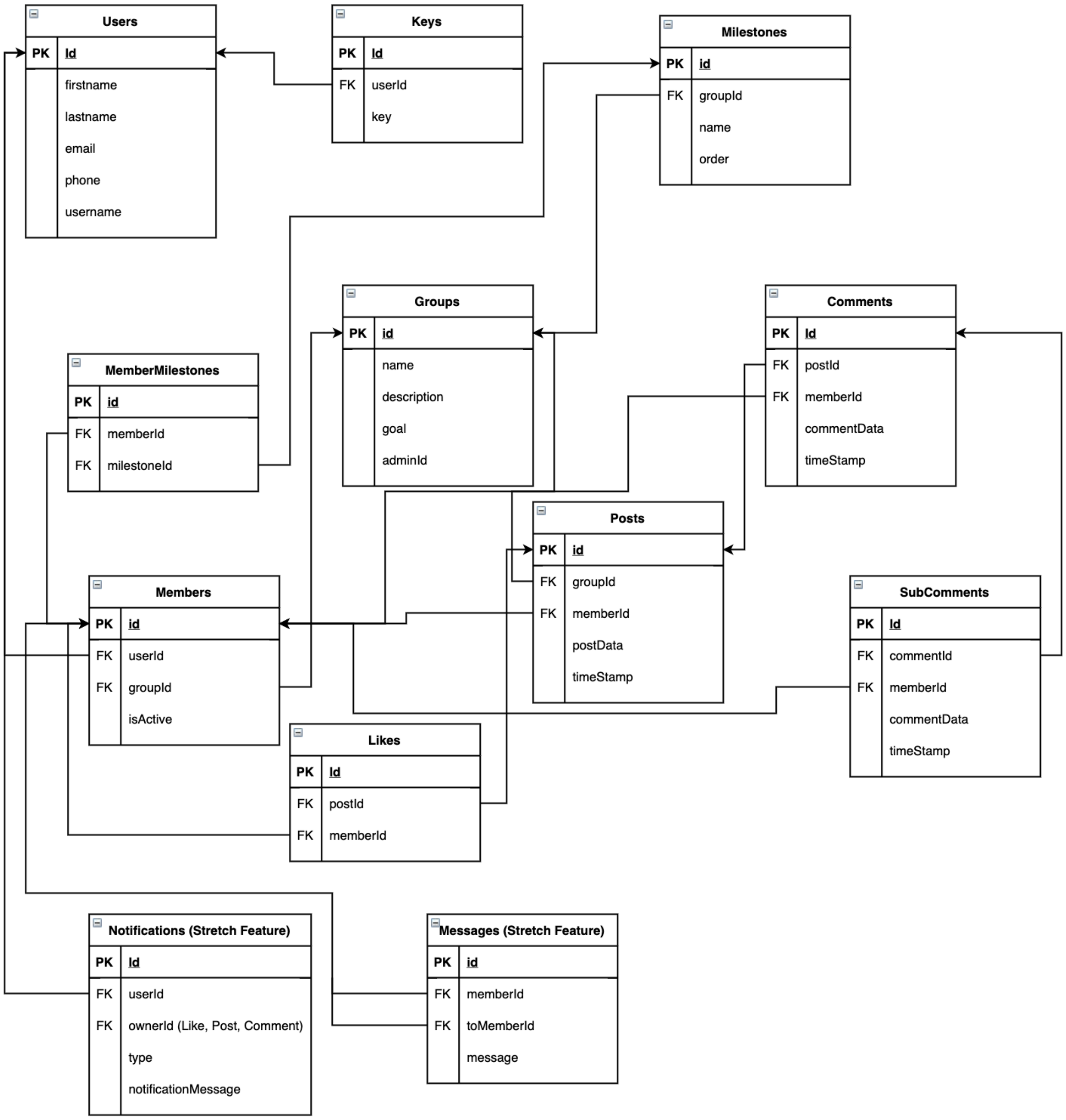
Database Revised

Database Technology

The database technology I am deciding to use will be MySQL for all data storage. I am not making any changes to my database technology. MySQL provides a wide range of connectors and drivers to connect my application in any way needed to my data. MySQL also provides great data protection for users information and password protection. Although my app will not have much personal data I still want to make sure I am able to properly secure people names and emails and phone numbers. User passwords will be hashed and salted before being stored within the database. That hash will then be used to compare to user input for proper authentication.



Database Structure/tables

Users - This table holds all data about each user of the app. When an account is created this table will be populated with that new registered member

Keys - This table holds the users password. When a new user account is created this table will be populated with the users password for authentication when logging into the web app. This table relates to only the users table as a One-to-One (one user has one password)

Groups - This table holds all the data about each group within the app. The description and the goal of each group are stored in this table. For this table the adminId is a one-to-many to the users table (one user can be the admin of many groups)

Members - This linking table establishes a many-to-many relationship with the user table and the group table. This is to show that may members can be a part of many groups. Each users gets a memberId for each group they are a part of. This table also keep track if a member decided to leave a group the status of them being active will be false. The default is true when the user becomes a member of a group.

Milestones - This table will store all milestones set up by the group admin for the group. Due to there being a possibility of a varying amount of milestones for each group this table has a one-to-many relationship with the groups table. This means that one group can have many milestones. Each milestones can only belong to one group. The admin will be able to change the milestones, update and delete them as well.

MemberMilestones - the data stored within this table will be the status of each members milestones within the group. As members strive through each milestones to reach the goals they will be able to mark what milestone they have reach and the group will be able to see each others progress. Milestones and Members tables Primary Keys are used create foreign keys to show each members progress. This table is updated when a member updates his/her progress in a group.

Comments / SubComments - this table(s) store all comments from the web app on posts and other comments. The relationship build with this table is a one-to-many with the Posts table as well as the members table. This means one post can have many comments and one member can make many comments. The sub-comments table hold all comments that are replies to comments within a post

Posts - this table holds all posts made by members in each group. This table has a one-to-many relationship with the members table to show that one member can make many posts and a one-to-many with the groups table to show that one group can have many posts.

Likes - This table will hold all likes a post will receive from users. Each member is only able to like a post one time or else that like will be removed if they like it a second time. This creates a many-to-many with members and posts because a member can like many posts and many posts can be liked by one member.

Stretch Features

Notifications - This table will hold all notifications that will be sent to each user that for them. This table has a one-to-many with the groups table, comments table, and posts table to show that each one of this entities can appear many times within a notification. This means that a notification record is created when user joins a group, requests to join a group, like a post, like a comment, and makes a post

Messages - This table holds all direct messages between members. This table replaces to the members table as only members are able to communicate with each that are in the same group. This may get replaced with a separate database one that is not relational so help with performance and response time.