

# Analytics Phase

## Team Details:

Kanad Pardeshi	: 190050056
Manan Agarwal	: 190050065
Tanu Goyal	: 190050123
Dhruva Dhingra	: 190070020

## Part A: Problem Description

As the institute has been in the online mode of operation for nearly two years, and we are finally opening up again, it might be difficult for all of us to get out of our shells and find new friends. We plan to develop a social network website for the institute, by the institute - InstiGram - using the concepts learnt in the course. The requirements are as follows:

1. A secure login-based authentication for the users.
2. Users can post on a page visible to their friends (or everyone), and also chat with their friends personally, or in groups.
3. Users can also analyze their usage of the website.
4. The administrators of groups can analyze group statistics.
5. Pages can also be made for organizations, with the maintainers of the page having the ability to post, while other users can follow these pages.
6. Users from across the institute can join the platform and find new friends from recommendations based on their branch, hobbies, and other common interests.
7. The administrators of the website would be able to analyze statistics about the usage of the website by all users.

A social network requires efficient storage, retrieval and analysis of a lot of data, and databases are the ideal choice to deal with this data. We plan to use a mixture of relational databases (for storing conventional data, since querying would generally be more geared towards the relational side), and graph databases (for recommendation of friends and pages, as that is best viewed as a graph).

## Part B: User Classes

Any person who uses the InstiGram can be one of the following:

### 1) User:

Person who uses the Instagram for the purpose of interaction over the Social Media platform is labelled, as User. Any User has access to the following features:

1. Create an Account
2. Delete their Account
3. Log-in/ Log-out
4. Modify Account Description - Profile Picture, Personal Details and Hobbies
5. Upload Status
6. Post on the Timeline
7. Share a Post
8. Delete a Post
9. Comment and React to a Post
10. Search for another User
11. Friend Requests - Send Friend Requests, and/or Accept the Requests
12. Visit timeline of Friends
13. Optional : Block a User
14. Send / Receive message from Friends
15. Create a Page
16. Follow a Page
17. Visit a Page
18. Create a Group Chat
19. Accept / Reject Invites to a Group Chat
20. View Analytics : Friendship and Page Recommendations

Any User can join a group, of which he or she can be a participant or group admin, giving them access to the following set of features:

**User - Participant :**

1. Send and Receive Group Messages
2. Delete a Message
3. Share the Group Invite with Friends
4. Exit the Group

Along with the features provided to the Participant, a Group Admin has access to some more features, as follows:

**User - Group Admin :**

1. Accept or Remove a User
2. Change Group Description or Profile Image
3. Delete Group
4. Make another Participant as Admin
5. Access to Group Analytics : Most active group members

Any User can also follow a page, of which he or she can be a follower or page admin, giving them access to the following set of features:

**User - Follower :**

1. Visit Private Pages, of which they are followers
2. Comment and Like on the Posts

If the user happens to be the Page admin, then he or she gets to:

**User - Page Admin :**

1. Create or Modify a Post
2. Delete any Post
3. Delete Page
4. Make another Follower as Admin
5. Access to Page Analytics : Average Followers Activity on Page

**2) Website Admin:**

Person who administers the Instagram website can log in, to check various analytics, and carry out changes to the website. Any website Admin has access to the following features

1. Log-in/ Log-out
2. Visit any User's Timeline
3. View any Post
4. Remove a User from the social network
5. Delete any Post/ Comment / Page or Group
6. View Website Analytics such as User Count, and Network Traffic

**Part C: Entities**

**User:**

1. User ID :
2. First Name
3. Last Name
4. Email :
5. (Hash of) Password :
6. Hobbies :
7. Branch :
8. Degree

9. Batch:
10. Roll No :
11. Residence :
12. Date of Joining InstiGram :
13. Profile Picture:
14. Birthday :
15. Private/Public :
16. Can User be auto-added to groups :

**Post:**

1. Post ID :
2. Content :
3. Content\_type:
4. Time Posted :
5. Tags :
6. Visibility:

**Status:**

1. Status ID :
2. Content :
3. Content\_type :
4. Time Posted :

**Page:**

1. Page ID :
2. Page Name :
3. Profile Picture :
4. Page Description :
5. Page Created On :
6. Page Created By :
7. Page Admin :
8. About :

**Group:**

1. Group ID :
2. Group Name :
3. Group Description :
4. Group Created On :
5. Group Created By :
6. Profile Picture :
7. Group Admin :

**Message:**

1. Message ID :
2. Message Content :
3. Time sent :
4. Deleted :

**Website Admin:**

1. Admin ID :
2. Email :
3. Password :

**Hobby:**

1. Hobby ID
2. Name
3. Category
4. Description

**Main Relations :****1. FRIENDSHIP****Degree : 2**

Entity 1 : User, Role : Friend\_1 :

Entity 2 : User, Role : Friend\_2 :

Time of sending request :

Time\_of\_Friendship = Time of accepting request :

Status : Pending / Accepted

**2. FOLLOWER****Degree : 2**

Entity 1 : User, Role : Follower:

Entity 2 : Page, Role : Followed :

Time\_of\_Following :

**3. TAGS****Degree : 2**

Entity 1 : Post, Role : Mentions:

Entity 2 : User, Role : Mentioned :

**4. POSTED\_BY\_USER****Degree : 2**

Entity 1 : Post, Role : Post: Total Participation

Entity 2 : User, Role : Owner : ->

## 5. POSTED\_BY\_PAGE

**Degree : 2**

Entity 1 : Post, Role : Post: Total Participation

Entity 2 : Page, Role : Owner : ->

## 6. REACTION\_BY

**Degree : 2**

Entity 1 : Post, Role : Post

Entity 2 : User, Role : User

Reaction

## 7. COMMENTED\_BY

**Degree : 2**

Entity 1 : Post, Role : Post : Total participation

Entity 2 : User, Role : User : ->

Comment\_id

Comment content

Time posted

Time last edited

Deleted

## 8. MEMBER\_OF

**Degree : 2**

Entity 1 : Group, Role : Group :

Entity 2 : User, Role : User:

Privilege : (Owner / Admin / Can post / can comment / can like / can only view)

Time of sending request :

Time\_of\_Friendship = Time of accepting request : (= time of sending request if user can be auto added to groups)

Status : Pending / Accepted (Accepted if user can be auto added to groups)

## 9. PRIVATE\_MESSAGE

**Degree : 3**

Entity 1 : User, (receiver)

Entity 2: User, (sender)

Entity 3 : Message

View Once Only : (To be done later)

## 10. GROUP\_MESSAGE

**Degree : 3**

Entity 1 : Group, (receiver):

Entity 2 : User (sender)

Entity 3: Message

## 11. UPLOADED BY

**Degree : 2**

Entity 1 : User, Role : User\_posted :

Entity 2 : Status, Role : Status\_Posted :

## Part D: Forms

We now enumerate all the uses cases, accessible to the different classes of users, as follows

### 1. Signing Up

- a. **Triggered automatically** on opening the webpage
  - i. If the user is signed in, then, the user is directed to the homepage.
  - ii. If not, the user is directed to the login/signup form.
- b. **Primary User** : Anybody who uses the site or wants to use the site is directed to this page.
- c. **Inputs** : User\_id, First\_Name, Last\_Name, Email, Password, Branch, Batch, Roll Number.
  - i. None of the inputs may be empty
  - ii. The user\_id must be unique from other users
- d. **Outputs** : Confirmation that the account has been created by redirection to friend recommendations page.
- e. **Exceptions** : If a user with that user\_id exists, an error message should be given that a user with that id already exists.
- f. **Pre-Conditions** : None relevant
- g. **Post-Conditions** : None relevant

### 2. Login Form

- a. **Triggered automatically**
- b. **Primary User** : Anybody who uses the site or wants to use the site is directed to this page.
- c. Option along with signup form on landing page
- d. **Inputs**: User\_id, password
- e. **Outputs** : Successful login via redirection to user's homepage or else an error message
- f. **Exceptions** : "Account does not exist" if invalid user\_id else "wrong\_password" message
- g. **Pre-Conditions** : None relevant
- h. **Post-Conditions** : None relevant

- i. JSON Web Token Authentication

### 3. User Home Page

- a. **Triggered automatically** on logging in
- b. **Primary User** : Anybody who uses the site or wants to use the site is directed to this page.
- c. **Pre-Conditions** : None relevant
- d. **Post-Conditions** : None relevant
- e. View posts from user's friends / pages followed
- f. Default sort by time (latest first)
- g. Other sorts : We will look into this a little later.
- h. Top say 20 or so posts will **probably** be maintained via a materialized view.
- i. **Input** by user : None
- j. **Input** for backend: User ID
- k. **Output** from backend: Posts with times, sorted in descending order

### 4. Friend Recommendations Page

- a. **Triggered manually**
- b. **Primary User** : All users
- c. **Pre-Conditions** : None relevant
- d. **Post-Conditions** : None relevant
- e. **Input** by user : None
- f. **Input** for backend: User ID
- g. **Output** from backend: Recommended Users

### 5. Search Form

- a. **Triggered manually**
- b. **Input** : Keyword Searching for other users
- c. **Output** : Search results if that keyword is:
  - A part of some User\_id
  - A part of some First name
  - A Roll number
  - A part of some Branch
- d. **Pre-Conditions** : None relevant
- e. **Post-Conditions** : None relevant

### 6. Send Friend Request Form

- a. **Triggered manually**
- b. Option to send friend request to user
- c. A confirmation would be needed from the user to whom the request has been sent



- d. **Input** : Click on send request to the user
- e. **Output** : Confirmation that the request has been sent

## 7. View and Accept/Reject Friend Requests Form

- a. **Input** by user to land on this page: None
- b. View a list of friend requests
- c. **Input** by user on each friend request: Options are accept/reject below each friend request.
- d. We plan to implement this interface similar to Facebook's Friend Request Page.
- e. **Output** : Confirmation that the request has been accepted / rejected (The accept / reject buttons will be replaced by request accepted / rejected)

## 8. Send message / view received messages form (Can be to user/group)

- a. **Input** by user to land on this page: None
- b. View a list of past chats to friends / groups
- c. Select a friend/group from this list or create a new chat (via a create new chat option and selecting a friend or group which the user is a part of).
- d. Send messages and view messages with this friend / group.
- e. **Input** : A text box to enter content
- f. **Output** : Message visible in the chat along with sending/sent/viewed

## 9. View/edit user's own page

- a. **Triggered manually** on that option being chosen
- b. Edit user credentials (name, branch, roll number, hobbies, etc)
- c. Edit profile picture
- d. **Input** to backend: User ID
- e. **Output** from backend: User information

## 10. View other user's pages

- a. **Triggered** on choosing to go to the page of the user
- b. View user credentials, and their profile picture
- c. **Input** to backend: User ID of user being viewed
- d. **Output** from backend: User information corresponding to user ID

## 11. View Pages

- a. View the page of a group, consisting of group information like members in the group, time when group was created, etc.
- b. **Input** to backend: Group ID33
- c. **Output** from backend: Group information

## **12. Create Group Form**

- a. Create a new group and add members (along with their privileges, multiple admins possible)
- b. Constraint: only admins can add known friends to the groups
- c. **Input** to backend: List of users and their privileges
- d. **Output** from backend: Notification of creation of group

## **13. Edit Group Form**

- a. Add users, remove users, change privileges of users
- b. This form is only visible to administrators of that group

## **14. Leave Group Form**

- a. Leave a group
- b. Available to all users
- c. Ask for confirmation regarding leaving of group
- d. If the leaving person is the sole admin, ask them to choose new admin(s)

## **15. Create Page Form**

- a. Create a page along with the description and people allowed to maintain (=post content to) it.

## **16. Post Content (for self)**

- a. Create a new post
- b. Options to add text or upload images (optionally videos)

## **17. Post content to Page (only for maintainers of Page)**

- a. Create a post for a page
- b. Options to add text or upload images (optionally videos)

## **18. User Analytics Page**

- a. Option for user to look at their statistics

## **19. Subscribe to Page Form**

- a. Option to subscribe
- b. Posts from the page would appear in the user's feed, that is, their homepage.

## Part E: ER Diagram

