recognized " + id);
212 }
213
214 @Override
215 public int commentPost(String handle, int id, String message) throws HandleNotRecognisedException,
216     PostIDNotRecognisedException, NotActionablePostException, InvalidPostException {
217     if (message.length() > 100){
218         throw new InvalidPostException ("The comment you are trying to post has more than 100 characters.
219             Please try again.");
220     }
221     else if(message.length() == 0){
222         throw new InvalidPostException ("The comment youa re trying to post is empty. Please try again.");
223     }
224     for (Post i: allPosts){
225         if(i.getID() == id){
226             if(i instanceof Endorsement){
227                 throw new NotActionablePostException ("The post you are trying to endorse is already an
228                     endorsement. Please try again.");
229             }
230         }
231     }
232     Account acc=null;

```

```

226     for (Account x : accounts){
227         if (x.getUsername() == handle){
228             acc=x;
229         }
230     }if(acc==null){throw new HandleNotRecognisedException("The handle you have entered does not match
any in our records. Please try again.");}
231     int uid = acc.getID();
232     Comment com;
233     if (i instanceof Comment){
234         com = new Comment(allPosts.get(allPosts.size()-1).getID()+1, acc.getID(), message,
acc.getRootIDPost(i.getID()));
235     }else{
236         com = new Comment(allPosts.get(allPosts.size()-1).getID()+1, acc.getID(), message, id);
237     }
238     allPosts.add(com);
239     acc.addComment(com);
240     for (Account x : accounts){
241         if (x.getID() == i.getAuthorID()){
242             x.commentPost(com, id);
243         }
244     }
245     return com.getID();
246 }
247 }throw new PostIDNotRecognisedException ("The ID isnt recognized " + id);
248 }
249
250 @Override
251 public void deletePost(int id) throws PostIDNotRecognisedException {
252     for (Post i: allPosts){
253         if(i.getID() == id){
254             Account acc=null;
255             int aid = i.getAuthorID();
256             for (Account x : accounts){
257                 if (aid == x.getID()){
258                     acc=x;
259                 }
260             }if (acc==null){return;};
261             if(i instanceof Original){
262                 acc.removeOriginal(id);
263                 i.setBody("The original content was removed from the system and is no longer available.");
264                 i.setAuthorID(0);
265                 accounts.get(0).addOriginal((Original) i);
266                 return;
267             }
268             for (Post x : allPosts){
269                 int rootID = acc.getRootIDPost(id);
270                 if (x.getID() == rootID){
271                     for (Account z : accounts){
272                         if (z.getID() == x.getAuthorID()){
273                             if (i instanceof Endorsement){
274                                 acc.removeEndorsement(id);
275                                 z.removeEndorsementFromPosts(rootID, id);
276                                 allPosts.remove(i);
277                                 return;
278                             }else if(i instanceof Comment){

```

```

279         acc.removeComment(id);
280         i.setBody("The original content was removed from the system and is no longer available.");
281         i.setAuthorID(0);
282         accounts.get(0).addComment((Comment) i);
283         return;
284     }
285 }
286 }
287 }
288 }
289 }
290 }throw new PostIDNotRecognisedException ("The ID isnt recognized " + id);
291 }
292
293 @Override
294 public String showIndividualPost(int id) throws PostIDNotRecognisedException {
295     for (Post i : allPosts){
296         if (i.getID() == id){
297             Account acc=null;
298             for (Account x : accounts){
299                 if (x.getID() == i.getAuthorID()){
300                     acc = x;
301                     break;
302                 }
303             }if (acc==null){return "";}
304             String text ="ID: " + i.getID() + " \nAccount: " + acc.getUsername();
305             if(i instanceof Endorsement){
306                 return text;
307             }else if (i instanceof Comment){
308                 text = text + " \nNo. endorsements: " + acc.getComment(id).getLenEndorsements();
309                 int temp = i.getRootID();
310                 int authorOGID = -1;
311                 int postOGID = -1;
312                 while (true){
313                     for (Post z : allPosts){
314                         if (z.getID() == temp){
315                             temp = z.getRootID();
316                             authorOGID = z.getAuthorID();
317                             postOGID = z.getID();
318
319                             break;
320                         }
321                     }
322                     if(temp == -1){
323                         break;
324                     }
325                 }
326                 for (Account z : accounts){
327                     if (z.getID() == authorOGID){
328                         if(z.getPost(postOGID).isLastComment(id) == true){
329                             text = text + " | No. comments: " + 0 ;
330                         }
331                         else{
332                             text = text +" | No. comments: " + 1;
333                         }

```

```

334     }
335 }
336 }
337 else{
338     text = text + " \nNo. endorsements: " + acc.getPost(id).getLenEndorsements() + " | No. comments: "
339         + acc.getPost(id).getLenComments();
340 }
341 text = text + " \n" + i.getBody() + " \n";
342 return text;
343 }
344 }throw new PostIDNotRecognisedException ("The ID isnt recognized " + id);
345 }
346
347 @Override
348 public StringBuilder showPostChildrenDetails(int id)
349     throws PostIDNotRecognisedException, NotActionablePostException {
350     for (Post i : allPosts){
351         if (i.getID() == id){
352             StringBuilder sb=new StringBuilder();
353             if (i instanceof Endorsement){
354                 throw new NotActionablePostException ("The post you entered has no children because it is an
355                     Endorsement. Please try again");
356             } else if (i instanceof Original){
357                 sb.append(showIndividualPost(id));
358                 Original og = (Original) i;
359                 for (Comment c : og.getComments()){
360                     sb.append("\n| > " + showIndividualPost(c.getID()));
361                 }
362                 sb = indent(sb);
363                 return sb;
364             } else{
365                 Comment com = (Comment) i;
366                 for (Post x : allPosts){
367                     if (x.getID() == com.getRootID()){
368                         sb.append(showIndividualPost(id));
369                         Original og = (Original) x;
370                         for (Comment c : og.getComments()){
371                             if (c.getID() > id){
372                                 sb.append("\n| > " + showIndividualPost(c.getID()));
373                             }
374                         }
375                         sb = indent(sb);
376                         return sb;
377                     }
378                 }
379             }
380         }
381     }
382     }throw new PostIDNotRecognisedException ("The ID isnt recognized " + id);
383 }
384
385 public StringBuilder indent (StringBuilder sb){
386     //This indents the Stringbuilder by the appropriate ammount as per the assignment
387     String temp = sb.toString();
388     String splits [];
389     sb.setLength(0);
390     splits = temp.split("\n");
391     int counter =0;

```

```

387     for(String i : splits){
388         for (int x=0; x<counter; x++){
389             i = "    " + i;
390         }
391         sb.append(i + "\n");
392         if (i.indexOf("| > ") !=-1){counter++;}
393     }
394     return sb;
395 }
396 @Override
397 public int getMostEndorsedPost() {
398     int[] values = {-1, -1};
399     for (Account i : accounts){
400         int[] temp = i.getMostEndorsed();
401         if (temp[0] > values[0]){
402             values = temp;
403         }
404     }
405     return values[1];
406 }
407
408 @Override
409 public int getMostEndorsedAccount() {
410     int id=-1;
411     int temp = -1;
412     for (Account i : accounts){
413         if(i.getEndorsementCount() > temp){
414             temp = i.getEndorsementCount();
415             id = i.getID();
416         }
417     }
418     return id;
419 }
420
421 @Override
422 public void erasePlatform() {
423     try{
424         new FileWriter("socialmedia/classes.ser", false).close();
425     } catch(IOException e){
426         System.out.print(e);
427     }
428 }
429
430 @Override
431 public void savePlatform(String filename) throws IOException {
432     try(ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream (filename))){
433         for (Post i : allPosts){
434             out.writeObject(i);
435         }
436         for(Account i : accounts){
437             out.writeObject(i);
438         }
439         Object a=null;
440         out.writeObject(a);
441     }

```



```

442     }
443
444     @Override
445     public void loadPlatform(String filename) throws IOException, ClassNotFoundException {
446         if(new File(filename).length() == 0){return;}
447         FileInputStream file = new FileInputStream(filename);
448         ObjectInputStream in = new ObjectInputStream(file);
449         Object obj = in.readObject();
450         Post p1 = null;
451         Account a1 = null;
452         while (obj != null){
453             try{
454                 if (obj instanceof Original){
455                     p1 = (Original) obj;
456                     allPosts.add(p1);
457                 }else if(obj instanceof Endorsement){
458                     p1 = (Endorsement) obj;
459                     allPosts.add(p1);
460                 }else if(obj instanceof Comment){
461                     p1 = (Comment) obj;
462                     allPosts.add(p1);
463                 }else if(obj instanceof Account){
464                     a1 = (Account) obj;
465                     accounts.add(a1);
466                 }
467                 obj = in.readObject();
468             } catch (EOFException eofEx){
469                 in.close();
470                 break;
471             }
472         }
473     }
474 }

```