

---

# LinkUs

Project Proposal CS 387

Naman Singh Rana 200050083

Parekh Dhairya 200050097

Shikhar Mundra 200050131

Utkarsh Ranjan 200050147

---

## Overview of application:

Do you often get frustrated looking for a link shared by someone? Do you like reading and sharing content with your friends but cannot keep track of the exchange? Fret not, for we have something just for you!!

We intend to create an app with WhatsApp like user interface meant to share links. True to the vision, there will be provisions for making groups of users or sharing with individuals. Along with the links, you can add a small description and add relevant tags which describe the contents of the link shared. Just like WhatsApp, you will have provisions for logging in/ signing up, with username, password functionality and email ID integration (add google id integration\*). There will be a unique username (by you and for you 😊) and a profile visible only to you. You can also bookmark links which then will be shown in the profile helping you manage your interests (and a brief history of links of your interests\*).

You will be surprised how simple it can make things for you. You can see what books you have shared just by searching for the tag 'books' and you will be shown filtered data. You will also have the freedom to like and dislike the shared links and can sort the links according to their popularity.

Creating groups with your peers is just a click away, all you need is their username, and you are done, it does not get simpler. You will be the admin of the group you create (I know you like it 😊) and will have the authority to add/remove the members of a group (or make another user admin\*). If you are just a member, then you can leave the group at your will.

\* Indicates the features which we will integrate once the other features are up and running smoothly

# Users:

- **After signing up**, which will involve adding the personal information of the user to the *Users* table, a user can login/logout by using their password. They can also change their password. They will also be able to sort/filter links for particular tags in a specific group or in their entire chats.
- There will be two types of users in a group, an admin, and the members.
  - Any user will have the following features,
    1. **They can send a link, along with the tags attached and a short description.** This will make a simple call to backend to add the relevant details to the *Conversation* database
    2. **They can react (like/dislike) to a message.** This will update the count stored in the *React* table
    3. **They can bookmark a tag.** This will add entries to the *Bookmark* table at the backend
    4. **They can create a new group.** This action will create a new entry in the *Groups* table. They also will be (by default) the admin of the group
    5. **They can leave a group.** This will involve changes in the *Participants* table
  - The admin will have the additional privilege of adding/removing any member. This too involves changes in the *Participants* table.
- In a private chat, both users will have the same privileges. They both can,
  - **send a link, along with tags and description**
  - **react to a message**
  - **bookmark a tag**

## Data stored and (tentative) schema:

Users			
User ID	Username	Password	Email ID
Primary key			
Numerical non nullable	String non nullable	String non nullable	String non nullable

Groups	
Group ID	Group Name
Primary key	
Numerical non nullable	String non nullable

Participants		
Group ID	User ID	Role
Primary key, foreign key Groups	Primary key, foreign key Users	
Numerical non nullable	Numerical non nullable	Admin / Member

Links					
Link ID	Group ID	Sender ID	Link	Time	Desc
Primary key	Foreign key Groups	Foreign key Users			
Numerical non nullable value	Numerical non nullable	Numerical non nullable	String non nullable	Timestamp non nullable	String nullable

Tags	
Link ID	Tags
Primary key, foreign key Links	Primary key
Numerical non nullable	String non nullable

Reacts		
User ID	Link ID	React
Primary key, foreign key Users	Primary key, foreign key Links	
Numerical non nullable	Numerical non nullable	Like / Dislike

Bookmarks	
User ID	Link ID
Primary key	Primary Key
Numerical non nullable	Numerical non nullable

## Application Structure:

The API functionalities in backend required would be\*

- Sign Up
- Login
- Change password
- Create group
- Alter members in a group
- Sharing a link
- Sorting or filtering links
- React to a link
- Bookmark a link

The user interface\* will include

- Sign Up Page
- Login (and change password)
- Home interface which involves,
  - (a) list of buttons to all groups a user is a member of (and the direct messages to other users\*)
  - (b) button to create group (and message directly to other users\*)
  - (c) button to user profile
  - (d) search across all groups
- Group interface which involves,
  - (a) all the links shared in the group, hovering on them will show their description and clicking on them will show complete information

- (b) sharing a new link
- (c) sorting/ filtering links shared in the group
- (d) leaving the group
- (e) adding/ removing members (for the admins only)
- User profile interface which involves
  - (a) Username and option to change it
  - (b) Email ID
  - (c) list of all bookmarked links

## Other aspects:

We will use express sessions (as used in lab 4) for session management of users. We intend to create the app on Flutter so that it will work on web as well as mobile devices.