

Go Jose!

Gordon Mo, Joshua Liu, Selena Ho

SoftDev

P00 -- Design Doc

2022-10-28

Time spent: 3

Target ship date: 2022-11-18

For **Scenario 1**

Program Components:

- HTML:
 - Login Page (Landing Page)
 - Html form for username and password and submit button for logging in.
 - Link to redirect user to create an account if they don't have one → leads to a registration page
 - Register Page
 - Form for username, password, and confirm password.
 - Submit button sends the user to the home page and registers them in the users table of the db. (also updates cookies, same for logging in and out)
 - Updates the db, registering the user.
 - * Usernames cannot be repeated or < 2 chars, passwords cannot be empty
 - User's Personal Home Page
 - Section for "You've Collaborated On"
 - Shows all the stories that the user has added to or created, allowing them to only read the story, NOT edit. Shows the whole story
 - Section for "Existing Stories"
 - Shows the stories that the user has NOT added to and pressing on an edit button next to the title of each story sends the user to the Edit page. Shows only the most recent edit.
 - Button at the top of page for new story
 - Button redirects users to add story page
 - Button for logging out
 - Edit Page
 - Displays the most recent edit and title of the story
 - Uses HTML form (POST)
 - Textarea for the user to add to the story
 - Submit button, updates the db and sends the user back to the home page
 - Add Story Page
 - features an HTML form (POST) with a textarea for user to write their story and a text box for the title of the story
 - Submit button at the bottom of the page

- Flask
 - Used for when users want to edit a story, login, register, and ties all the pages/links together
 - Handles POST requests to save and display the user's edits by updating the database's content.
- SQLite3
 - Used to create the tables that will store the information
 - Tables:
 1. User Table: user id | username | password
 2. Story Table: story id | story title | entire story | most recent edit
 - Every new submission to a story overrides the data in the most recent edit column so the table only keeps track of the most recent edit
 - Use update instead of insert when simply updating the table rather than adding something new to it
 - Story id will be based on the order in the table. For example, the first story will have an id of 0, the second will have an id of 1, and so on.
 3. Contributed to table: user id | story id
 - Every entry in the table features a user and a story id so we can query through to see which stories one particular user has already contributed to.
 - Composite key of uid and sid
- Multiple Python files.
 1. One file dedicated for Flask related actions: handling form requests and button clicks.
 2. One file for database updating (when username/password added for example) fetching data, and inserting data

Tasks Required For Project Completion:

- Create HTML files for each of the pages
- Create python file for Server-side processes
- Create python file for database updating and retrieving
- Add CSS last IFF all required elements of the project are functioning

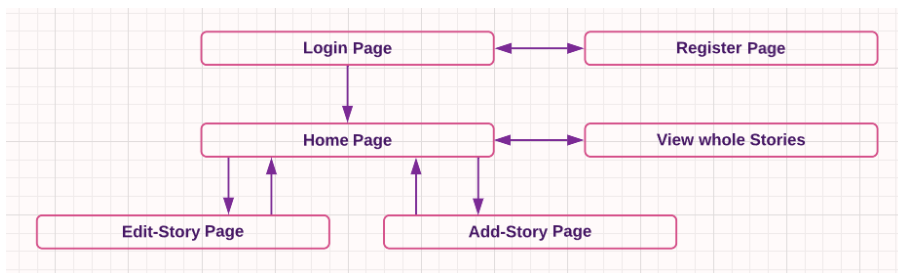
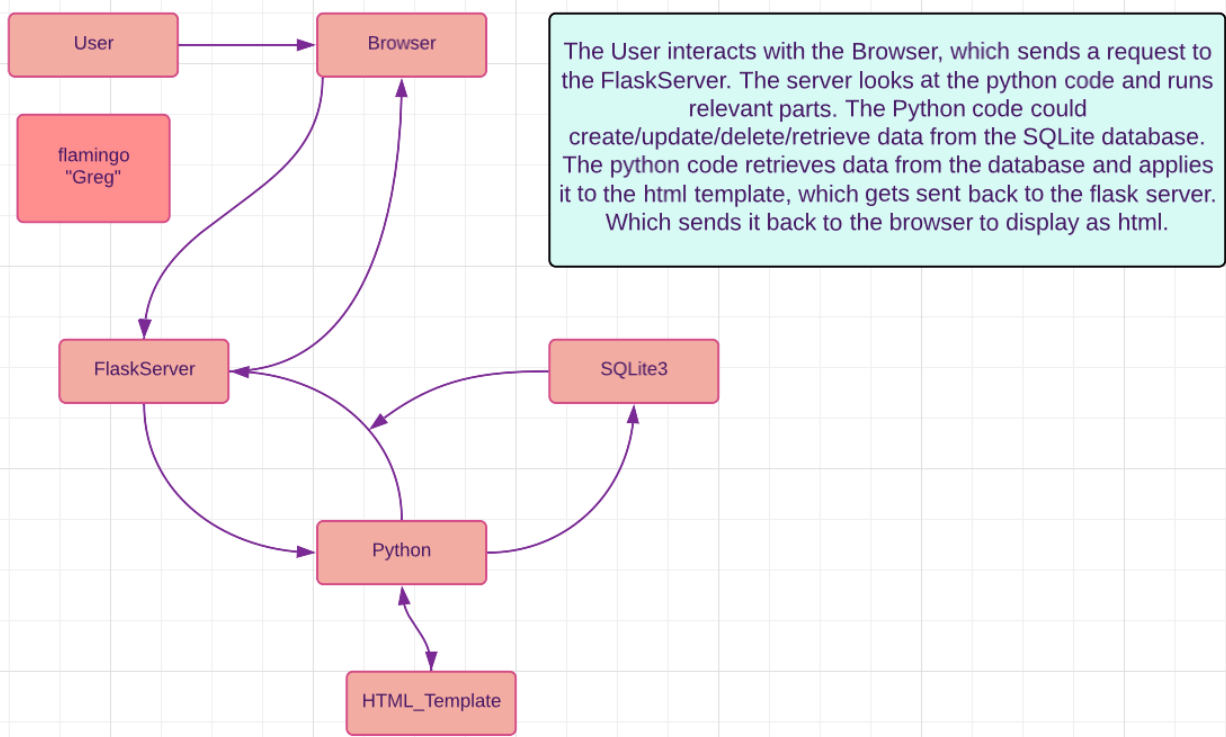
The Order of Operations:

- Display basic home page w/ Flask
- Create login-related db and code
- Create register capabilities
- Allow for the adding of stories to the db
- Display the Home page based on user (separate collaborated on and other stories)

- Allow for editing of stories

Breakdown of Tasks:

- Joshua does Flask- related python code
- Gordon does SQLite-related python code
- Selena does Frontend and SQLite
- But, each of the different tasks are connected so everyone may help/work on different aspects that aren't directly assigned to them



user		
PK	uid	integer
	username	varchar(20)
	password	varchar(30)

contributions		
CK	uid	integer
CK	story_id	integer

content		
PK	story_id	integer
	story_content	varchar(MAX)
	story_title	varchar(50)
	last_edit	varchar(MAX)