**Cairo University Faculty of Computers and Information**



**CS352 – Software Engineering II**

**Phase 3-b: Testing 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 |

Contents

[1. Backend Testing 3](#_Toc419064163)

[2. Frontend testing 39](#_Toc419064164)

[Github 50](#_Toc419064165)

[Ownership report 51](#_Toc419064166)

# 1. Backend Testing

* 1. **CustomPrivacyTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.1.1** | **canSeeUserPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeUserPost function in CustomPrivacy to see if current user can see user post**  **Assumption entity: is custom** | **Passed** |
| **1.1.2** | **canSeeFriendPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeFriendPost function in CustomPrivacy to see if current user can see Friend post**  **Assumption entity: is custom** | **Failed** |
| **1.1.3** | **canSeePagePost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeePagePost function in CustomPrivacy to see if current user can see Page post**  **Assumption entity: is custom** |  |
| **1.1.4** | **canSeeSharedPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeSharedPost function in CustomPrivacy to see if current user can see Shared post**  **Assumption entity: is custom** |  |
| **1.1.5** | **isInCustom(boolean result , String postID , String userName)** | **Testing function for isInCustom function in CustomPrivacy to see if userName in specific post ID is allowed to see custom post** |  |

* 1. **NotificationTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.2.1** | **getNotification(String result,String ID)** | **Testing function for getNotification function in Notification to get Notification ID that points to it** | **Passed** |

* 1. **GroupMessageTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.3.1** | **getGroupMsg(Entity result, String ID)** | **Testing function for getGroupMsg function in GroupMessage to get group msg entity with ID** | **Passed** |
| **1.3.2** | **getNewID()** | **Testing function for getNewID function in GroupMessage to get new ID** |  |

* 1. **GroupMsgMemberTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.4.1** | **getAllGroupNames()** | **Testing function for getAllGroupNames function in GroupMsgMember to get all group names** | **Passed** |
| **1.4.2** | **getAllMembers(Set<String> Result ,String GroupName)** | **Testing function for getAllMembersfunction in GroupMsgMember to get all member in specific group** |  |

* 1. **HashtagTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.5.1** | **saveHashtag()** | **Testing function for saveHashtag function in Hashtag to save Hashtag Entity in the datastore** | **Passed** |
| **1.5.2** | **getAllHashtags()** | **Testing function for getAllHashtags function in Hashtag to get all hashtags existing in the datastore** | **Passed** |
| **1.5.3** | **getHashtagPosts(String hashtagName, String uname)** | **Testing function for getHashtagPosts function in Hashtag to get all posts belonging to hashtagName that user with username “uname” has access to see.** | **Passed** |

* 1. **HashtagComparatorTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.6.1** | **compare(Hashtag x, Hashtag y)** | **Testing function for compare function in HashtagComparator that implements java class Comparator to customize the comparison between two hashtags and make it according to each hashtag’s posts count** | **Passed** |

* 1. **LikeTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.7.1** | **attach(NotificationObserver observer)** | **Testing function for attach function in Like that adds a new notification observer to the list of like observers in the Like class** | **Passed** |
| **1.7.2** | **notifyAllObservers()** | **Testing Function for notifyAllObservers in Like to iterate over the list of Notification observers in the Like class and update** | **Passed** |

* 1. **FriendRequestTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.8.1** | **attach(NotificationObserver observer)** | **Testing function for attach function in FriendRequest that adds a new notification observer to the list of friendRequest observers in the FriendRequest class** | **Passed** |
| **1.8.2** | **notifyAllObservers()** | **Testing Function for notifyAllObservers in FriendRequest to iterate over the list of Notification observers in the FriendRequest class and update** | **Passed** |
| **1.8.3** | **sendFriendRequest()** | **Testing function for sendFriendRequest in FriendRequest to save a friend request entity in the datastore after checking that receiver is a valid user and both users are not already friends. After saving, it notifies all observers to send notifications.** | **Passed** |
| **1.8.4** | **acceptFriendRequest()** | **Testing function for acceptFriendRequest in FriendRequest to getting a friend request entity from the datastore, changing its “pending” attribute to zero and putting it back. And then, saves a new Friend entity in the datastore meaning that both users are now friends and then it notifies all observers.** | **Passed** |
| **1.8.5** | **getFriendRequest(String ID)** | **Testing Function for getFriendRequest in FriendRequest to get a friend request entity from that datastore by ID** | **Passed** |
| **1.8.6** | **isFriends(String user1 , String user2)** | **Testing Function for isFriends in FriendRequest to check if those 2 users are friends by iterating over the Friends table from the datastore** | **Passed** |

* 1. **PersonalMessageTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.9.1** | **sendMessage()** | **Testing function for sendMessage function in PersonalMessage that calls function saveMessage and then notifyObservers to send a notification.** | **Passed** |
| **1.9.2** | **saveMessage()** | **Testing Function for saveMessage in PersonalMessage to save new message entity in table personalMsg in the datastore.** | **Passed** |
| **1.9.3** | **getNewID()** | **Testing function for getNewID in PersonalMessage to load all the “personalMsg” table from the datastore in a list and return list’s size + 1** | **Passed** |
| **1.9.4** | **getPersonalMsg(String id)** | **Testing Function for getPersonalMsg in PersonalMessage to get a personal message entity from that datastore by ID** | **Passed** |

* 1. **PublicPrivacyTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.10.1** | **canSeeUserPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeUserPost function in PublicPrivacy to see if current user can see user post**  **Assumption entity: is public** | **Passed** |
| **1.10.2** | **canSeeFriendPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeFriendPost function in PublicPrivacy to see if current user can see Friend post**  **Assumption entity: is public** | **Failed** |
| **1.10.3** | **canSeePagePost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeePagePost function in PublicPrivacy to see if current user can see Page post**  **Assumption entity: is public** |  |
| **1.10.4** | **canSeeSharedPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeSharedPost function in PublicPrivacy to see if current user can see Shared post**  **Assumption entity: is public** |  |

* 1. **PrivatePrivacyTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.11.1** | **canSeeUserPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeUserPost function in PrivatePrivacy to see if current user can see user post**  **Assumption entity: is Private** | **Passed** |
| **1.11.2** | **canSeeFriendPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeFriendPost function in PrivatePrivacy to see if current user can see Friend post**  **Assumption entity: is Private** | **Failed** |
| **1.11.3** | **canSeePagePost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeePagePost function in PrivatePrivacy to see if current user can see Page post**  **Assumption entity: is Private** |  |
| **1.11.4** | **canSeeSharedPost(Post result,Entity entity, String onWall,String currentUser)** | **Testing function for canSeeSharedPost function in PrivatePrivacy to see if current user can see Shared post**  **Assumption entity: is Private** |  |
| **1.11.5** | **handlingSharedPost(boolean result ,String owner,String onWall, String currentUser , String type)** | **Testing function for handlingSharedPost function in PrivatePrivacy to see if current user can see original post** |  |

* 1. **PrivacyTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.12.1** | **OriginalSharedPostPrivacy(boolean result ,String ID , String currentUser)** | **Testing function for OriginalSharedPostPrivacy function in Private to see if current user can see original post** | **Passed** |

**1.13 PostTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.13.1** | **getAllPostsForUser(String currentUser)** | **Testing function for getAllPostForUser function to see if current user can get all his posts** | **Passed** |
| **1.13.2** | **getPostsForTimeLine(String onWall, String currentUser)** | **Testing function for getPostsForTimeLine function to see if current user can see all posts on his timeline** | **Passed** |
| **1.13.3** | **getPostbyID(String postID)** | **Testing function for getPostByID function to see if post got is the same with the ID given** |  |

**1.14 UserPostTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.14.1** | **getPost(Entity entity)** | **Testing function for getPost**  **function to get the post saved in the datastore by the entity** | **Passed** |
| **1.14.2** | **savePost()** | **Testing function for savePost**  **function to save User Post in the datastore** | **passed** |

**1.15 FriendPostTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.15.1** | **getPost(Entity entity)** | **Testing function for getPost**  **function to get the post saved in the datastore by the entity** | **Passed** |
| **1.15.2** | **savePost()** | **Testing function for savePost**  **function to save Friend Post in the datastore** | **passed** |

**1.16 PagePostTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.16.1** | **getPost(Entity entity)** | **Testing function for getPost**  **function to get the post saved in the datastore by the entity** | **Passed** |
| **1.16.2** | **savePost()** | **Testing function for savePost**  **function to save Page Post in the datastore** | **passed** |

**1.17 SharePostTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.17.1** | **getPost(Entity entity)** | **Testing function for getPost**  **function to get the post saved in the datastore by the entity** | **Passed** |
| **1.17.2** | **savePost()** | **Testing function for savePost**  **function to save shared Post in the datastore** | **passed** |

**1.18 UserEntityTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.18.1** | **getUser(String name, String pass)** | **Testing function for getUser**  **function to form User Entity class contains user data from datastore** | **Passed** |
| **1.18.2** | **isUser(String name)** | **Testing function for isUser**  **function to check if the user is in the datastore** | **passed** |
| **1.18.3** | **saveUser()** | **Testing function for saveUser**  **function to save user object in datastore** | **Passed** |
| **1.18.4** | **getAllUsers()** | **Testing function for saveUser**  **function to get all the users in datastore** | **Passed** |

**1.19 MessageReceiverTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.19.1** | **readPersonalMessage(String ID)** | **Testing function for readPersonalMessage function to get the personal message by ID** | **Passed** |
| **1.19.2** | **readGroupMessage(String ID)** | **Testing function for readGroupMessage function to get group message by ID** | **Passed** |

**1.20 ReadPersonalMessageCommandTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.20.1** | **execute(String ID)** | **Testing function for execute function to return personal message by ID** | **Passed** |

**1.21 ReadGroupMessageCommandTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.21.1** | **execute(String ID)** | **Testing function for execute function to return group message by ID** | **Passed** |

**1.22 PageTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.22.1** | **checkPageEistence(String pageName)** | **Testing function to check if a page exists in the datastore or not given the page Name** | **Passed** |
| **1.22.2** | **isOwner(boolean result, String owner, String pageNae)** | **Testing function to check if ta certain user is the owner of a specific page or not**. | **Passed** |
| **1.22.3** | **getPageOwner (String owner, String IDss)** | **Testing function to get the owner of a specific page**. | **Passed** |

**1.23 LikePageTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.23.1** | **isLikePage(boolean reslt, String ID, String UserName)** | **Testing function to check if a page is liked by a user or not** | **Passed** |

**1.24 SeenPagePostTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.24.1** | **getNumberOfSeen( int counter, String ID)** | **Testing function to get number of seens for a post on a page** | **Passed** |

# 2. Frontend testing

* 1. **HashtagServicesTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **2.1.1** | **getTrendsService()** | **Testing function for getTrendsService function in**  **HashtagServices that returns the top 10 hashtags in a JSON array**  **Assumption: Total number of hashtags is less than 10** | **Passed** |
| **2.1.2** | **getHashtagPostsService(@FormParam("hashtagName") String hashtagName, @FormParam("uname") String uname)** | **Testing function for getHashtagPostsService function in HashtagServices that returns a JSON array containing all posts belonging to a specific hashtag** | **Passed** |

* 1. **NotificationServicesTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **2.2.1** | **addFriendService()** | **Testing function for addFriendService function in**  **NotificationServices that returns a JSON object containing the success status of this operation Assumption:**  **user is sending a friend request to himself**  **Expected Result:**  **“Status”:”yourself”** | **Passed** |
| **2.2.2** | **getAllNotificationsService(@FormParam("uname") String uname)** | **Testing function for getAllNotificationsService function in NotificationServices that returns a JSON array containing all notifications belonging to a specific user**  **Assumption:**  **User has got a friend request that he has already accepted**  **Expected Result: That request won’t appear in the notifications page** | **Passed** |

* 1. **NotificationInvokerTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **2.3.1** | **handleNotificationService(String result , String ID , String type)** | **Testing function for handleNotificationService function in**  **NotificationInvoker that returns the action of the command in a JSON array** | **Passed** |

* 1. **PostsServicesTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **2.4.1** | **CreateUserPostService(String owner, String content, String onWall, String privacy, String feeling, String customUsers)** | **Testing function for CreateUserPostService function in**  **PostsServices that returns UserPost in a JSON object** | **Passed** |
| **2.4.2** | **CreateFriendPostService(String owner, String content, String onWall, String privacy, String customUsers)** | **Testing function for CreateFriendPostService function in**  **PostsServices that returns FriendPost in a JSON object**  **Assumption: Current user is not his/her friend**  **Expected Result: status”You are not friends”** | **Passed** |
| **2.4.3** | **CreatePagePostService(String owner, String content, String onWall, String privacy, String customUsers)** | **Testing function for CreatePagePostService function in**  **PostsServices that returns PagePost in a JSON object**  **Assumption: Current user is not the owner of the page**  **Expected Result: status”You are not the owner of this page”** | **Passed** |
| **2.4.4** | **CreateSharePostService(String originalPostID, String owner, String content, String onWall, String privacy, String customUsers)** | **Testing function for CreateSharePostService function in**  **PostsServices that returns SharePost in a JSON object**  **Assumption: The post is shared on a non-friend’s timeline**  **Expected Result: status”you are not friends”** | **Passed** |
| **2.4.5** | **GetPostsForTimeLineService(String onWall, String currentUser)** | **Testing function for GetPostsForTimeLineService function in**  **PostsServices that returns posts of a timeline in a JSON array**  **Assumption: Timeline is not found**  **Expected Result: “There is no such TimeLine”** | **Passed** |
| **2.4.6** | **GetNewsFeed(String currentUser)** | **Testing function for GetNewsFeed function in**  **PostsServices that returns all possible posts in a JSON array to the current user** | **Passed** |
| **2.4.7** | **LikePost(String currentUser, String ID)** | **Testing function for LikePost function in**  **PostsServices that returns post in a JSON object**  **Assumption: post is already liked by currentUser**  **Expected Result: “you've already liked this post”** | **Passed** |

# 

* 1. **PageServiceTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **2.5.1** | CreatePageService**Test()** | **Testing function for creating a new page that returns a JSON object containing the success status of this operation Assumption:**  **user is creating a page with a non existign name**  **Result:**  **“Status”:”Page created successfully”** | **Passed** |
| **2.5.2** | LikePage**Test()** | **Testing function for Liking a page that returns a JSON object containing the success status of this operation Assumption:**  **user is liking a page w**  **Result:**  **“Status”:”Page liked successfully”** | **Passed** |

2.6 **UserServicesTest class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **2.6.1** | registrationService **(Username, Password, Email)** | **Testing function for Create new user that returns UserPost in a JSON object**  Ex: "Registration success" if a new user is registering | **Passed** |
| **2.4.2** | **loginServiceTest(String username, String Password)** | **Testing function for loging in a user to the home page**  **void**  **Assumption: Current user approved**  **result-> directed to home page** | **Passed** |

# Github

[**https://github.com/heba-khazbak/FCI-CS352-CS\_2**](https://github.com/heba-khazbak/FCI-CS352-CS_2)

# 

# Ownership report

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Testing Functions numbers** |
| 20120516 | Heba Khazbak | 1.1.1, 1.1.2, 1.1.3, 1.1.4, 1.1.5, 1.2.1, 1.3.1,1.3.2 , 1.4.1 ,1.4.2, 1.10.1 ,1.10.2,1.10.3 ,1.10.4, 1.11.1 ,1.11.2,1.11.3 ,1.11.4, 1.11.5, 1.12.1 , 2.3.1 |
| 20120191 | Sarah Hany | 1.22.1 , 1.22.2 , 1.22.3 1.23.1 , 1.23.2 , 1.24.1 2.5.1 , 2.5.2 , 2.6.1 , 2.6.2 |
| 20120166 | Dalia Maher | 1.13.1, 1.13.2, 1.13.3, 1.14.1, 1.14.2, 1.16.1, 1.16.2, 1.17.1, 1.17.2, 1.18.1, 1.18.2, 1.18.3, 1.18.4, 1.19.1, 1.19.2, 1.20.1, 1.21.1, 2.4.1, 2.4.2, 2.4.3, 2.4.4, 2.4.5, 2.4.6, 2.4.7 |
| 20120185 | Ziad Mohamed | 1.5.1, 1.5.2, 1.5.3, 1.6.1, 1.7.1, 1.7.2, 1.8.1, 1.8.2, 1.8.3, 1.8.4, 1.8.5, 1.8.6, 1.9.1, 1.9.2, 1.9.3, 1.9.4, 2.1.1, 2.1.2, 2.2.1, 2.2.2 |