using 'LAB 2 - REQUIREMENTS';

1. Requirements regarding: [FOLLOWS]

As a user I want to be able to receive follow suggestions based on common groups, followers, organisations or geographical areas.

- The list of friend suggestions will be displayed in a page section called "Friend suggestions". Each displayed user will have a short description of the things that they have in common with us and [to be implemented] a button "Add" that allows us to send a follow request to that user.
- A suggested user's short description consists of the common groups, common followers, common organisations or the same geographical zone as us.
- The list of friend suggestions will be displayed taking into account the following categories:
  - [to be implemented] Common groups: users that are part of the same groups as our user. The users which are part of a larger number of common groups have a higher priority when displaying the list of friend suggestions.
  - Common followers: users that follow the same people as our user. The users which have a higher number of common followers will have a higher priority when displaying the list of friend suggestions.
  - Common organisations: users that are in the same organisations as our user. The users which have a higher number of common organisations will have a higher priority when displaying the list of friend suggestions.
  - Geographical zone: users that are in the same geographical area as our user. The users that are in the same city/village have a higher priority than the others.
  - The order in which this categories are considered depends on the type of user account. For a professional account, the order is: number of common organisations, common skills, common groups, location. For a casual account, the order is: number of common friends, common hashtags, common groups, common organisations, location. The algorithm for sorting the profiles is the following. At first, the users that are followed by the accounts that the user follows are considered (closeness degree 2) and also those followed by them (closeness degree 3). Those are sorted by the first criteria, then at each step the first quarter of the users sorted in the last step is sorted by the next criteria and concatenated with the rest of the results that remain in the old sorting order. If the user follows nobody, then the suggestions are randomly chosen from people with the same hometown.

- A friendship is defined as two persons following each other. Depending on the privacy settings of the profile, follow requests can be accepted by default or not.

As a user, I want to be able to manage the degree of closeness of a friendship and the relationship with other users;
- There will be the a button to 'follow' and to 'send friend request' on each personal page of a user; the friend requests will be pending until accepted by the receiving party; if not accepted or deleted ('declined') after a predetermined amount of time (e.g. 6 months), the request is automatically deleted;
- Once a friend request is sent, the user has to wait either for the response of the other user, or until the predetermined period of time after which the request expires automatically is up, in order to be able to send another request;
- This feature implies the separation between friends and close friends; in the friend list, each friend will have a button that would allow changing the status from 'friend' to 'close friend'. The same button would be used to change the status between 'close friend' to 'friend'. (Implementation Idea: Boolean field);

As a user, I want to be able to restrict who is able to befriend and follow me, as well as interact with my posts;
- This feature implements the beloved 'blocking' a friend feature, which would restrict the access of the blocked person to the blocker's posts and inbox; the implementation would imply a 'block' button, listed next to their name in the friend list.
- There will be a list of blocked users, with the possibility to unblock one or multiple people at once (checkbox list);
- [to be implemented] A user should be able to block multiple people at once, using the option to edit the friend list, and checking the corresponding boxes to the people the user wants to block.

As a user, I want to only interact with the people I am interested in, without having to unfriend people myself [to be implemented];
- [Timer for Friends] This feature would improve the user experience by pruning the inactive parts of the user's social network by setting a default 2 YEAR renewal policy for friendships;
- The user will be able to choose the 'duration' of their friendships: 6 months, 1 year, 2 years, 4 years.

Bonus Features:
As a user, I want to be able to easily manage my friend list;
- [CRUD operations] The user should be able to remove one or multiple users at once from their friend list, or edit their status;
- The user would be able to unfriend multiple people at once, or change their status between 'friend', 'close friend' and vice-versa;

- **[Categorising friends]** For improving the user experience, the user could categorize their friends (similar to a tagging system) based on suggested tags (family, friends, workplace) or self-defined tags ("besties", "nerds", "Aragorn fanboys");

2. Requirements regarding: [Default/Homepage Feed]

COMMON TO ALL FEEDS: The posts which pass through the filter with non-zero points will be cached respectively to their corresponding cache module. If the quota of posts which should have passed the filter when filtering the posts for each feed isn't met, a new fetch call is performed for the past 6 to 3 days, and the actions are repeated.

All of the posts which pass through the filtering process will be sorted ascending on their attributes. In addition, for the custom, homepage and following feeds, the posts will also be sorted on their date of upload on time buckets (posts uploaded from current time to 3h ago will be sorted from latest to oldest in their respective time bucket, posts from 3h ago to 6h ago and so forth.)

DEFAULT FEED/HOMEPAGE FEED

The user should see posts from people that they follow on their default homepage after clicking on the button "homepage" inside the feed carousel.
- We will implement an algorithm for the order in which the posts are displayed, based on time buckets of 3 days and relevance given by the the authors of the posts, if they are followed, friends or close friends of the user.

     1. The posts of all users who the owning user is following (taking into account the degree of closeness) will de displayed, as each close friend will be given 5 points, friend 2 points, followed 1 point, while the ones of unrelated users 0 points;

     2. OPTIONAL: If the location of the post matches the location of the user, then it is given 1 point;
- These posts will be split in multiple "time-buckets", and posts within a time bucket will be displayed in the order of relevance.
- Ads will be integrated on the user's default feed based on their relevancy, and implement a mechanism so that they will not overwhelm the feed. (if the group managing adds handles the add frequency)

Bonus Features:

As a user, I can select "not interested","don't see this kind of content","don't see this post" from a menu associated with each post - this part will be handled by completely excluding the posts which have been given that status, and lowering the frequency and position lower on the feed other similar posts (common hashtags, posts from the user who had uploaded the ignored post)

2.5 Requirements regarding: [Following Feed & Trending Feed & Controversial Feed]

FOLLOWING FEED

The user should see posts from people that they follow on their following page after clicking on the button "following" inside the feed carousel.

Steps for the algorithm for the Following configuration:

-Only posts from users that the current application context user is following will be displayed.

-The posts will be displayed in reverse chronological order, as the latest posts have the highest priority on the feed; posts which are from authors the user follows will be given 1 point, while posts of unrelated authors 0 points;

-If there are no entries stored in the "following" attribute in the following feed class, then no posts are returned.

-Divide posts into time-buckets based on when they were posted (1h/12h/24h/3days etc).

TRENDING FEED

Steps for the algorithm for the Trending feed configuration:

-Main service preparation: Considering this configuration is shared between all users, and the influx of posts will be the greatest and the viral state of a post is subject to rapid change, the cache module will have a shorter lifetime. Searching for new posts which are trending will imply being fetched from the posts service. The first batch of posts will contain only posts from the past 3 days. When this batch of posts are filtered, they will go through:

-Method:

Check whether or not the engagement metrics fulfil the criteria for a minimum like count, comment count and view count which are set in the attributes of the Trending class.

> 1. Filter by view count: high views -> 3 points, middle views 1 points, low to no views -> 0 points
> 2. Filter by like count: high likes ->3 points, middle likes: 1 point, no to 0 likes->0 points;
> 3. Filter by comment count: high nr of comments -> 3 points, medium nr of comments->1 points, low nr of comments->0 points.
> 4. If the negative reaction count is higher than the positive ones, the post is excluded from trending

-Further filtering: [BONUS] If there are posts fulfilling the engagement metrics which are from authors the user follows or is friends with, they will be displayed higher on the page.

CONTROVERSIAL FEED

Steps for the algorithm for the Controversial feed configuration:

-Main service preparation:

-Method: Check whether or not the engagement metrics fulfil the criteria for a minimum like count, comment count, a high ratio of positive reactions (like, love) to negative reactions (angry, sad) which are set in the attributes of the (reddit formula)
Controversial class.
- Sorted descending using the following formula: nb of reactions/max(abs(positive-negative), 1)

3. Requirements regarding: [Feed, Search & Filtering]

[to be implemented] As a user I want to be able to search for another user with a given username, first name or last name.

- A search bar and a button will be used to implement this functionality.
- After the user types in the search bar, they will press the search button and the list of users that have the username or first name & last name matching with the searched string.
- If there are no users matching the written username, first name or last name, a message will be shown in the space where the list of users was meant to be displayed. The message will be "No users found matching the description"

As a user I want to be able to configure custom feeds based on hashtags, geographical zones, users and [to be implemented] organisations.

- The user should have an option to let them switch to the configured feed where posts with given hashtags, geographical zones, users and organisations as input should be displayed. At least one or more of these fields must be given as input.
- The user should be able to switch between multiple feed configurations
- The posts for this type of feed will be shown according to these categories:
    - Users: [to be implemented - now works on id] posts from users given as input will be shown (if something was typed in this field). If nothing was given as input for this field, this category will be ignored and posts from any users which are not excluded through options such as block/ban/ignore are displayed on the feed.
    - Geographical zone: the location displayed on the post is in the same geographical area as one of the given locations (if something was typed in this field).
    - Hashtags: posts that have the matching hashtags in the description will be displayed (if something was typed in this field). If nothing was given as input for this field, this category will be ignored.
    - Algorithm for custom feeds:
        - The posts will be sorted in reverse chronological order.

- For each visible post in the current timeframe, a score is computed, taking into account the following criteria. Only the most relevant N posts are taken into account, where N is max(1000, nbFetchedPosts).
- 1. Filter by hashtags: full hashtag match: 5 points, partial hashtag match: 3 if nearly all of the hashtags are match, 1 if less than half the hashtags are matched, 0 if no hashtags match;
- 2. Filter by usernames: if the post owner is matched->3 points, otherwise -> 0 points
- 3. Filter by location: if the location of the post is matched with one of the set locations, 5 points are given;

Additional considerations:
- All features that take groups or organisations in consideration, in any way, will be bonus features, as the recommendation system can function independently without them, based on friends, hashtags and locations.