

1. Phase 1

The problem with current social media is that they are pretty good at connecting you with your inner but not very good at connecting you with people that share interest or passions. This can become an issue when you have a really cool idea or project in which you would like to collaborate with other people but your everyday friends don't share your interest or passion.

I proposed a solution to this problem through the creation of a temporary social media that helps materialize your projects and growth your personal network by motivating you to spread your ideas like wildfire. The idea behind it is to have ownership over your post and that it will have to meet a daily number of shares in order to stay online. There can also be an option that follows the kickstarter model where the user defines a goal of share in certain amount of time in order to keep the post public. In the case the post doesn't reach the goal it just gets erased. The way that it will work is by having a standard web interface (required) and a mobile web or app interface (desired) that will allow the user to:

- Create an account
- Create a profile page to have an identity behind his or her posts.
- Link his or her profile page with other mainstream social medias (Facebook, LinkedIn, etc.) in order to facilitate connection with other users.
- Create a post. The post can have text and images. Additionally there is a desired feature of supporting videos.
- Access page that will be enabled after a post reaches its daily quote for a week or reaches the declared goal in order to allow user to collaborate on the project.
- See other's users posts filtering by categories
- Comment on other's user's posts.
- Share and Promote other user's posts.
- Send a limited number of private messages to other users. In other to encourage public collaboration.

The main data required will be:

- User account information: username (primary key), password, image URI, real name, age, description.
- Post: postId (primary key), title, description, image URI, video URI, has it meet daily quote, today's shares count, time left to meet goal, current shares count
- Post Followers: username, role
- Post Contributions: contributionId (primary key), title, description, image URI
- Groups: groupId, goals, tasks

All this data can be obtain by following the MVC (Model-View-Controller) or the MVVM (ModelView-View-Model) patterns. Web technologies (HTML5, CSS, and JavaScript) can be used on the client-side to create presentation models. Then, this

With this description and options of technologies there will be a need for web developers for each layer of the stack (client, server, and database) and the setup for a web server running PostgreSQL or MongoDB.

```

    erDiagram
        User ||--o{ Post : creates
        User ||--o{ Follower : Follows
        User ||--o{ Group : paticipates
        User ||--o{ MediaPost : "isa"
        Contributions ||--o{ Follower : Makes
        Group ||--o{ GroupMessage : publishes
        Group ||--o{ Events : Hosts
        Group ||--o{ SmallerGoals : Has
        MediaPost ||--o{ Post : "isa"

        User {
            string Username
            string Password
            string Email
            string Pic_URI
            string Desc
            string UserId PK
        }
        Post {
            string PId PK
            string Content
            float Share_Count_for_deadline
            string Deadline
            bool isPermanent
            float Current_Goal
            float Total_Share_Count
        }
        MediaPost {
            string Media_URI
        }
        Follower {
            string email PK
        }
        Contributions {
            string CId PK
            string Content
        }
        Group {
            string GId PK
            string Name
            string Main_Goal
        }
        GroupMessage {
            string Mid PK
            string Content
        }
        Events {
            string Eid PK
            string Date
            string Place
            string Purpose
        }
        SmallerGoals {
            string GSId PK
            string Description
            string Deadline
        }
  
```

The Users have an account they have a username, password, email, personal description, a profile picture URI, and a unique identifier. They create posts, and admin or collaborate in a group.

The Groups have a Name, a main goal, and a unique identifier. A group might host event host events, have group messages/announcement that are only visible to users that are part of the group and have smaller goal that are milestone to its main goal.

A smaller goal has to be owned by a Group it has a deadline by when it should be accomplished a description and a unique identifier.

An event must be hosted by a Group it has a date, place, purpose, and a unique identifier.

A group message/announcement must be published by a group (specifically a group admin) it has a content and a unique identifier.

The Posts are the central concept of the application the ideas that ignite that once proven popular might lead to the creation of groups. A post has a unique identifier, a current goal of shares, a deadline to reach a goal of shares, share count for the current deadline, total shares count, and some text content. A user could also post a mediaPost, which is basically a Post with some multimedia content.

People that are interested in the message a post might become followers of that post. Ideally they will create an account and be active part of the app community but they just have to submit an email to follow a specific post.

Followers can also make contributions, which have a unique identifier, and some content to help define or improve the objective of the post they are following.

Schema

Users (UserId, username, password, email, picURI, desc)

Groups (GId, name, mainGoal)

GSmallerGoals(GId, GSId, description, deadline)

GEvents (GId, EId, purpose, place, date)

GMessages (GId, Mid, Content)

Posts(UserId, PId, content, cShareCount, deadline, isPermanent, cGoal, tShareCount, MediaURI)

Followers(PId, email)

Followers(email, Cid, content)

SQL

```
CREATE TABLE Users(  
  userId SERIAL PRIMARY KEY NOT NULL,  
  username VARCHAR(50) NOT NULL,  
  password VARCHAR(50) NOT NULL,  
  email VARCHAR(50) NOT NULL  
  picURL VARCHAR(200),  
  desc TEXT);
```

```
CREATE TABLE Groups(  
  GId SERIAL PRIMARY KEY NOT NULL,
```

Name VARCHAR(50) NOT NULL,
mainGoal TEXT NOT NULL);

CREATE TABLE GSmallerGoals(
GId SERIAL NOT NULL,
GSId SERIAL NOT NULL,
description TEXT NOT NULL,
deadline DATE NOT NULL,
PRIMARY KEY (GId, GSId));

CREATE TABLE GEvents (
GId SERIAL NOT NULL,
EId SERIAL NOT NULL,
place VARCHAR(100) NOT NULL,
purpose TEXT NOT NULL,
date DATE NOT NULL,
PRIMARY KEY (GId, EId));

CREATE TABLE GSmallerGoals(
GId SERIAL NOT NULL,
MId SERIAL NOT NULL,
content TEXT NOT NULL,
PRIMARY KEY (GId, MId));

CREATE TABLE Posts(
userId SERIAL KEY NOT NULL,
Pid SERIAL KEY NOT NULL,
content TEXT,
cShareCount INTEGER,
deadline DATE,
isPermanent BOOLEAN,
cGoal INTEGER,
tShareCount INTEGER,
MediaURI TEXT,
PRIMARY KEY (userId, Pid));

CREATE TABLE GSmallerGoals(
Pid SERIAL NOT NULL,
email VARCHAR(100) NOT NULL,
PRIMARY KEY (Pid, email));

CREATE TABLE GSmallerGoals(
email VARCHAR(100)) NOT NULL,
Cid SERIAL NOT NULL,
content TEXT,
PRIMARY KEY (email, Cid));

2. Survey:

1. What languages are you familiar with?

JAVA, C++, PHP, HTML, Javascript, CSS, and C#

2. Do you have any experience with databases?

Yes

3. If yes, please list any DBs you have used:

Microsoft SQL server and MongoDB

4. Have you developed an App (iPhone/Android)?

Yes, iPhone and Android but a lot more experience with Android than iOS

5. Have you used a version control system?

Yes, git and svn

6. Do you have experience designing websites? yes/no

Yes

7. Do you want to go to grad school?

Masters

8. What are your career goals?

Becoming an entrepreneur as technical co-founder

9. Have you taken software engineering?

Taking it this semester

10. What other courses are you taking?

Software Engineering, Object Oriented Programing, and Web Application Development.