Storytellin’

By The Whole Enchilada

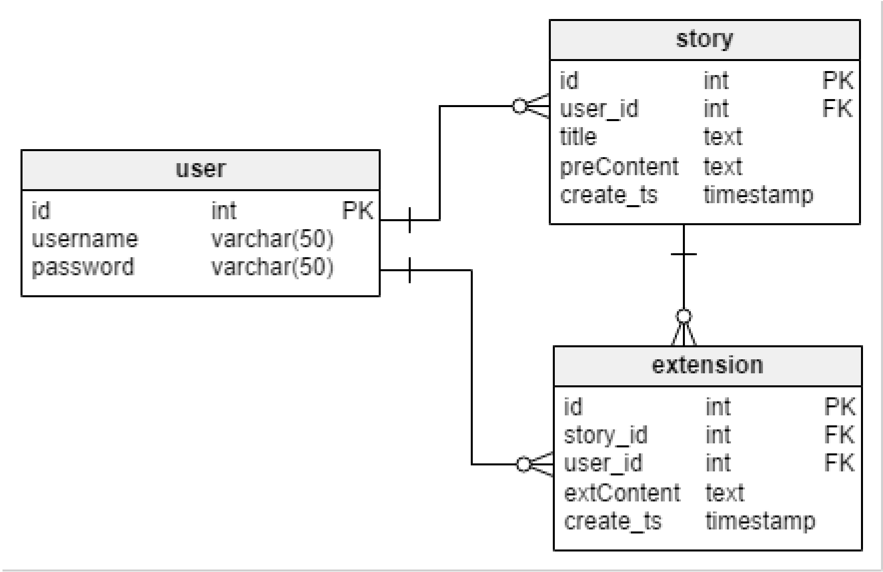
Kathy Lau, Daniel Monteagudo, Kelly Wang, William Xiang

Program Components

Users will have to register to use the site.

- Logged in users can either start a new story or add to an existing story.
- When adding to a story,
- Users are shown only the latest update to the story, not the whole thing.
- A user can then add some amount of text to the story.
- Once a user has added to a story, they cannot add to it again.
- When creating a new story,
- Users get to start with any amount of text and give the story a title.
- Logged in users will be able to read any story they have contributed to on their homepage (the landing page after login).

Database Schema



user

id	int	PK
username	varchar(50)	
password	varchar(50)	

story

id	int	PK
user_id	int	FK
title	text	
preContent	text	
create_ts	timestamp	

extension

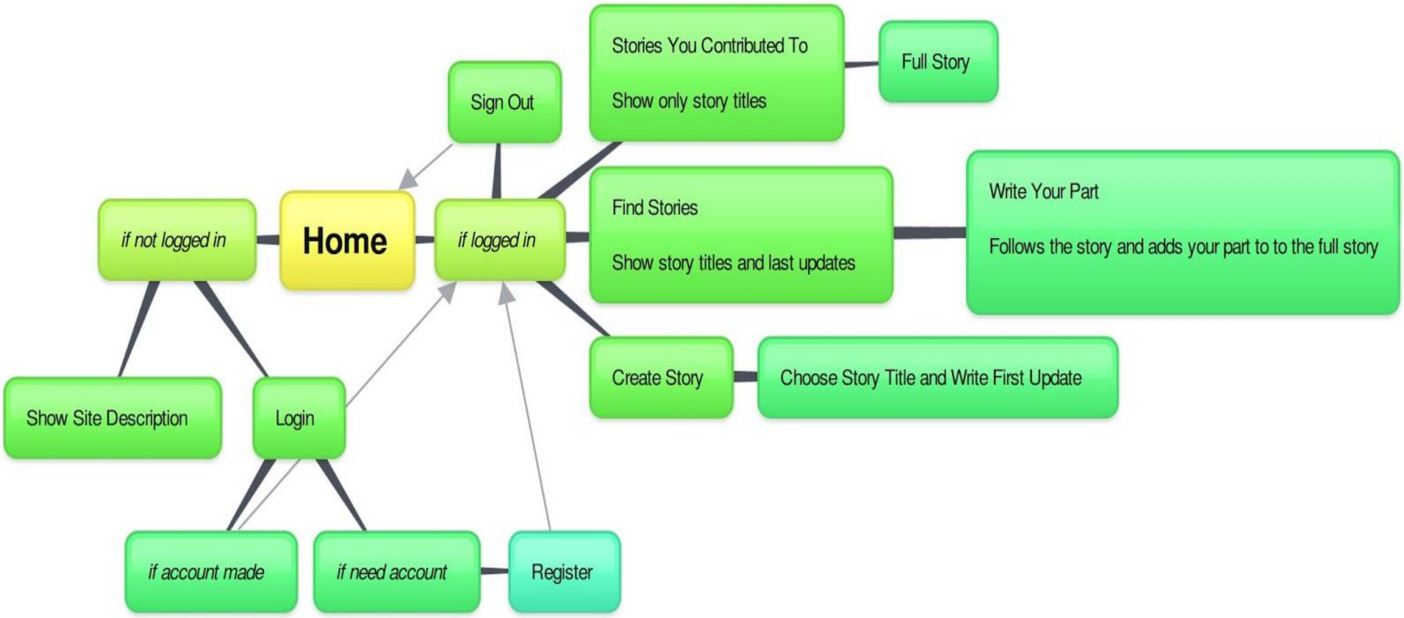
id	int	PK
story_id	int	FK
user_id	int	FK
extContent	text	
create_ts	timestamp	

Navigation

- Home page (index.html) will have register and login option
    - Register will require username and password, and python script will auto generate an id to add to the SQL table for users

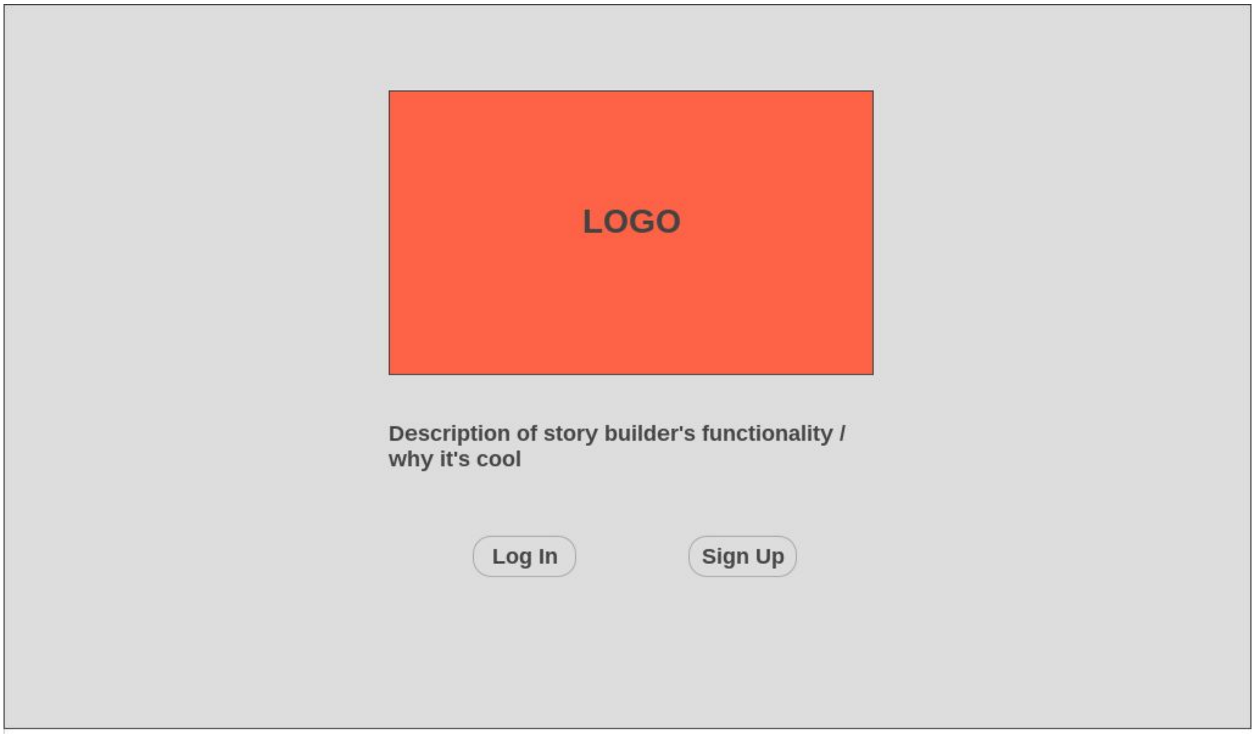
        - Login will check the SQL table for username and password match, store key in flask session, then redirect to main page if authenticated
- Main page will have options to start a new story, add to an existing story, or sign out
    - To start a story: will store id of story, user id, timestamp, title, and starting content in SQL table for stories
    - To add to existing story:
        - Will fetch only the most most recent update by any user to the story
        - What you input will be stored in SQL table of story id, user id, content, timestamp format
    - View other stories: will fetch full stories that have your user id as contributor
    - Sign out : Pops your key out of flask session
    - Additional Features
        - Admin Page that can take down stories, pin story titles (that you haven’t contributed to)
        - Upvote/downvote stories
        - Follow stories

Site Map

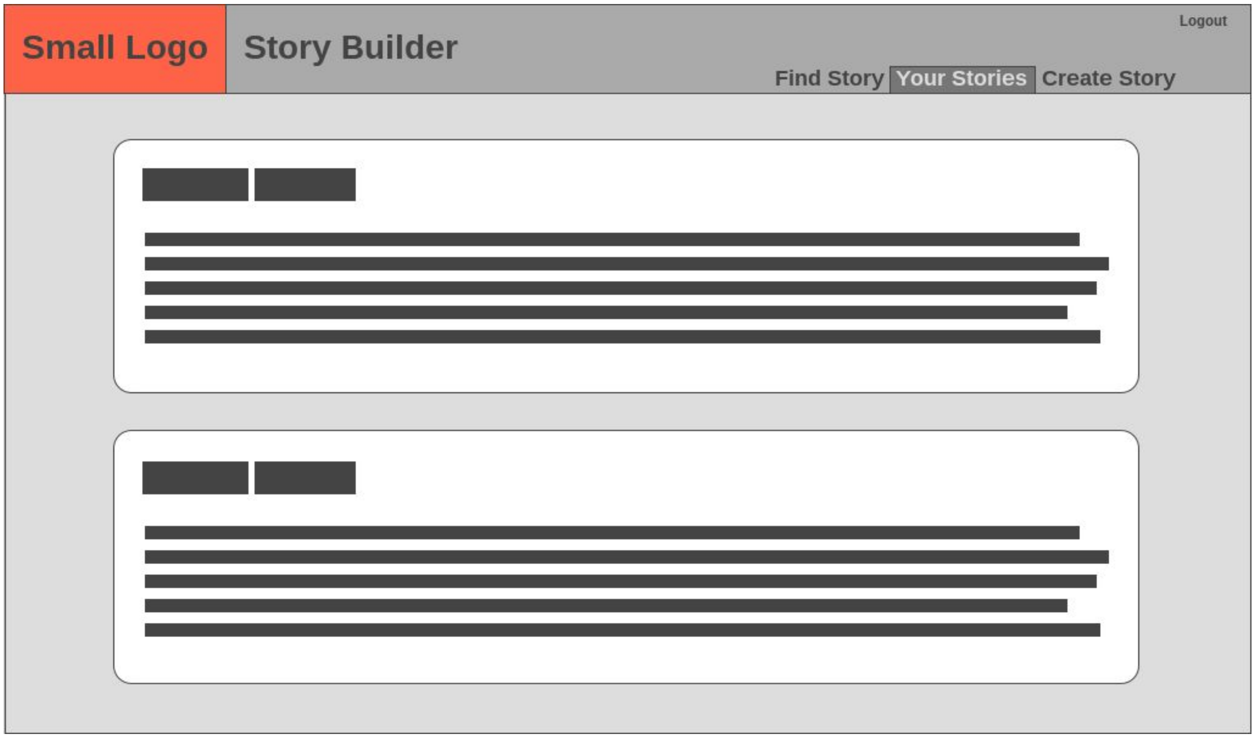


Page Mockups

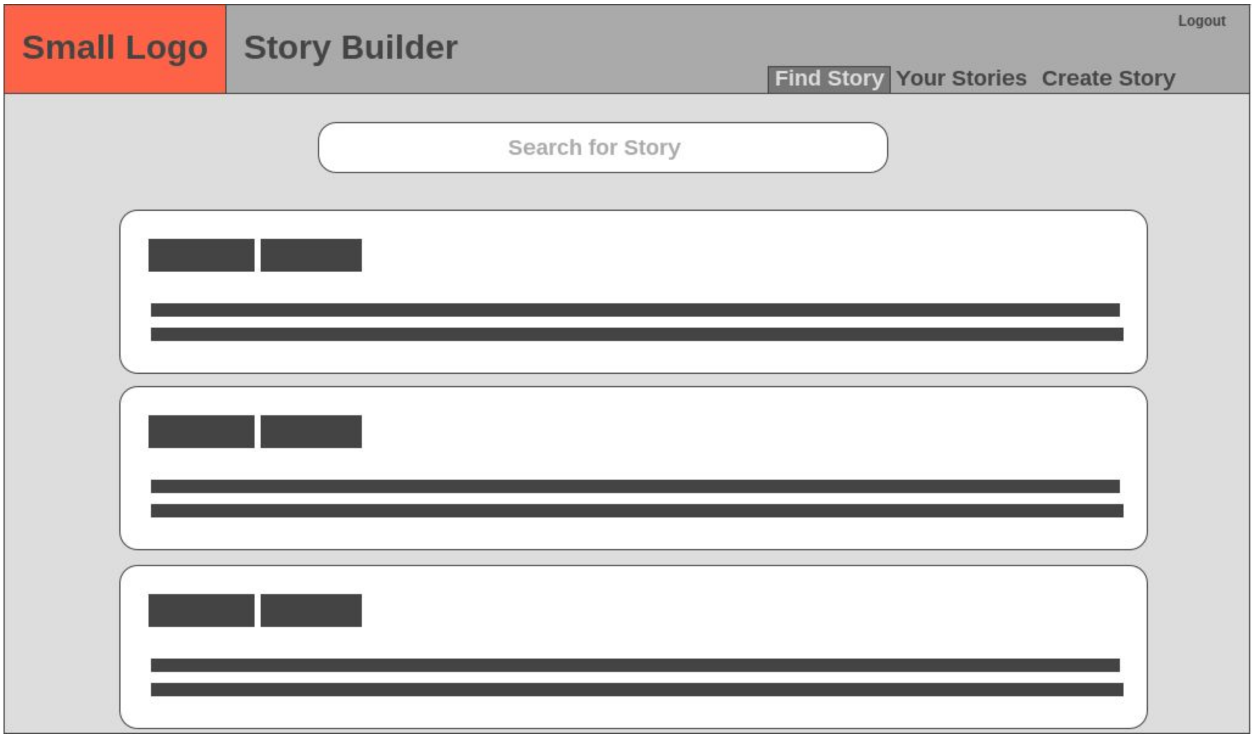
index.html


LOGO
Description of story builder's functionality / why it's cool
Log In
Sign Up

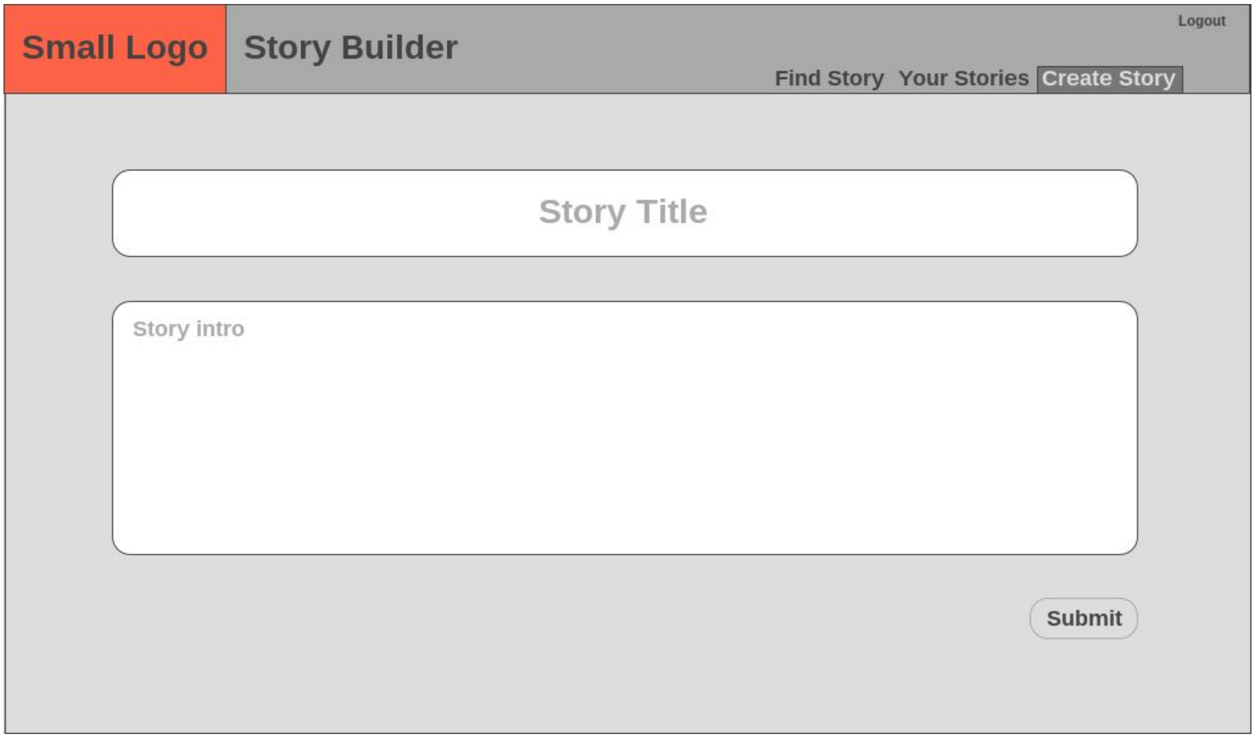
main.html


Small Logo
Story Builder
Logout
Find Story
Your Stories
Create Story

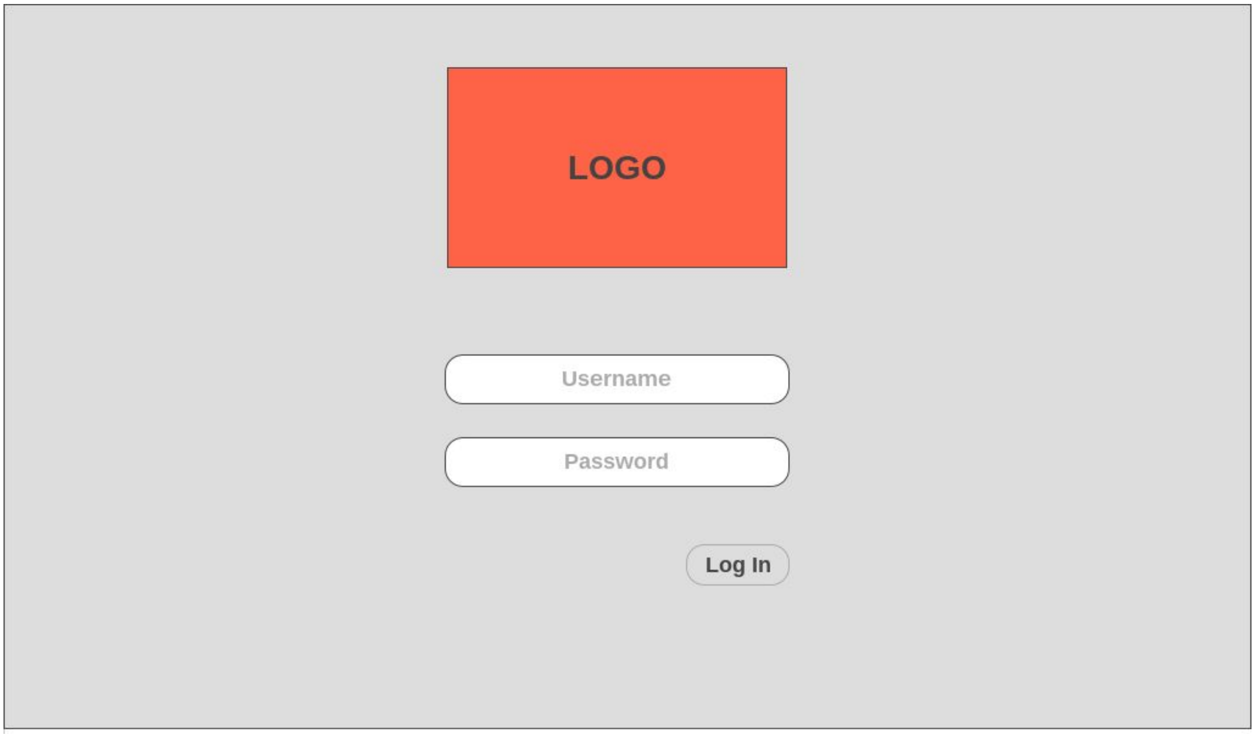
feed.html


Small Logo
Story Builder
Logout
Find Story
Your Stories
Create Story
Search for Story

create. html

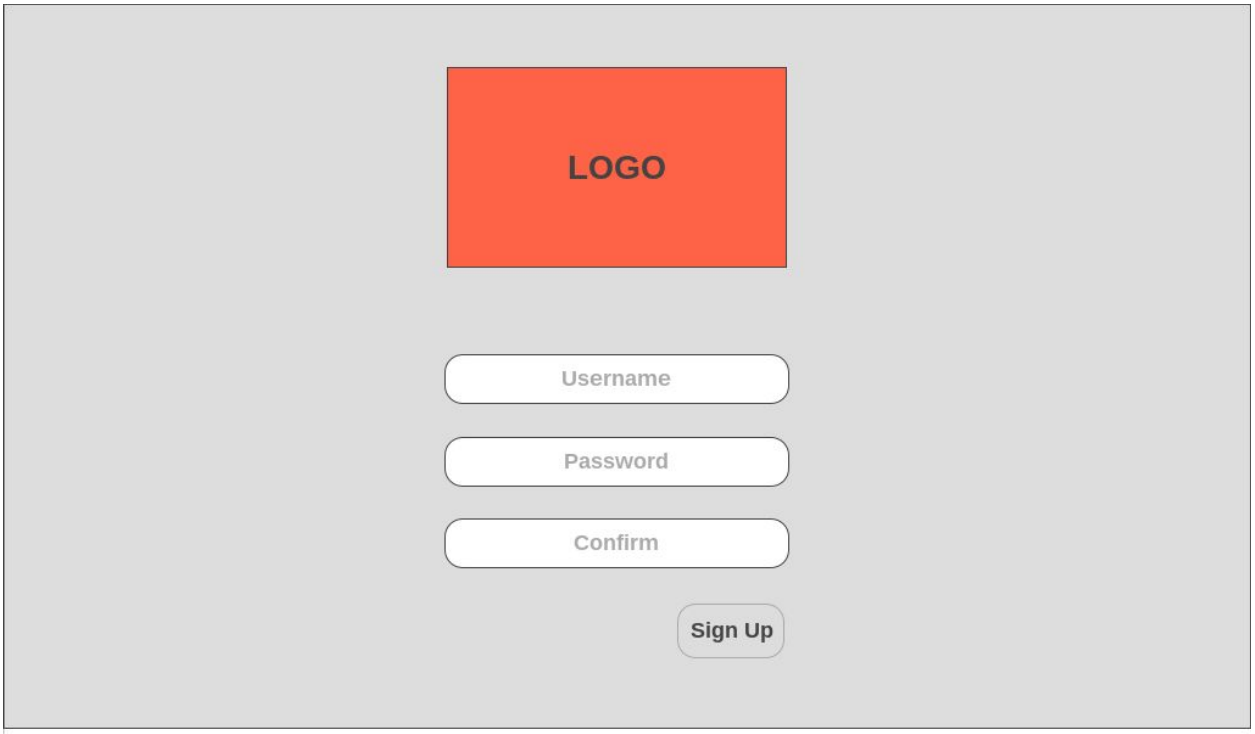


login.html



LOGO
Username
Password
Log In

signup.html



LOGO
Username
Password
Confirm
Sign Up

Interactive version here

Task Delegation

- **Leader + Middleware (Kathy)**
  - Makes sure everyone is on track
  - Glue UX, Flask, and SQL together
  - Test prototype frequently
  - Step in to code if needed
  - Modify plan as needed
  - Req: Be in frequent contact with all members
- **UX Designer (Daniel)**
  - Design general frontend (HTML, CSS, JS, etc.)
  - Format Content
  - Jinja2 templating
  - Ease of Navigation
  - Req: Must be in frequent contact with Leader
- **Python Developer (Kelly)**
  - Uses Flask microframework to set up web backend
  - Page routes and how they interact
  - Write functions to handle login, register, creating stories, fetching stories, extending, etc.
  - Req: Must be in frequent contact with Leader and DB admin
- **Database Administrator (William)**
  - Work with SQLite3 to write functional queries to hand off to flask dev
  - Structure database schema
  - Req: Must be in frequent contact with Leader and Python Dev