

Tell Me A Story - The Story Writing Telephone Game

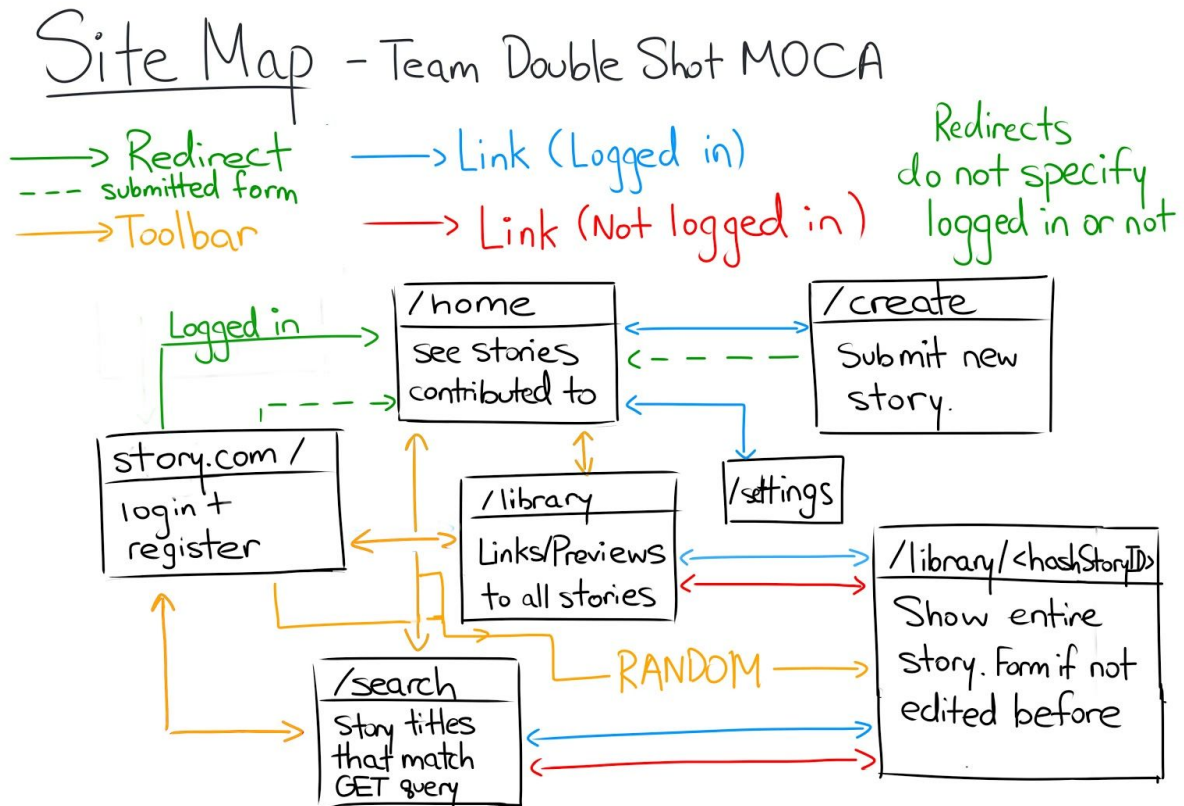
Team Double-Shot MOCA: Constantine, Maddie, Jessica, Joel

Components and How They Work (Description)

- *Homepage*
Redirects to /login if not logged in, redirects to /home if logged in.
- *Homepage/Registration: story.com/*
 - HTML Form
 - Register
 - References validity using AccountInfo Table
 - Writes to AccountInfo Table
 - Redirects back to /login with success message (in template, see if message exists)
 - Login
 - References AccountInfo, Logs in
 - Redirects to /home
- *Homepage generation*
story.com/home
Redirects to / if not logged in.
Extract stories from storyIDs (userTable), for each storyID reference mostRecentEditID from Story Table. Reference and pull edit message with ID = mostRecentEditID from Edit Table.
Generate last edit per story contributed to (preview) + link to full story.
- *Create new story: story.com/create*
 - HTML Form. Do Not allow same person to make same story title.
 - Title, First message
 - Writes to Story Table and Edit Table
- *Editing story: story.com/library/<hashStoryID>*
 - Edit story form at top if logged in AND storyID not in user's storyIDs
 - Blocked if post is in storyIDs list of user
 - Writes to Edit Table and increments Story Table
- *Search bar/Library/Toolbar*
 - At the top of all pages
 - Directs to story.com/search?Query =<searchMsg>
 - Library prints "most recent update + link to full story" using written function
 - See *search.py* under Program Components

- User Settings: story.com/settings
 - Change password - from UserAccountInfo page

Site Map



story.com/
 Standard
 Redirects to homepage if logged in

story.com/home
 displays stories (most recent edit)

story.com/create
 add new story

story.com/library
 /url
 list all stories

story.com/library/<storyHash>
 Individual story page

story.com/search

Useful Link:

https://www.tutorialspoint.com/sqlite/sqlite_update_query.htm

Database Schema

Table for all Stories

Title, StoryID, mostRecentEditID

- This table is used to keep track of all stories (but not the whole content of them). The title is the title of the story given by the creator of the story, the StoryID represents the story by a given number (in order of creation), and the mostRecentEditID is the number given by the most up to date edit for that story.

Table for all Edits

EditID # (Not unique across stories), Timestamp, StoryID, UserID, EditContent

- This table is used to keep track of all edits made to every story. The EditID is the edit number to a story. For example, if an edit is the 4th edit to a story, then its EditID is 4. The timestamp is the time the edit was made. The StoryID is the story in which the edit was made in. Two edits can have the same EditID as long as they have distinct StoryIDs. The UserID keeps track of what user made the edit, and the EditContent is the story string itself.

Table for people

UserID, StoryIDs (list)

- This table is used to identify the user logged in and display the correct story previews in his or her homepage. UserID matches with the AccountInfo UserID. StoryIDs is a list of all of the stories (list of StoryID's) that the user has contributed to. We can then use this list to refer to the stories in the Stories table.

Table for Account Info

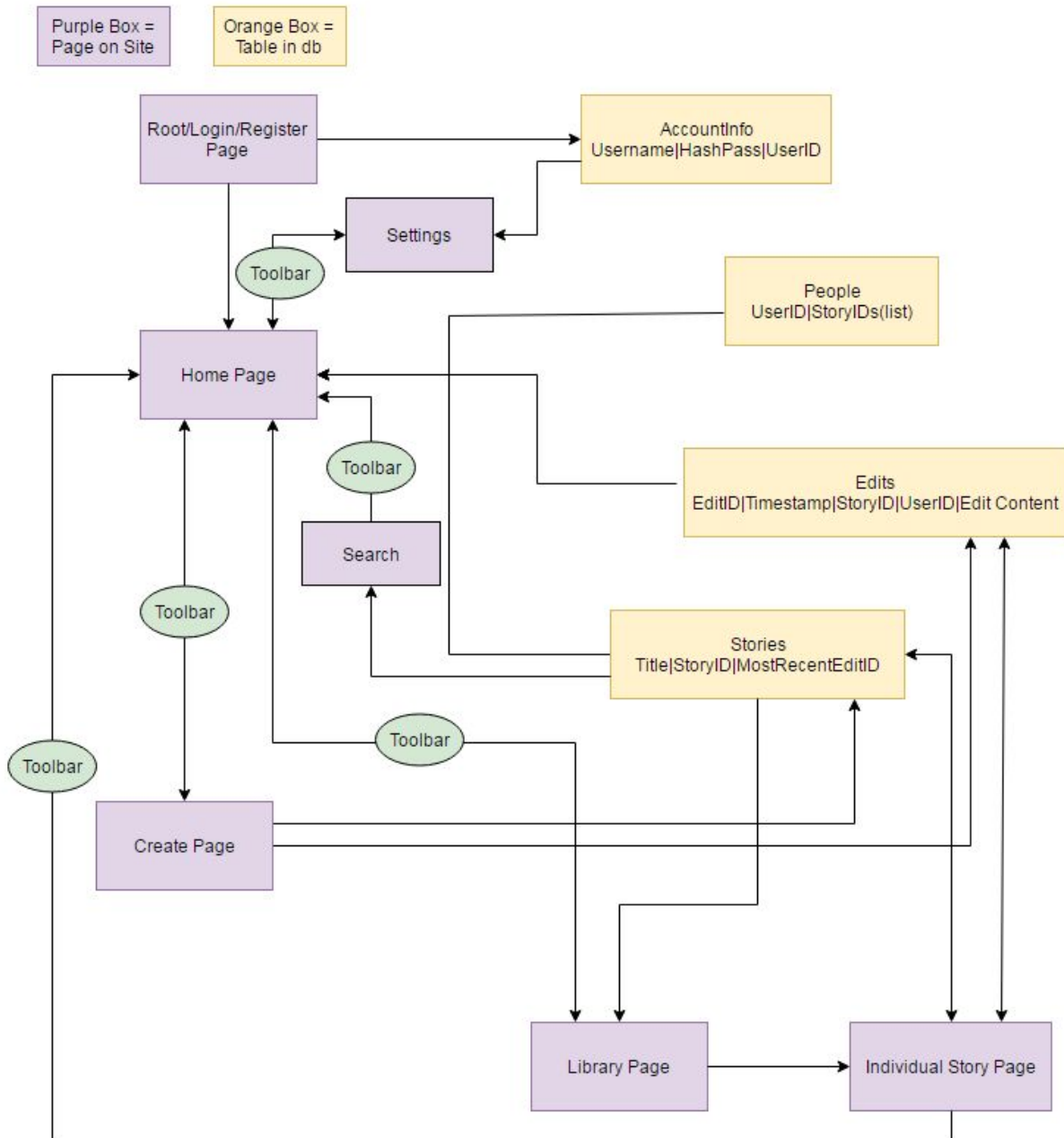
Username, HashedPass, UserID

- This is a separate table just for the login/registration process in order to be more organized. The username is the string given by the user, the HashedPass is the hash given to the password string inputted by the user, and UserID is the ID number given to each user in order to not have to refer to them by their username. The number is assigned by the order that he or she registered in. Ex: 5th registered user has an UserID of 5

Program Components (Files)

- *app.py*
 - Information from forms are put into their respective tables in the database, information from tables that are selected using helper functions in utils are written into html templates
- *data/*
 - *story.db*
 - Refer to Database Schema
- *templates/*
 - *login.html*
 - Page with form that has a username and a password field with a login and a register button
 - *homepage.html*
 - Page that prints previews of all the stories the user has contributed to
 - *storytmpl.html*
 - Template for `story.com/library/<storyHash>`, which prints the entire story
 - *library.html*
 - Page that prints a list of all the stories
 - *settings.html*
 - Page with a form to change password
- *utils/*
 - *auth.py*
 - Requests information from form, checks if username matches the hashed password from the People Table, adds key to/removes key from the session (cookie), redirects user if successful login
 - *create.py*
 - Checks if title has been used before by the user, adds new entry to Story Table, Edit Table, and appends to StoryIDs list in People Table
 - *edit.py*
 - If storyID not in StoryIDs list (People Table), render template with a form for an edit, add new entry to Edit Table, append storyID to StoryIDs list (People Table), increment mostRecentEditID (Story Table)
 - *search.py*
 - If query string in title (Story Table), get the storyID and mostRecentEditID associated with it (Story Table), print Edit Content (Edit Table)
 - *settings.py*
 - If hashed input password matches password in People Table, overwrite password for the user with the hashed new password

Component Map



Work Distribution

- Constantine - Project Manager
 - Database management and linkage to backend
- Jessica
 - Template and maybe CSS management.
- Maddie
 - Story pages, creating stories/editing stories/ random
- Joel
 - Main page generation, toolbar management.
 - Library