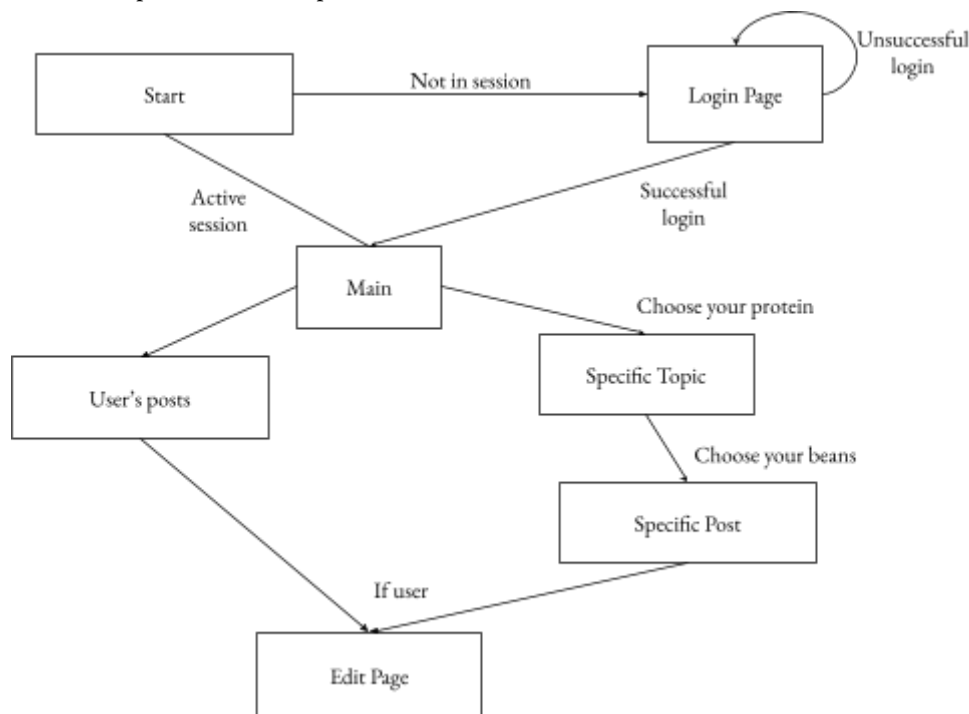


*Scenario Two: Your team has been contracted to create a **web log** hosting site.*

### Program Components

- Landing page is the login page
- Home page with topic selection dropdown, and option to create a new post
- Adding/editing/viewing entries
  - Adding a new post redirects to a different page, and once you submit it you can view your own post and edit it from there
  - You can click on the title of a post in the homepage to view it
  - You can both your own posts and others' from the home page, which you can return to from any page
- Checks if session is active
  - If somebody tries to look at a specific post and just goes to that site right away, they'll still be prompted to log in unless they are already logged in
  - Can logout if "logged in": pops session

### Relationship Between Components



We will be accessing the database when logging in and when looking at posts, which will allow us to access all the info but also to check whether or not a user can access the specific page.

## Database Organization

Tables for:

- Username: password
- Username : password : [blog title: blog post and id]

## Site Map

- “/” → landing (login) page
- “/main” → once logged in, the user will see a page with an introduction, header with links to different topics, and a way to look at their account and view/edit old posts or add new ones
- “/<topic>” → main page will link to a topic page, which will show all the posts that are under that topic
  - “/<topic>/<post>” → this will show a specific post on its own, <post> will be name/number of the post
    - If the specific post is your own, you will be able to edit it
- “/myposts” → this will show the user all their posts
- “/view/<post>” → this will allow the user to view a post;
- “/edit/<post>” → this will allow the user to edit a post; we can also use this when they add a post, except the post they’re editing will just start out blank

## Tasks

- Database
  - Add info into username, password database
    - Check for existing info
    - Add new info (from sign up) (no repeat usernames allowed)
    - Use for log in, check if pair works
      - Return wrong pass
      - Enter username
      - Enter pass, enter matching username and password
    - Password confirmation for signup
  - Database for text for blog
    - Add new with topic, title, and content (linked to username)
- HTML
  - Show text for blog
  - {{username}}, {{topic}}, {{title}}, {{postcontent}}
- FLASK
  - Add to database, rid of it, check info in database, use info from database for HTML (to display the blog)

## *Sean’s (and Patrick’s) Tasks*

- Database management

- Making database.py to simplify using the database in \_\_init\_\_.py and make using sqlite easier

#### *Ella's Tasks*

- Frontend design and component management
  - Signup and login authentication
  - Moving between pages
- PM duties

#### *Shyne's Tasks*

- Frontend design and component management
  - Creating posts
  - Accessing posts under different topics

#### *Tina's Tasks*

- Frontend design and component management
  - Editing posts
  - Creating more new posts after submitting
- User session technicalities (popping sessions, prompting logins, etc.)