Indigo (Stanley Hoo, Jacob Lukose, Naomi Lai, Colyi Chen)
SoftDev
P00: Move Slowly and Fix Things
TARGET SHIP DATE: 11/11/24

Scenario 1

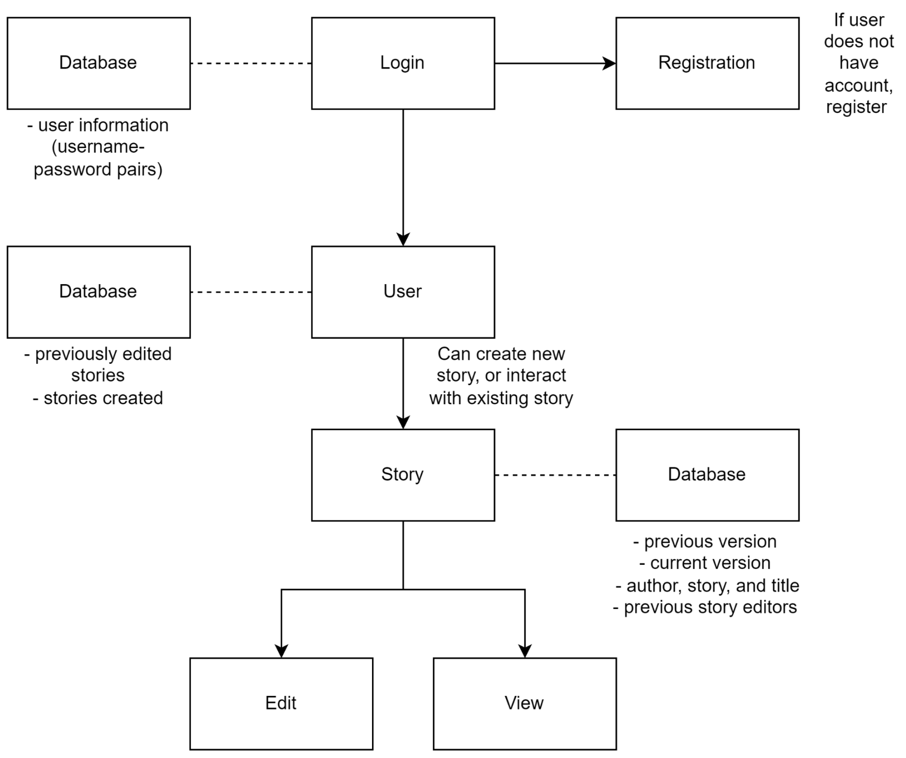
Components

- User accounts
  - login/logout
- Route for users to create stories and add to our page.
- Route for users to add to stories
- Route to view stories once you've edited it
- Search bar to search for stories with keywords
- SQLite Database with a unique value for each story and a list of the usernames of everyone who's already created a story.
- A page where those not signed up can sign up, and if already signed up, you can log in.
- A homepage that everyone has access to when logged in.
- Templates for:
  - Homepage
  - View Story page
  - Edit Story Page
  - Create story page
  - Signup page

Component Relations

- User accounts are linked to specific data in the database
- Routes for users to create stories – users must be logged in to create stories
- Route for users to add to stories – only users can add to stories. We will also check the list of users who already edited the story, to make sure that they do not edit it again.
- Route for users to view – displays full story if logged in and contributed to story, otherwise last update.
- The search bar will have to retrieve the stories from the database, may or may not implemented if we have time
- SQLite database will store all of the story information, past edit history, story IDs, and user info. Separate ones for users and story attributes/ history.
- Route to view story will retrieve info from databases
- Sign up/login page will send and retrieve data to databases
- Homepage will retrieve data from the databases (stories)
- Templates allow us to use the same html code for multiple pages easily while changing some variables

Component Map:


Database
Login
Registration
If user does not have account, register
- user information (username-password pairs)
Database
User
- previously edited stories
- stories created
Can create new story, or interact with existing story
Story
Database
- previous version
- current version
- author, story, and title
- previous story editors
Edit
View

All users can view the page, whether or not they are logged in. If not logged in, they will only be able to view the last edit of the stories, if they try to edit will prompt them to login. If they are logged in, will have 2 separate pages: one that has full stories that they've already edited and one that only has the last edits of the stories they haven't added to yet. All of this will be managed by sending and retrieving data from the database.

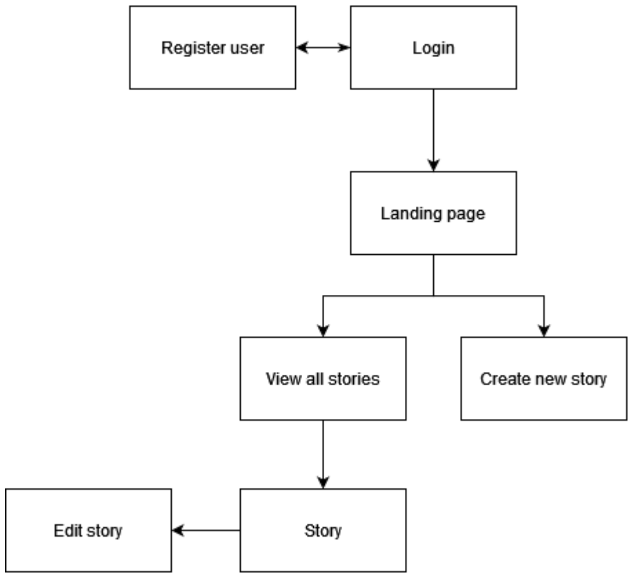
Database Organization:

- 3 tables:
  - User info
    - Tracks their usernames and passwords, all usernames must be unique
    - We will assign a unique ID to each user to make it easier to access later
    - Here we will also store all of the story IDs that the user has edited so each user can easily access them if need be.
  - Story info
    - Here will we track the story info
    - Each story will receive a unique ID (AUTO INCREMENT)

        - Also store ID of each user that has edited in the past along with what they edited and the order in which it was edited (ie. user1 edits first, user2 edits 2nd, etc. This could allow for better content moderation.
        - Store the story title
    - Page info
        - Here we will store the reference info for each created story, including unique ID, users edited, and data created
        - Makes it easier to display and retrieve stories from the stories table by linking it with a story ID

Site Map

- /home: Homepage where user can log in with their existing username+password, login container in top right
- /register: creates username password; username cannot already exist
- /create_story: users can create a title, edit story text, and submit the story
- /view: just displays a story (title, author, story, date of most recent change, add to story)
- /edit: edit existing story
- Note: No login route needed, we will create a login container on the homepage where users can login



Tasks:
Storing and retrieving info from the databases – Stanley
Python files to add, display, and webpages – Jacob/Naomi
Html templates, CSS, user accounts - Colyi (non existent person)