

## 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

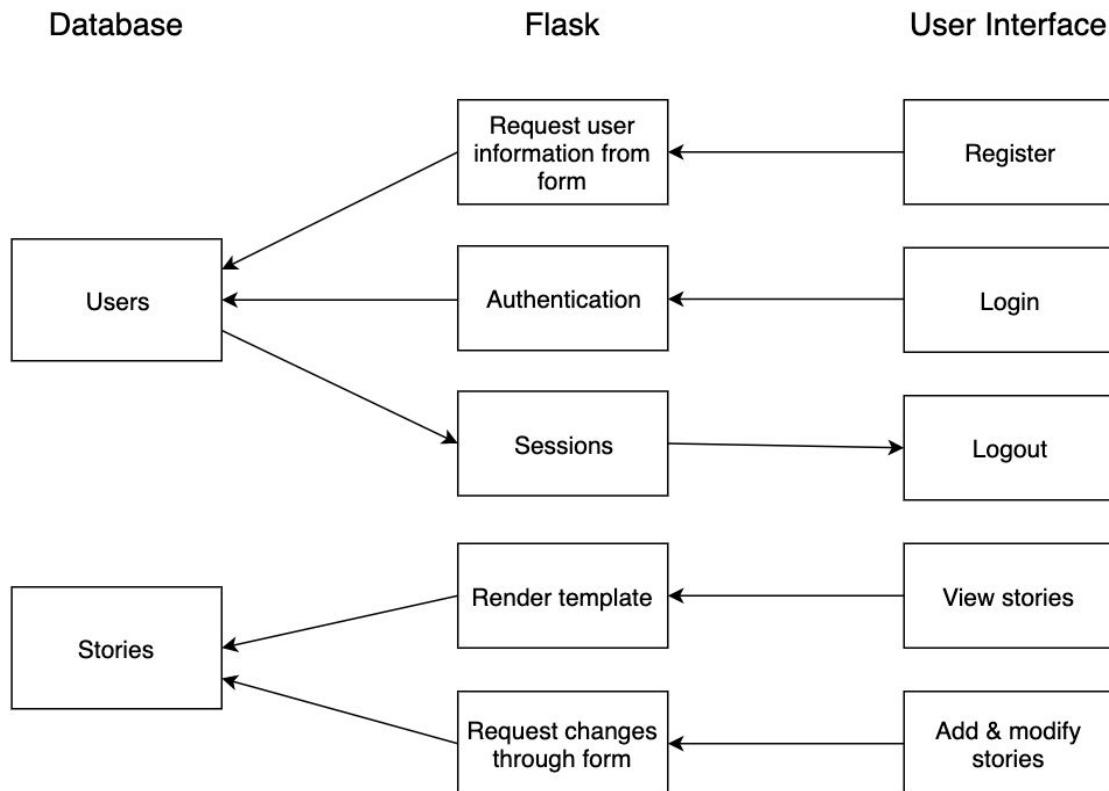
Devin Lin: Frontend

Alice Ni: Backend

Joseph Yusuf: Backend

Hilary Zen: Project Manager + Frontend

### Component Map



## Database Layout

### Users

user_id INTEGER	username TEXT	password TEXT	date_created DATE
1	"kingthomas13"	"Lebron23"	2019-10-21
2	"hotchocolate"	"12345"	2019-10-22

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

story_id INTEGER	user_id INTEGER	title TEXT	full_text TEXT	edit TEXT
1	1	"The Marshmallow"	"One day the marshmallow went for a walk. He fell into a cup of hot cacao."	"One day the marshmallow went for a walk. He fell into a cup of hot cacao."
1	2	"The Marshmallow"	"One day the marshmallow went for a walk. He fell into a cup of hot cacao. It was hot."	"It was hot."

Story\_id: identifying number for each story, assigned in the order they are created

User\_id: taken from the users table, identifies the author of the change

Title: text identifying the title of each story

Full\_text: text that stores the entire story

Edit: text that stores a user's edit to a story

## 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
    - Check user credentials against the user table entries in the database
    - If credentials match a row in the DB, add a user to the Sessions dictionary.
  - Registration System
    - Query DB, add a row containing user credentials to the user table.

- Redirect back to login page.
- 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

