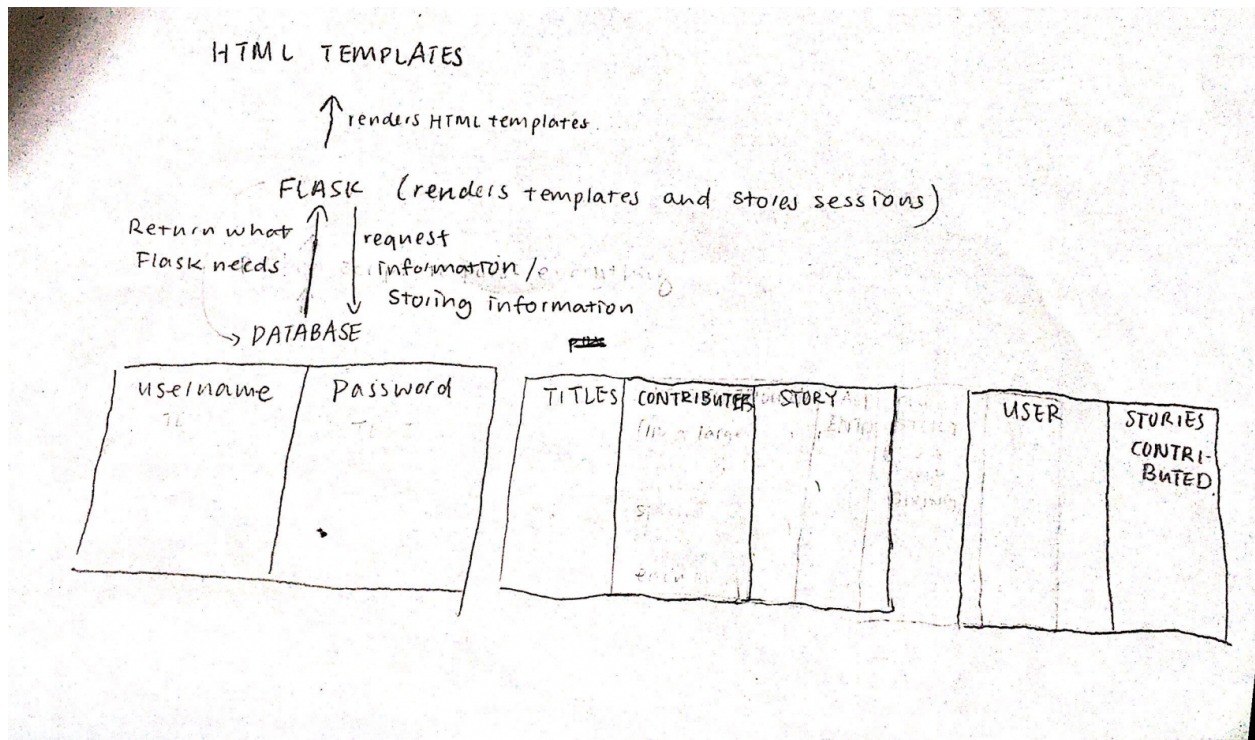Team Redacted: Rachel Xiao, Michael Borczuk, Yuqing Wu, David Chong
SoftDev
Project 00 -- Move Slowly and Fix Things
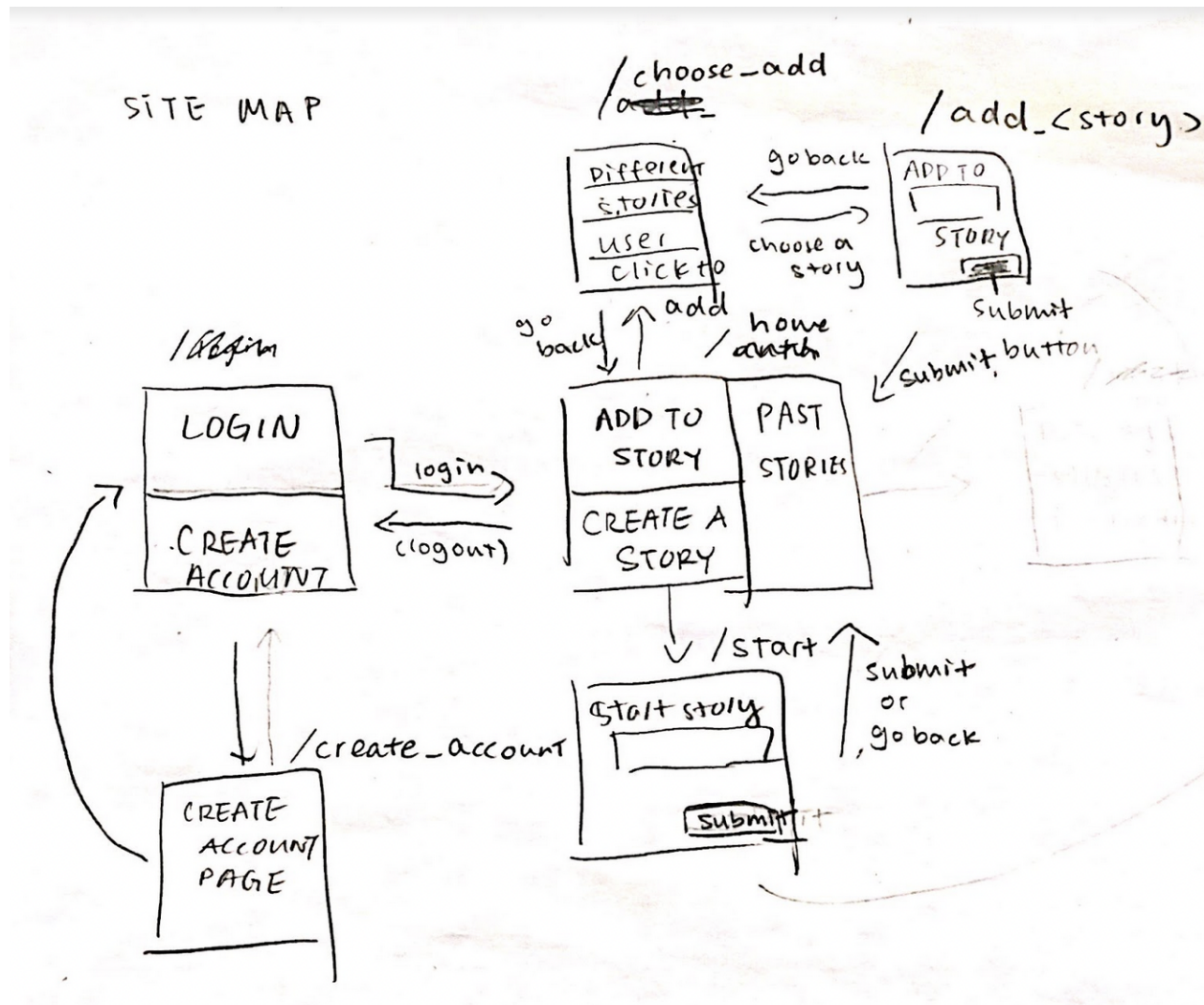2021-10-27

Program Components:

- HTML
    - Templates for Login Page, Create Account Page, Home Page, Choose Story to Add, Add to Story Page, Create New Story, View Stories Page
        - Buttons/links to move from page to page
- Flask
    - Getting password/username input from HTML (to verify against the database)
    - Storing session (so user stays logged in)
    - Render the HTML templates, filling them in using data from the SQL database
- SQL
    - Database stores pairs of usernames and passwords
    - Store story title, the stories, and users contributed to the story in large strings separated by special characters. Possibly an ID field so that stories can have the same titles because they have different IDs.
    - Stores list of stories users contributed to.
    - Functions to write to the database / get things from the database

Components Map:

HTML TEMPLATES

↑ renders HTML templates

FLASK (renders templates and stores sessions)

Return what ↑ request
Flask needs information /
              storing information

↳ DATABASE

username	Password

TITLES	CONTRIBUTER	STORY

USER	STORIES CONTRI-BUTED

Database organization is shown in the tables of the components map

Site map:

SITE MAP

/choose_add
/add

/add_<story>

Different stories
user click to add

go back

choose a story

ADD TO
STORY
Submit

/login

LOGIN

login →

← (logout)

go back ↑ add

home
/auth

ADD TO STORY | PAST STORIES

CREATE A STORY

Submit button

.CREATE ACCOUNT

↓ /create_account

CREATE ACCOUNT PAGE

/start

Start story

Submit

Submit or go back

Breakdown of each page:
- Login Page @app.route("/")
    - Template for Login Page
    - Database to store Login Information
    - Render template with flask and store cookies
    - Redirect to ("/create_account") if user clicks the create account button
    - Redirect to ("/") when the user successfully logs in
- Create an account ("/create_account")
    - Use flask to store the password and username and confirmed password
        - If both password entries are the same, then store the password
        - If the username is unique, then store it in the database

- Error messages if username already exists or if the passwords don't match
    - After creating an account, the user is redirected to the "/login" page
- Home Page ("/home") → landing page after login
    - Contains links to add to or create a story
    - Redirect to:
        - Create Story Page ("/create")
            - Uses an HTML template to take inputs for the "title" and "content"
                - Check if "title" is unique"
        - Add to Story Page ("/add")
            - When user clicks on certain stories would redirect to "/add_<story>"
                - Display the latest entry in the database
            - Form to input a limited amount of text in the story
            - Store the story title into the database for that user
    - View stories that they have contributed to
    - Preview Featured Story (advanced feature)

Tasks:

HTML: Create simple and elegant templates for each page, creating an easy to use format for the python backend to display data. Current pages are the login page, home page, create and add page, and create account page → David

Flask: Bring information from the database to HTML and vice versa. Maintain session login and redirect users accordingly → Rachel

SQL: Store login information when creating a new account. Store stories information and create functions to access the correct information (like most recent addition to a story) → Michael and Yuqing