

Go Jose!

Gordon Mo, Joshua Liu, Selena Ho

SoftDev

P00 -- Design Doc

2022-10-28

time spent: 1.6

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 (should tell the user if the username already exists and send them back to the form)
 - Link to redirect user to create an account if they don't have one → leads to a registration page
 - User's Personal Home Page
 - Section for "You've Collaborated On"
 - Shows the stories that the user has added to and clicking on a story allows them to only read the story, NOT edit
 - Section for "Existing Stories"
 - Shows the stories that the user has NOT added to and pressing on one of the stories redirects the user to edit page for that specific story
 - Button at the bottom of page for new story
 - Button redirects users to add story page
 - Edit Page
 - Displays the most recent edit
 - Uses HTML form
 - Textarea for the user to add to the story
 - Submit button
 - Add Story Page
 - features an HTML form with a textarea for user to write their story and another for the title of the story
 - Submit button at the bottom of the page
- Flask
 - Used for when users want to edit a story
 - A POST request is used to save and display the user's edits by updating the databases content data which contains the story in its entirety and also another table which contains only the most recent edit
- SQLite3
 - Used to create the tables that will store the information

- Tables:
 1. User Table: user id | username | password
 2. Page Name and URL
 3. 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.
 4. 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.
- Multiple Python files
 1. One file dedicated for Flask related actions
 2. One file for database updating (when username/password added for example)

Tasks Required For Project Completion:

- Create HTML files for each of the pages
- Create python file for Flask
- Create python file for database updating
- Add CSS last IFF all required elements of the project are functioning

What We're Doing First:

- Display basic home page w/ Flask
- Create login-related db and code
- Create story-related db and code

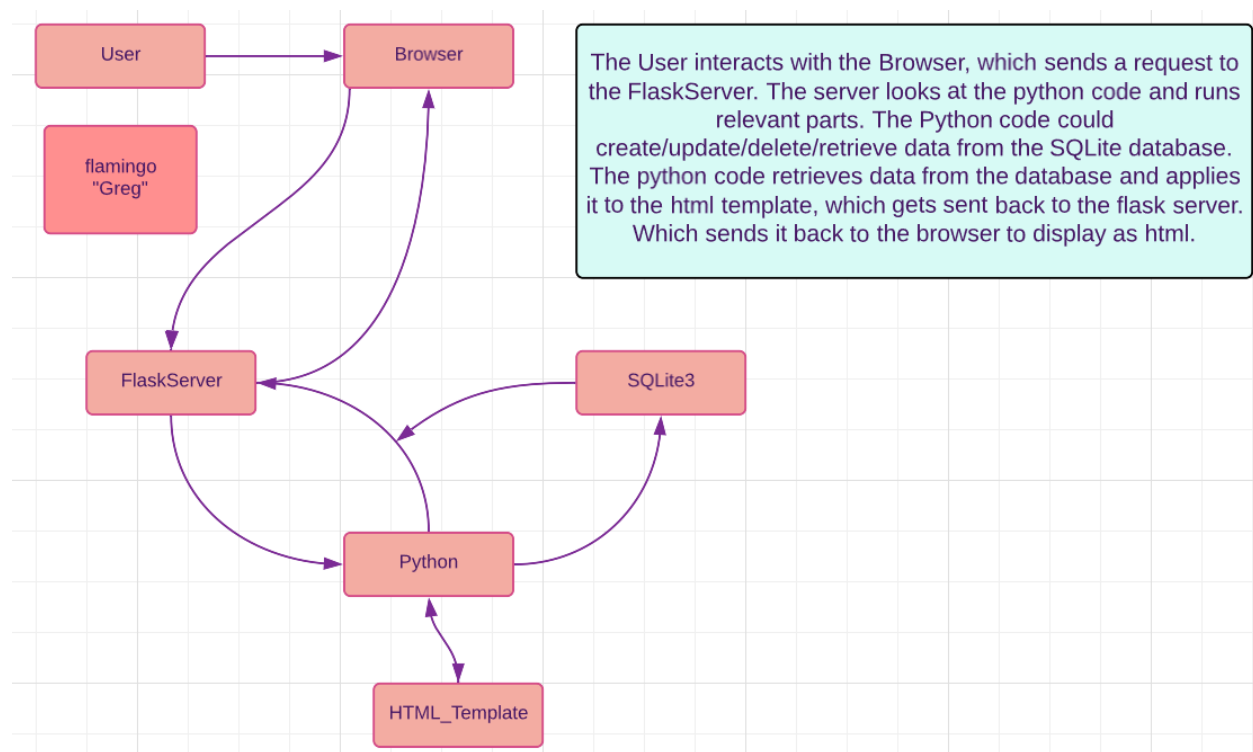
Breakdown of Tasks:

- Joshua does Flask
- Gordon does SQLite
- Selena does HTML
- But, each of the different tasks are connected so everyone may work on different aspects that aren't directly assigned to them

Extra Ideas:

- Search for certain stories by story_id
- Max word count when contributing to a story (Not sure if we want this)
- A way of viewing how many contributions a story has in total.

Component Map:



Tables:

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

page		
PK	page_name	varchar(20)
	url	varchar(50)

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)

Site Map:

