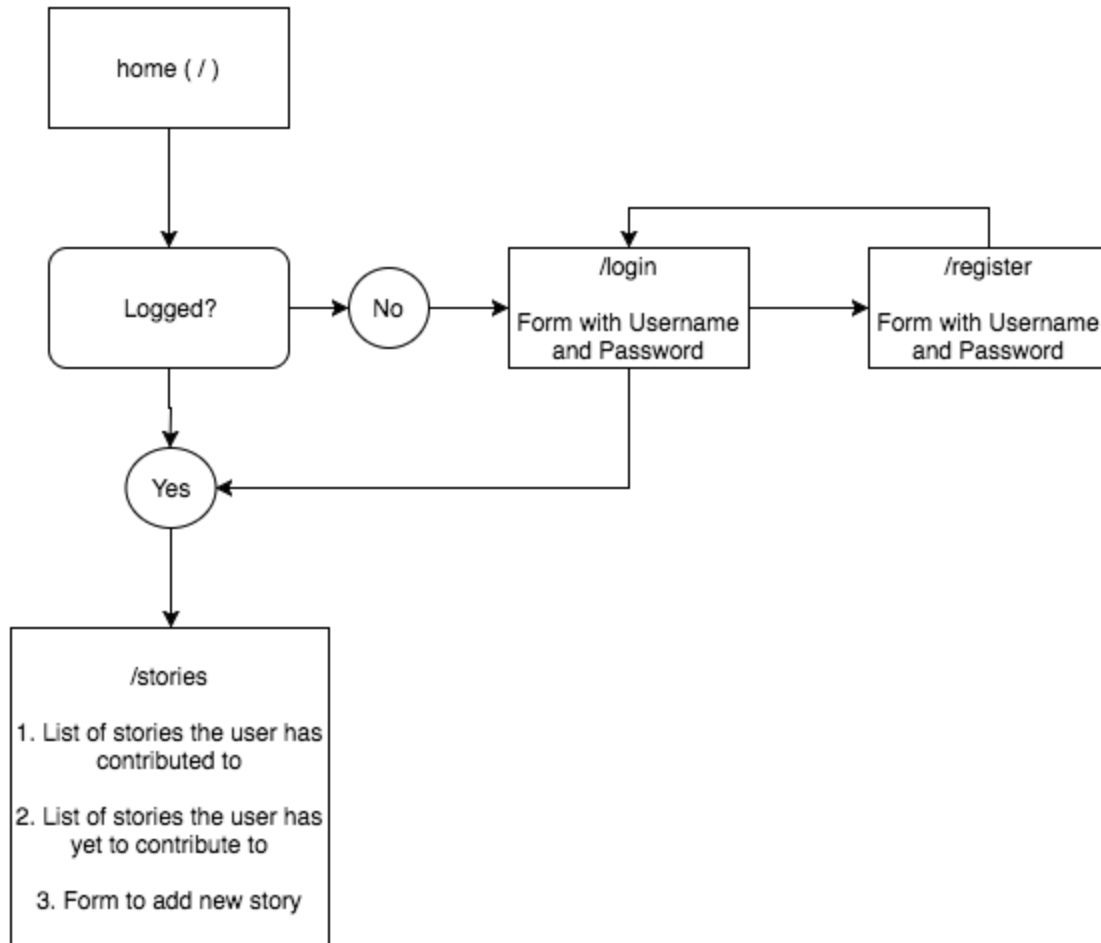


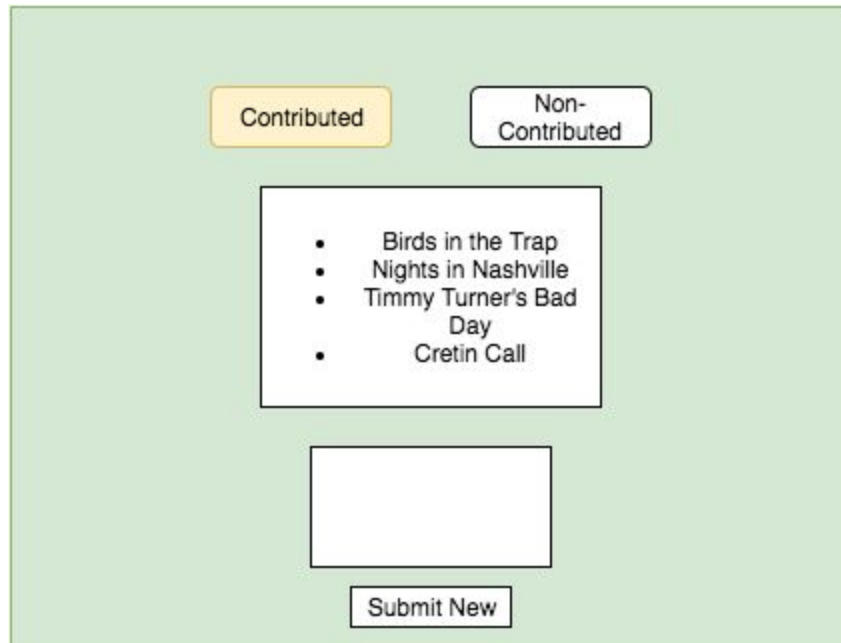
MeisterMeisters' Storytelling' Design

Vincent Liok
Jason Mohabir
Giovanni Topalini
Henry Zhang

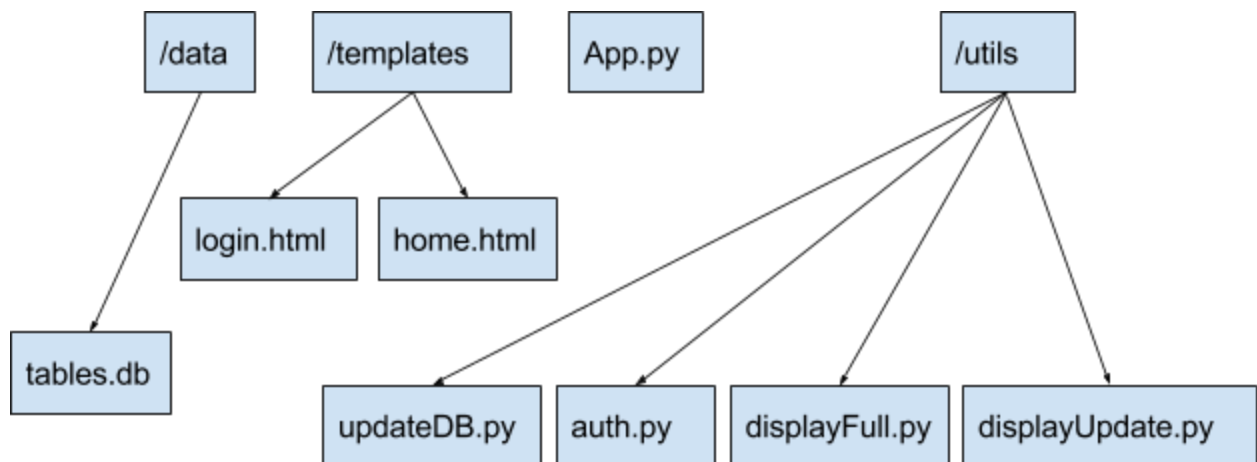
Site Map



Example /stories



Component Map



Database Structure & Samples

'Users' Table

UserID	Username	Password
1	Henry	8743b52063

'Updates' Table

Primary key

Update	UserID	StoryID	Update Number
It was a dark and stormy night...	1	1	1
in the city of Gary, Indiana	1	1	2

'Stories' Table

StoryID	Content	Title	Last Update
1	It was a dark and stormy night... in the city of Gary, Indiana	The Red Awning	2

A. Core Functionalities

- Login / Register
- Display stories contributed and stories not contributed to
- Add New Story form
- Read full story / add to Story

B. Site Map and Database Interaction

- Three tables:
- userPass: user / pass / userid
- contributors: title / storyid / userid
- stories: storyid / full story / last update

C. Task Delegation

- Authentication | Python / Flask - Gio
- Database management | Python / SQL - Henry / Vincent
- Front-end redirection | HTML, Jinja - Vincent / Jason
- Middleman | Python, HTML, SQL - Jason

D. Additional Features

- profile
- see list of contributions of other users

Component Map:

- Utils: auth, displayFull, displayUpdate, updateDB
- App.py
- Templates: login.html, home.html, newStory.html, read.html, add.html
- Data: tables.db

Auth:

- Get username and password in query
- If logging in:
 - Checks if user in username in user/pass table:
 - If yes:
 - Compares hashed pass to pass associated w/ user in table
 - If pass is good it puts username in session, redirects to home page, if bad pass, go back to login page w/ error message
 - If no:
 - Go back to login page w/ error message
- If registering:
 - Checks to see if username in user/pass table:
 - If yes, goes back to login with an error message
 - If no, checks compatibility of password
 - If good then adds user, hashed pass to table, w/ rank 2 (normal user), redirects to login
 - If bad, goes back to login page with error message

Database management:

- Store/update data after user submission:
 - After creating new story:

- Insert into "Updates" table and "Stories" table
- After updating old story:
 - Insert into "Updates" table and update "Stories" table
- After making an account:
 - Insert into "Users" table
- Transfer data into templates:
 - Stories that you have not contributed to:
title, username, update
 - Contributed stories: title, username, content

Front-end redirection:

- If user successfully adds to an existing story:
 - Redirect to the full story with success message displayed
- If user successfully creates a new story:
 - Redirect to "list of stories" page with the new title displayed