



# CS352 – Software Engineering II

## Phase 3-a: Review Report

### 2015

Project Team - Desoky AbdElqawy

ID	Name	Email	Mobile
20120516	Heba Ahmed Khazbak	heba.khazbak@gmail.com	01009595942
20120185	Ziad Mohamed	Ziad_mohamed9473@hotmail.com	01271777806
20120191	Sarah Hany Tammam	sarahHtammam@gmail.com	01008332618
20120166	Dalia Maher Mohamed	dalia.maher94@gmail.com	01111267812

### The Team That We Reviewed

<https://github.com/Bazma/FCL-CS352-MiraAyman>

20120412	ميرا ايمن حمدى زكى عبد الله
----------	-----------------------------



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

#### Contents

Review Check List .....	3
List of Issues Found.....	4
Adherence to SOLID Principles .....	8
Suggestions for improvements.....	8
Team Member's Contributions .....	10



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

#### Review Check List

##### Design and Code Checklist

###### Design Principles

- |   |           |                        |
|---|-----------|------------------------|
| 1- Does the design follow SOLID principles? | <u>  </u> | 30 % ..... 1,5,6 ..... |
| 2- Does the design follow OOP rules?        | <u>  </u> | 70% ..... 3,4 .....    |
| 3- Is the design simple and easy to modify? | <u>  </u> | 30% ..... 1,2 .....    |

###### Coding Standards

- |   |           |                                  |
|---|-----------|----------------------------------|
| 4- Is the code understandable and readable? | <u>  </u> | 60 % ..... 11 .....              |
| 5- Does the code follow Java Coding Style?  | <u>  </u> | 40% ..... 7,9,12,17 .....        |
| 6- Is indentation used properly?            | <u>  </u> | 85% ..... 18 .....               |
| 7- Do variables have good names?            | <u>  </u> | 60% ..... 8,11,13,14,15,16 ..... |

###### Comments

- |  |           |                              |
|--|-----------|------------------------------|
| 8- Is the code commented enough?           | <u>  </u> | 30 % ..... 24 .....          |
| 9- Is every class and method commented?    | <u>  </u> | 35 % ..... 24.....           |
| 10- Do comments follow Javadoc style?      | <u>  </u> | 50% ..... .....              |
| 11- Is Javadoc generated for all the code? | <u>  </u> | 20% ..... 19,21,22,23 .....  |
| 12- Are there useless / wrong comments?    | <u>  </u> | 70 % ..... 19,21,22,23 ..... |

###### Code Structure

- |   |           |                        |
|---|-----------|------------------------|
| 13- Does the code follow the design precisely?      | <u>  </u> | 30 % ..... .....       |
| 14- Are there very long classes or methods?         | <u>  </u> | 60 % ..... 25,26 ..... |
| 15- Is there repeated code ?(put put in a function) | <u>  </u> | 20 % ..... 36: .....   |

###### Error Handling



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

- 16- Does the code handle errors and exceptions? II 60 % ..... 27 .....
- 17- Is defensive programming used to avoid errors? II 60 % ..... 27 .....

#### Logic

- 18- Do loops have correct conditions and bounds? II 80 % ..... 29.30.31 .....
- 19- Do loops always terminate? II 100% ..... .....

#### Overall

- 20- Are the design and code of good quality? II 52.1%

### List of Issues Found

#### Design Principles

- 1- There is one very very big class ""PostEntity"" which violates single responsibilities and open closed principle. It is responsible for seen , Tag, LikePost , savePost , checkHashtag, ViewPublic!
- 2- No different types of posts, there is only PagePost which extends postBuilder which does almost nothing! There is one function calls ""writePost"" just pass postEntity and string privacy and add this privacy to the postEntity.
- 3- useless class called ""Post"" which is empty!
- 4- abstract class ""privacypost"" and it's subclasses ""privatepost"" , ""publicpost"" , ""custompost"" all they do is setPrivacy ""this.privacypost=""private"";"" and there is redundant string in subclasses. It is better to put it in parent class
- 5- Wrong use of observer design pattern and wrong responsibilities in (NotificationEntity) as there is no attach nor notify functions which are fundamental functions in observer DP .
- 6- Function update in Observer takes ""NotificationEntity"" as parameters for it which is specific type of notification(only message notification because it refers to this in the code). Also violation of open closed principle as if we want to add friendRequestObserver or LikeObserver, we should modify the code.



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

#### Coding Standards

- 7- variables names doesn't follow naming conventions as the second word doesn't start with capital letter. Examples in controller class : currentactive , emailuser , emailfrom. Also functions names in controller class as "responseaddfri()", "responseadd()". Same goes for other classes such as FriendEntity.java( e.g: getemailto , getemailfrom,...)
- 8- Notification class is used for messages only so the name should be more specific
- 9- few classes don't start with capital letter as "privacypost" , ""publicpost"
- 10- The usage of System.out.print to debug should be erased after fixing the bug, however it's still in the services class AddFriendService() and WriteMessagesService()
- 11- Class "Subject" has bad and unclear name and doesn't describe the class functionality.
- 12- in class pageEntity.java the parameters names are all "n" in setters which violates the variable names standards
- 13- This line -> Entity employee = new Entity("conversation"); violated the naming and logic standards because it uses an entity named employee and the function deals with conversation entity, and same problem in NotificationEntity.
- 14- there are two functions in "ConversationEntity.java" that are supposed to get page ID which are:  
public static String GetConversation(String json), and  
public static String Getonversations(String title) , the first one searches using a string 'title' and the second using string json; the naming of the function should have been clearer. Also the same thing with function "showSeen1" and "seen1", "Number" and "KONumberOfHashTag" in "PostEntity.java"
- 15- Improper naming conventions for the classes "SendRequest.java" and "Message.java" which are used to extend command abstract class. It's better if their name contained the word "command" to refer that they actually extends it.
- 16- Same Problem of naming the variables "n", "m", "p" in setters and getters of "Notification.java", "NotificationEntity.java" and in the functions "x", "y", "z" are used in "PostEntity.java".



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

17- Function names should start with a verb, its first letter is small.

18- Missing white spaces and Indentations in some of the code.

#### Comments

19- In the controller class most of the javaDoc is not correct as it describes signup or Login function while it should describe another functions (wrong description). Examples for wrong description functions : Notify , createPage, CreatePost and ect ...

20- There is commented code in the controller class in many functions as TimeLine() , Notification() and also in the services class and the model classes such as FriendEntity.java, MessageEntity.java, PostEntity.java and NotificationEntity.java

21- JavaDoc is not correct in Postpram.java

22- JavaDoc in PageEntity.java, LikePageEntity.java are repeated for all functions; which means javaDoc is totally not right

23- Same problem in “PostEntity.java”, removing the javaDoc is better than leaving it wrong.

24- More comments can be added instead of printing in the code itself.

#### Code Structure

25- There is one big class for all the controllers.

26- There is one big class for all the services

#### Error Handling

27- Little checks before saving the entities to datastore.



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

#### Logic

- 28- In class "Subject" , "attachtolist" function takes list of "ObserveUser" instead of one object. so if we want to add new observer we should put it first in a list !
- 29- In class MessageEntity.java getMessage function will return true even if the messages datastore is empty.
- 30- Same problem in getFriend function in FriendEntity.java
- 31- and same problem to checkPage in PageEntity.java
- 32- Function AcceptRequestFriend in class "Reciver.java", it just returns null so, it should has complete implementation or be removed.

#### General

- 33- There is an error in the Controller and service classes because of the merging (so the project wasn't able to run).
- 34- In the controller class unnecessary parameters are passed as in function "responseaddfri" parameter "emailfrom" is passed then assign its value from current active user. Also some function could use @GET instead of @POST as it takes unnecessary parameters as function "getfriendss"
- 35- In class controller there is unclear function called "AttachNotification(ArrayList<ObserveUser>)", should not be in the controller class as it violates MVC design pattern.
- 36- in class "ConversationEntity" , the function of "getConversations" has redundant code, in addition to the unnecessary messages that will be printed, the return of the function is redundant, because the code could have possibly returned the boolean variable 'check1' instead of adding the if condition .



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

#### Adherence to SOLID Principles

Single Responsibility is violated in many classes like “UserController”, “Service” and “PostEntity”. Post Entity is responsible for everything related to the post (like , seen , HashTag and privacy) which is completely unacceptable. There are many reasons to change. Open/Close principle is also violated. if we want to add new type of notification (other than message), we should modify in many classes.

Existence of many classes that completely violates the ability to extend them, and since they also don't contain the pre and post conditions before the functions included in them, Liskov's Substitution principle is also violated.

General OOP principles, most of the variables are private and using setters and getters which is good.

#### Suggestions for improvements

1. Divide class “UserController” to many classes. each class will be responsible for a component in the social network. for example (MessageController , PostController , NotificationController and AuthenticationController).
2. Divide class “PostEntity” to many classes as this class now is responsible for all types of posts and a lot of other functions
3. Follow coding standards in naming class (should start with capital letter), if the variables name consists of more than one word, the second and third and so on should start with capital letter. for example “currentactive” should be currentActiveUser
4. getMessage function in MessageEntity.java, I suggest changing the following condition if (check==true && check1==true) to if (check==true && check1==false). That way, the function will work properly





## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

5. suggestion 4 will also work in function getFriend in class FriendEntity.java because it's the same mistake.
6. Follow Observer Design pattern correctly in implementing notification component.  
This link may help [http://www.tutorialspoint.com/design\\_pattern/observer\\_pattern.htm](http://www.tutorialspoint.com/design_pattern/observer_pattern.htm)
7. the two classes "Conversation" and "ConversationEntity" can be merged.
8. Use parameter names that can be understandable.
9. Remove commented code that is useless.
10. Make your code a bit more readable with proper indentations and spaces. Hint: you can click "Ctrl + Shift + F" to each file.



## CS352: Phase 3 – CS\_2, SocialNetwork

### Phase 3- Review Report of Team FCI-CS352-MiraAyman-master

#### Team Member's Contributions

Name	ID	Piece of Code Reviewed (Class, Service, function, ....)	The Issues/Suggestions Found (refer to #s above)	Time Spent
Heba Khazbak	20120516	controller class, Class custompost, Observer, ObserveUser, PagePost, Post, Post Builder, privacy post, private post, public post, Subject	Issue: 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 19, 20, 25, 28, 33, 34, 35 Suggestions: 1, 2, 3, 6 Total: 17 issues + 4 sugg.	5 hrs
Ziad Mohamed	20120185	services class, Class FriendEntity, Message, MessageEntity, Postpram, Receiver, SendRequest	Issue: 10, 20, 21, 26, 29, 30 Suggestions: 4, 5 Total: 6 issues + 2 sugg.	1 hr
Sarah Hany	20120191	Conversation, conversationEntity, PageEntity, LikePageEntity, User	Issue: 12, 13, 14, 22, 31, 36 Suggestions: 7, 8 Total: 6 issues + 2 sugg.	3hrs
Dalia Maher	20120166	Services class, Class PostEntity, Receiver, SendRequest, Message, Notification, NotificationEntity, Command, PostEntity, UserPost	Issue: 15, 16, 17, 18, 23, 24, 27, 32 Suggestions: 9, 10 Total: 8 issues + 2 sugg.	3.5 hrs