Members: Misha Kotlik, Zicheng Zhen, Seiji Yawata, Max Bertfield

Basic Features
- 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 and all the latest stories on their homepage (the landing page after login).
- Users will have access to stories they may contribute to on their homepage.

Tentative Features
- Users will be able to see all the posts they've contributed to on their profile.
- Users will have the ability to follow stories that they find particularly interesting.
- Users will have the ability to reset usernames and passwords through email confirmation.
- Stories will have tags associated with them to allow users to search through stories of a certain genre.
- Users will be able to randomly generate writing prompts to start off their stories.

Database Schema
- One database with three tables:
  - Users Table
    - username | password | email
      - username - username to track whether a user committed
      - password - hashed passwords of users
      - email - email address of user
  - Story Content Table
    - username | storyID | content | sequence
      - storyID - id for the sequence of content of a story
      - content - string (up to 200 characters) with text of the story
      - sequence - order in which content appears in a story
  - Story Identification Table
    - storyID | title
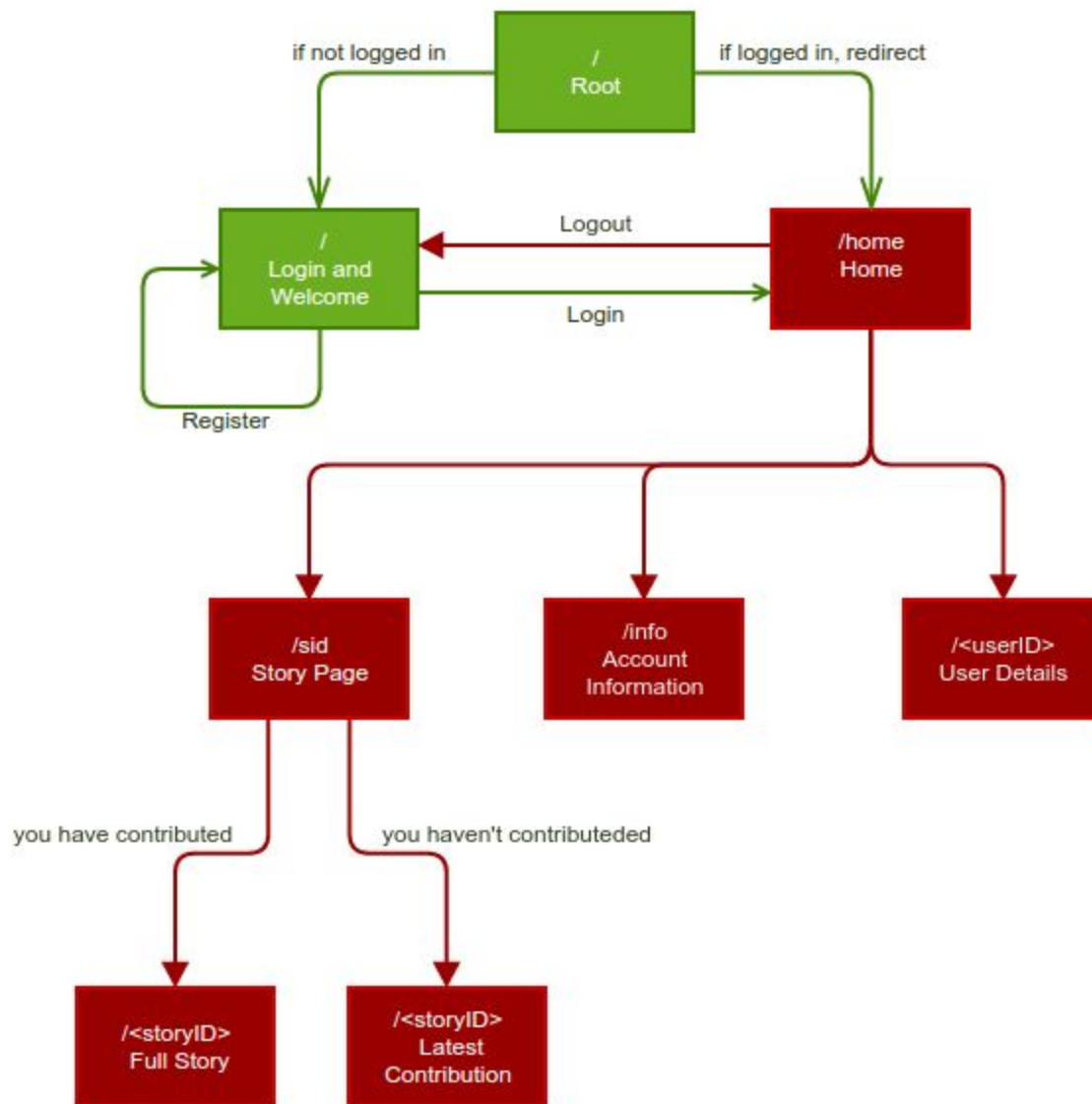   - title - title of story

Python Modules:
- app.py
  - Primary module of the server. Deals with and evaluates requests, server settings, sessions and cookies, and server initialization.

- ○ Includes appropriate modules, including registration and login (auth.py), content posting and retrieval (content.py) to help with data parsing.
- ● auth.py (/utils)
  - ○ Handles back-end involving login authentication and password hashing.
  - ○ Handles registration and login by reading from and writing to the database.
- ● users.py (/utils)
  - ○ Modifies user information such as username and email.
- ● content.py (/utils)
  - ○ Retrieves stories and stores contributions within the story databases.
  - ○ Mostly database editing.

Breakdown of Tasks:
- ● Misha:
  - ○ Project Manager.
  - ○ Back End - Manage user login and registration and define functions for hashing and storing usernames and passwords in the database.
  - ○ Back End - Define functions for modifying user information.
- ● Zicheng
  - ○ Back End - Create system of accessing and storing stories in the correct sequence depending on whether the user has contributed in the past.
  - ○ Back End - Create functions for easier modification of the SQL database.
- ● Seiji:
  - ○ Front End - Create HTML Templates, CSS Sheets, Javascript.
  - ○ Back End - Parse input from SQL databases to display stories properly using templates.
- ● Max:
  - ○ Front End - Generate effective and simple designs for each of the pages to allow for better user navigation.
  - ○ Back End - Control website routing and server management in app.py.

Site Map

Component Map