|  |
| --- |
| **CLASS**  **User** |
| **RESPONSBILITY**   1. Maintain the user name. 2. Maintain the user id. 3. Maintain the user gender. 4. Maintain the user email. 5. Maintain the user phone number. 6. Maintain the user biography. 7. Maintain the user birthdate. 8. Maintain the user profile picture. 9. Maintain the user cover picture. 10. Associate a Google account. 11. Update the user profile picture. 12. Update the user cover picture. 13. Update the user biography. 14. Get the friends list. 15. Get the number of friends. 16. Refresh all the data maintained with the new one in the server. 17. Send a message to **Friendship** to send or cancel a friend request to another user. 18. Send a message to **Friendship** to respond to a friend request from another user. 19. Send a message to **Friendship** to terminate friendship with a friend 20. Send a message to **Post** to write a new post to specific target. 21. Send a message to **Post** to interact with post. 22. Send a message to **Post** to Save a post. 23. Send a message to **Post** to get posts of group/page/profile/timeline. 24. Send a message to **Post** to get the number of posts. 25. Send a message to **Group** to create a group. 26. Send a message to **Group** to join/cancel join request in a group. 27. Send a message to **Group** to leave a group. 28. Send a message to **Page** to create a page. 29. Send a message to **Page** to follow/unfollow a page. 30. Send a message to **Chat** to send a chat message to user/group. 31. Send a message to **Marketplace** to add/remove a product. 32. Send a message to **Marketplace** to request a product. 33. Send a message to **Notification** to fetch all notifications. |
| **COLLABORATION**   1. Class **Friendship**. 2. Class **Group**. 3. Class **Page**. 4. Class **Post**. 5. Class **Chat** 6. Class **Notification** |

# 7.b. CRC cards:

|  |
| --- |
| **CLASS**  **Friendship** |
| **RESPONSBILITY**   1. Show the relationship between two **Users**. 2. Send or cancel a friend request from **User** to another **User**. 3. Response to (accept or cancel) the friend Requests of the **User**. 4. Terminate/remove the friendship between two **Users**. 5. Update the data of both sides of relationship. 6. Send a message to **Notification** on receiving new requests to notify **User** of having new friend requests. 7. Send a message to **Notification** to notify **User** about any news regarding accepting or cancelling his friend requests. |
| **COLLABORATION**   1. Class **User**. 2. Class **Notification**. |

|  |
| --- |
| **CLASS**  **Post** |
| **RESPONSBILITY**   1. Submit the post that the **User** wrote. 2. Display friends' posts on the homepage. 3. Display the **User** posts on his profile page. 4. Display Group Posts. 5. Display Page Posts. 6. Calculate and Send the number of posts to **User**. 7. Interact with (like, comment or share) a post. 8. Save a Post when a **User** request it. |
| **COLLABORATION**   1. Class **User**. |

|  |
| --- |
| **CLASS**  **Verification** |
| **RESPONSBILITY**   1. Verify the entry of data. 2. Check if there are any applied restrictions on each input type. |
| **COLLABORATION**  **---** |

|  |
| --- |
| **CLASS**  **Chat** |
| **RESPONSBILITY**   1. Send a Chat Message to **User**/group of users. 2. Get Chat Messages from **User**/group of users. |
| **COLLABORATION**   1. Class **User**. |

|  |
| --- |
| **CLASS**  **Notification** |
| **RESPONSBILITY**   1. Show Notifications of a **User** on requesting. 2. Store a notification to a **User/Group Owner** contains a body of notification and the sender name **(User/Group/Page)**. 3. Store a notification to a User to inform him about **Friendship** statues with another **User** 4. Store a notification to a User to inform him about **Marketplace** order. 5. Send a message to **User** to notify him/her about new notifications to fetch. |
| **COLLABORATION**   1. Class **User**. 2. Class **Friendship**. 3. Class **Group.** 4. Class **Group** **Owner.** 5. Class **Page.** 6. Class **Marketplace.** |

|  |
| --- |
| **CLASS**  **Group** |
| **RESPONSBILITY**   1. Receive and add the new join requests from **User** to the pending requests. 2. Remove the join request from the pending requests on receiving cancel   request from the **User**.   1. Remove a member from the group on receiving leave request from the member. 2. Create a new group on receiving creation request and make the User who requested group creation a **Group** **Owner**. 3. Send a message to **Notification** to notify Group Owner about new join requests. 4. Send a message to **Notification** to notify User about the action taken on his join requests. 5. Handle the pending join requests (accept or refuse) on receiving the action from the **Group** **Owner**. 6. Kick a member from the group. 7. Delete a group on receiving delete request from the group owner. |
| **COLLABORATION**   1. Class **Notification**. 2. Class **User**. 3. Class **Group** **Owner**. |

|  |
| --- |
| **CLASS**  **Group** **Owner** |
| **RESPONSBILITY**   1. Send a message to **Group** to handle a pending request with a specified action. 2. Send a message to **Group** to kick a member from the group. 3. Send a message to **Group** to disband the group if desired. |
| **COLLABORATION**   1. Class **Notification**. 2. Class **Group**. |

|  |
| --- |
| **CLASS**  **Page** |
| **RESPONSBILITY**   1. Handle follow/unfollow requests from **Users**. 2. Give administration to a follower of the page on receiving a give admin request from **Page** **Owner**. 3. Send a message to **Notification** to notify a **User** that he/she is now a **Page** **Owner**. 4. Ban a follower from the page on receiving ban request. 5. Create a page on receiving creation request from a **User**. 6. Delete a page on receiving delete request from the **Page** **Owner**. |
| **COLLABORATION**   1. Class **User**. 2. Class **Page** **Owner**. 3. Class **Notification**. |

|  |
| --- |
| **CLASS**  **Page** **Owner** |
| **RESPONSBILITY**   1. Send a message to **Page** to give administration to a specific user. 2. Send a message to **Page** to ban a follower from the page. 3. Send a message to **Page** to disband the Page if desired. |
| **COLLABORATION**   1. Class **Page**. |

|  |
| --- |
| **CLASS**  **Marketplace** |
| **RESPONSBILITY**   1. Add a new product to the **User** marketplace. 2. Remove an exist product from the seller marketplace. 3. Show all products in the marketplace of the seller. 4. Send a message to **Notification** to notify seller that a **User** wants to order a product. |
| **COLLABORATION**   1. Class **User**. 2. Class **Notification**. |

|  |
| --- |
| **CLASS**  **Analyst** |
| **RESPONSBILITY**   1. Collect statistics and data from the server (number of users, posts, pages, groups, and number of online users) |
| **COLLABORATION**  **----** |

|  |
| --- |
| **CLASS**  **Tracker** |
| **RESPONSBILITY**   1. Ban a User, Group, or Page from the website. 2. Unban a User, Group, or Page from the website. |
| **COLLABORATION**  **----** |

|  |
| --- |
| **CLASS**  **Admin** |
| **RESPONSBILITY**   1. All Analyst Duties. 2. All Tracker Duties. |
| **COLLABORATION**   1. Class **User**. 2. Class **Group**. 3. Class **Page** |