

Team HotCacao

Devin Lin, Alice Ni, Joseph Yusuf, Hilary Zen

Core Functionality

- Login / registration system
- Cookies to store data
- Start a new story
- View all stories
- Record who has added to what
- User homepage/dashboard

Roles

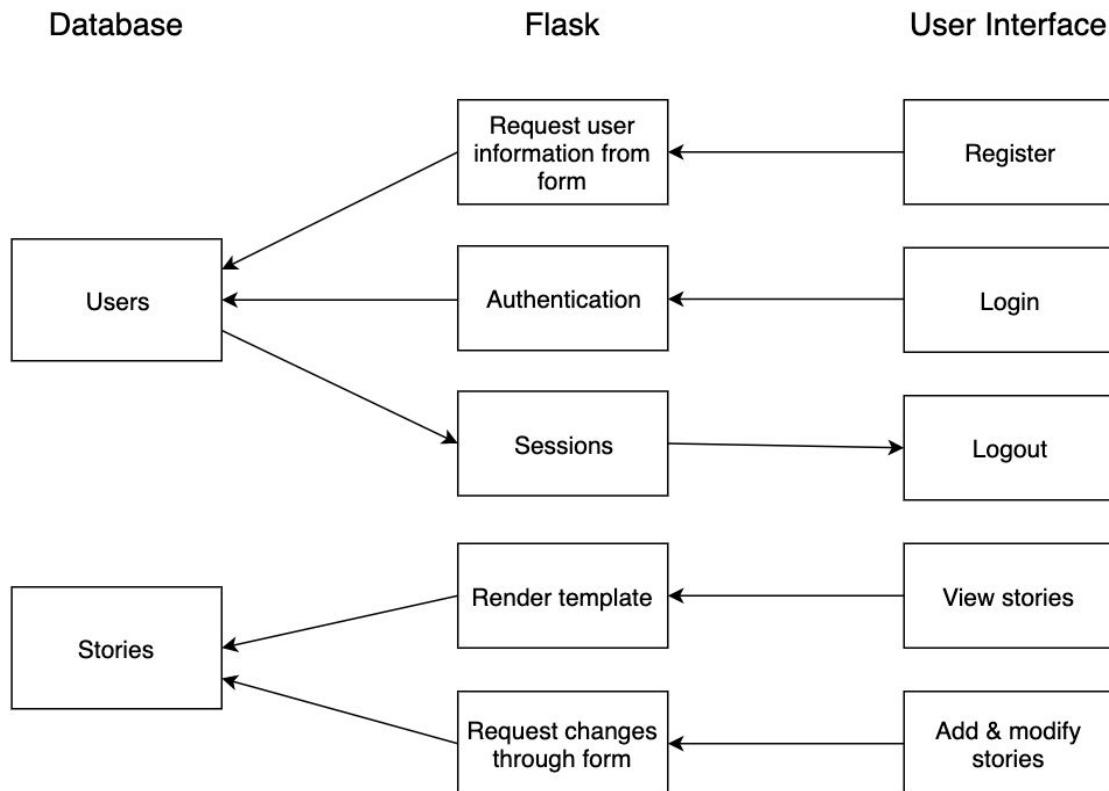
Devin Lin: Frontend

Alice Ni: Backend

Joseph Yusuf: Backend

Hilary Zen: Project Manager

Component Map



Database Layout

Users

user_id INTEGER	username TEXT	password TEXT	date_created DATE	stories_edited TEXT
--------------------	------------------	------------------	----------------------	------------------------

User_id: identifying number for each user, assigned automatically

Username: the displayed name for each account, entered by the user

Password: chosen by the user and stored as a string

Date_created: stores when the account was started, added to database automatically

Stories_edited: a string that stores the story_id for all the stories that the user has edited

Stories

story_id INTEGER	title TEXT	full_text TEXT	last_edit TEXT
---------------------	---------------	-------------------	-------------------

Story_id: identifying number for each story, assigned in the order they are created

Title: text identifying the title of each story

Full_text: text that stores the entire story

Last_edit: text that stores the last edit to a story made by a user

Tasks

- Create Flask app and database (Back) (Joseph)
 - Write a script to create a story and add to one (Back) (Joseph)
 - Make login / registration system (Back) (Alice)
 - Allow logged in users to add/modify stories (Back) (Joseph)
 - Prevent users from contributing twice to the same story (Back) (Alice)
 - Make sure users only see the last update when modifying a story (Back) (Alice)
-
- Create templates to display stories (Front) (Hilary)
 - Make a homepage where all the stories can be seen (Front) (Devin)
 - Create a registration form / login form (Front) (Hilary)
 - Create a add new / modify existing form (Front) (Devin)
 - Create homepage where logged in users can read stories contributed to (Front) (Devin)

Front-end

- Landing.html
 - Checks if user is logged in or not and displays actions appropriately
 - Logged in users will be able to see “edit” and “add” to story buttons and the stories they have contributed to in full
 - Users that are not logged in are shown the titles of all of the stories and the option to log in: they are unable to edit or create a story
 - Once a logged-in user creates or edits a story, they will not be shown the buttons to edit said story again therefore rendering them unable to contribute to the same story twice
- editStory.html
 - A form page where the logged in user is shown the last contribution to the story and a text box to add a specified maximum number of characters to the story
 - Once submitted reroutes user back to landing
- createStory.html
 - A form page where the logged in user is able to write a story and set a title
 - Once submitted reroutes user back to landing

Back-end

- app.py
 - Login System
 - Registration System
 - Creates the users and stories databases
 - Routes for all pages and functions
 - “/”
 - Render “welcome.html” if user *is not* logged in
 - Render “homepage.html” if user *is* logged in
 - “/login”
 - If user is NOT LOGGED IN, redirect to “/”
 - Render “login.html”
 - login(reg args), and redirect to “/”
 - “/register”
 - If user is NOT LOGGED IN, redirect to “/”
 - Render “register.html”
 - register(reg args), and redirect to “/”
 - “/create”
 - If user is NOT LOGGED IN, redirect to “/”
 - Render “create.html”

- create(reg args), and redirect to “/”
- “/modify”
 - If user is NOT LOGGED IN, redirect to “/”
 - Render “modify.html”
 - modify(reg args), and redirect to “/”
- “/dash”
 - If user is NOT LOGGED IN, redirect to “/”
 - Render “dash.html”
- “/logout”
 - If user is NOT LOGGED IN, redirect to “/”
 - Pop user from session, redirect to “/”
- create() function
 - @param: user id
 - @param: data to add to database
- modify() function
 - @param: story id
 - @param: user id
 - @param: data to add to database
- login() function
 - @param: username provided by user
 - @param: password provided by user
- register() function
 - @param: username provided by user
 - @param: password provided by user

Site Map

