Bug Busters: Victor Casado, Ethan Sie, Marco Quintero, Qianjun Zhou
SoftDev
P00: Move Slowly and Fix Things
2024-10-28
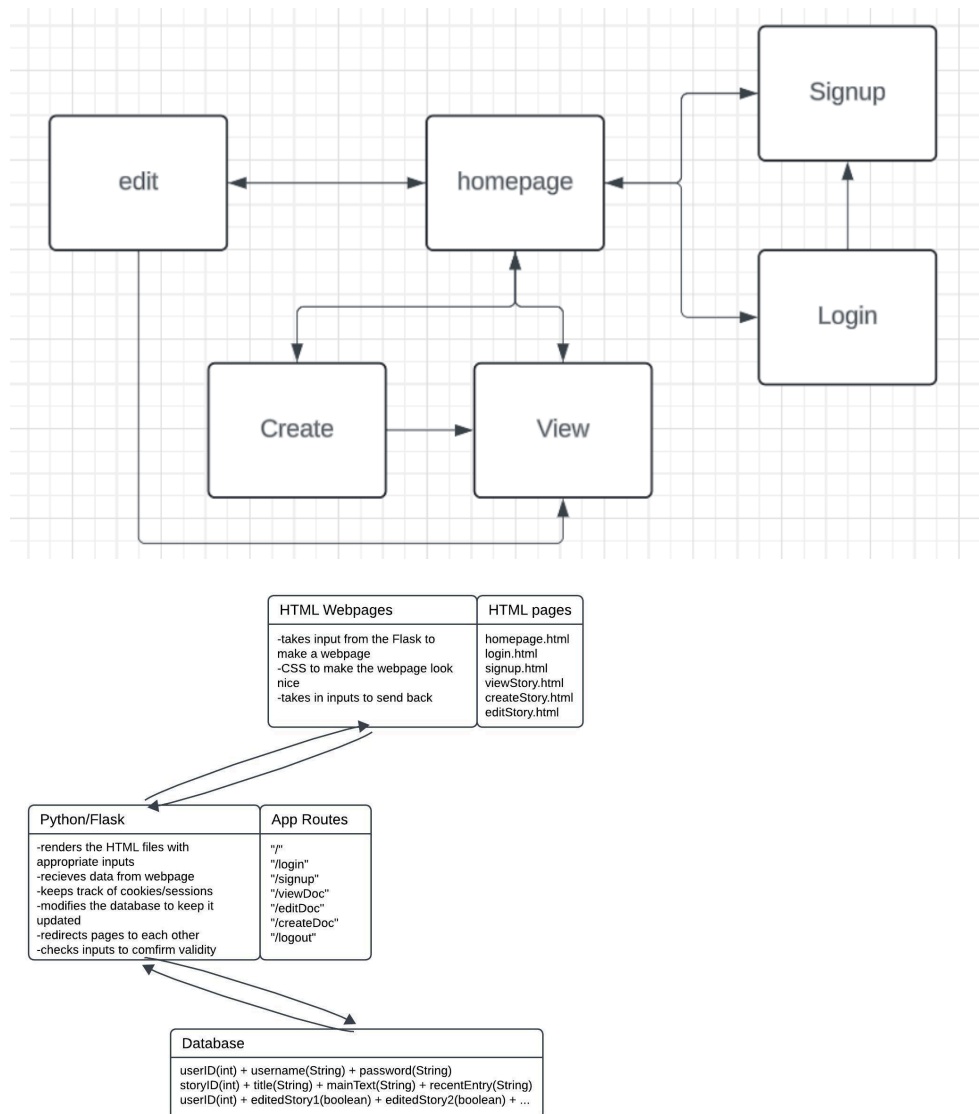**TARGET SHIP DATE: 2024-11-11**

Program Components
Frontend Components:
1. HTML
     a. homepage.html — Displays a list of stories and provides navigation options.
     b. login.html — Allows users to log into their account and contains a button to go to signup.
     c. signup.html — Enables new users to create an account.
     d. viewStory.html — Shows a story in its entirety, including the title and main text.
     e. createStory.html — Provides a form for users to create a new story.
     f. editStory.html — Allows users to edit a story once.

Backend Components:
1. Flask/Python - Stored session with username, password, and if we just signed in (a Boolean)
     a. "/" (Root)
          i. If no session found, redirect to login
          ii. If session found, if we just did login, if user/password combo is correct, do not redirect
          iii. If session found, if we just did login, if user/password combo is incorrect, redirect to login
          iv. If session found, if we just did sign up, if user/password combo is unique, add user/password to db
          v. If session found, if we just did sign up, if user/password combo is not unique, redirect to login
     b. The remaining are described through their names

2. SQLite DBs
     a. User Table: Userid (unique integer), username (unique string), password (string)
     b. Stories Table: story id (unique integer), title (string), main text (string), most recent entry (string), creator (Userid from User table)
     c. Stories Contributed To: user (Userid from User table), edited story1? (boolean), edited story2? (boolean), ... (each column represents one story)

Site and Component Maps:

HTML Webpages	HTML pages
-takes input from the Flask to make a webpage -CSS to make the webpage look nice -takes in inputs to send back	homepage.html login.html signup.html viewStory.html createStory.html editStory.html

Python/Flask	App Routes
-renders the HTML files with appropriate inputs -recieves data from webpage -keeps track of cookies/sessions -modifies the database to keep it updated -redirects pages to each other -checks inputs to comfirm validity	"/" "/login" "/signup" "/viewDoc" "/editDoc" "/createDoc" "/logout"

Database
userID(int) + username(String) + password(String) storyID(int) + title(String) + mainText(String) + recentEntry(String) userID(int) + editedStory1(boolean) + editedStory2(boolean) + ...

Tasks:

HTML/CSS (makes website look cool and displays all vital information): Victor

Flask/Python session-based logic (allows you to log in and stay logged in): Ryan

Flask/Python other logic (connects HTML code with the SQL code so website has information): Ethan

SQLite (handles storing of data and creates simple data manipulation methods): Marco