

Team Name: git_kermmmit (Aleksandra Koroza (PM), Maryann Foley, Mohtasim Howlader, Robin Han)
SoftDev1 pd8
P #00: Da Art of Storytelling'
2018-10-27

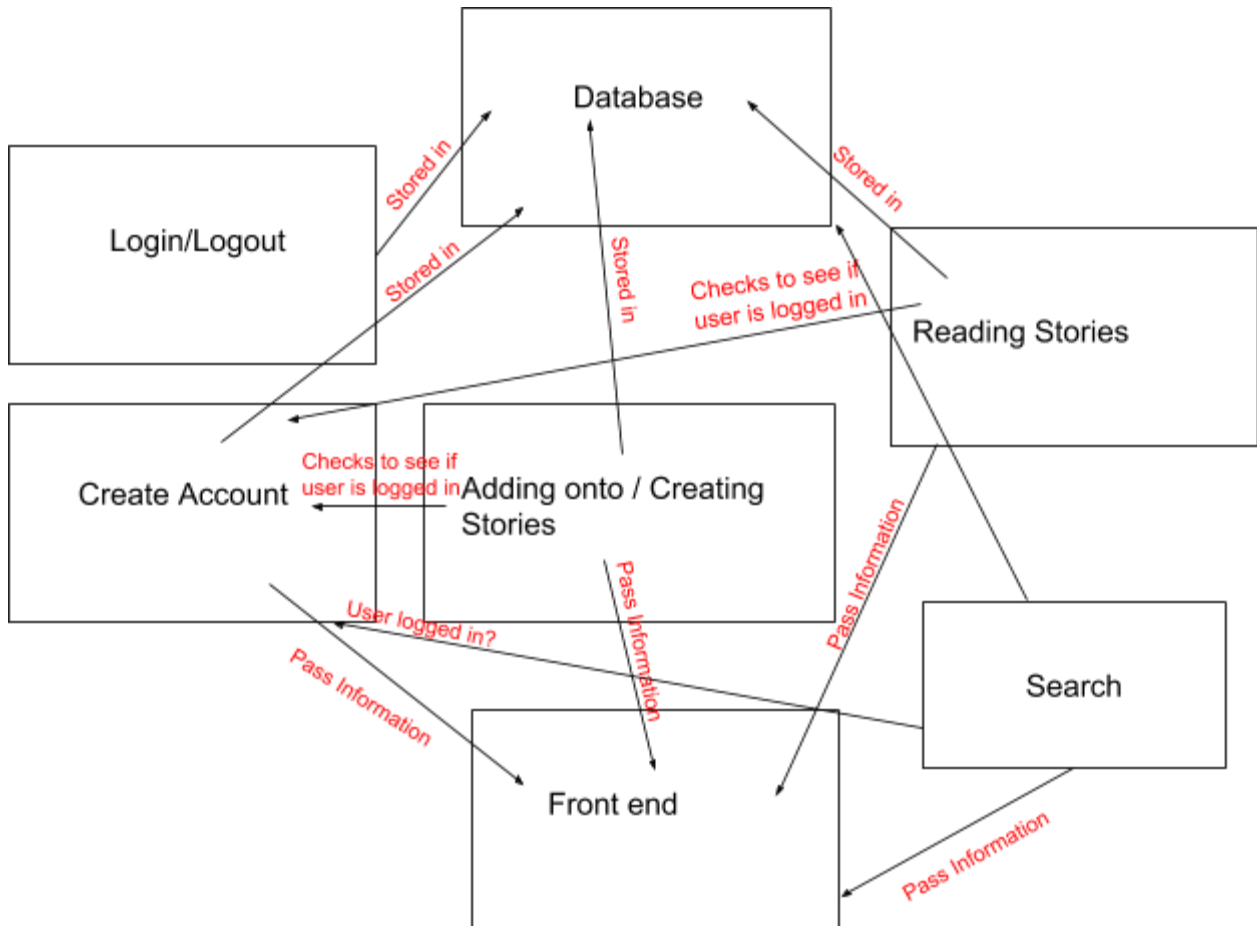
Program Components:

- Database
- Login/Logout/Create Account
- Adding to Stories
- Creating New Stories
- Reading Stories
- Search Bar
- Front end

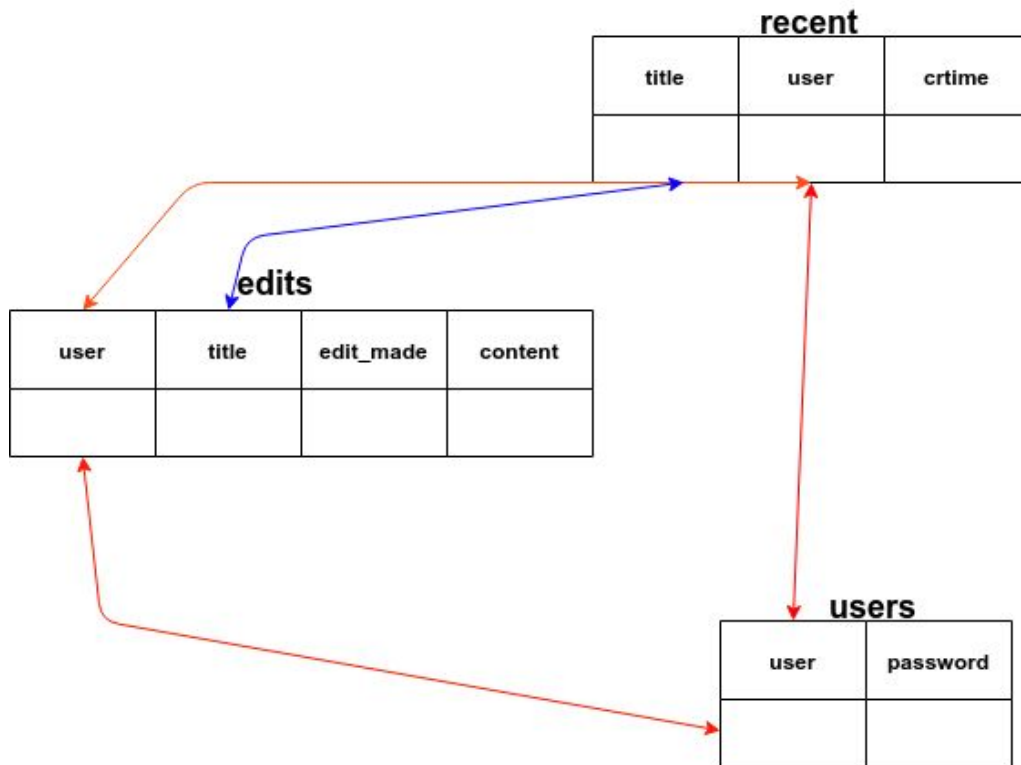
Explanation of how each component relates to the others.

1. The Database holds all the information of the storytelling game. It stores all the created accounts, and the stored, edited and added pages made by the users.
2. The Login/Logout relates to the database to search for the already created user and the Adding/Creating Stories component because only logged in members can make changes to the data stored in the Database.
3. The Create Account is also related to the Database, as every new account created is stored there which would be later retrieved and reviewed with the Login/Logout.
4. Creating a new story would create a new entry that is stored the Database.
5. Adding to Stories looks for the existing story in the Database and adds the lines the user inputs to the most recent version of the story.
6. When reading stories or adding to a story, the site checks whether the user is logged in or not.
7. The Front end relates to every other component because the user could have the option to login/logout/create account and search keywords using the site. For the users that are logged in, they can then create new stories or add onto existing stories in the Database. The Front end is also connected to the Database because it will need to display the latest revision to each story and each story's contents created by other users. The user does not have direct access to the database.

Component map visualizing relationships between components:



Database Schema:

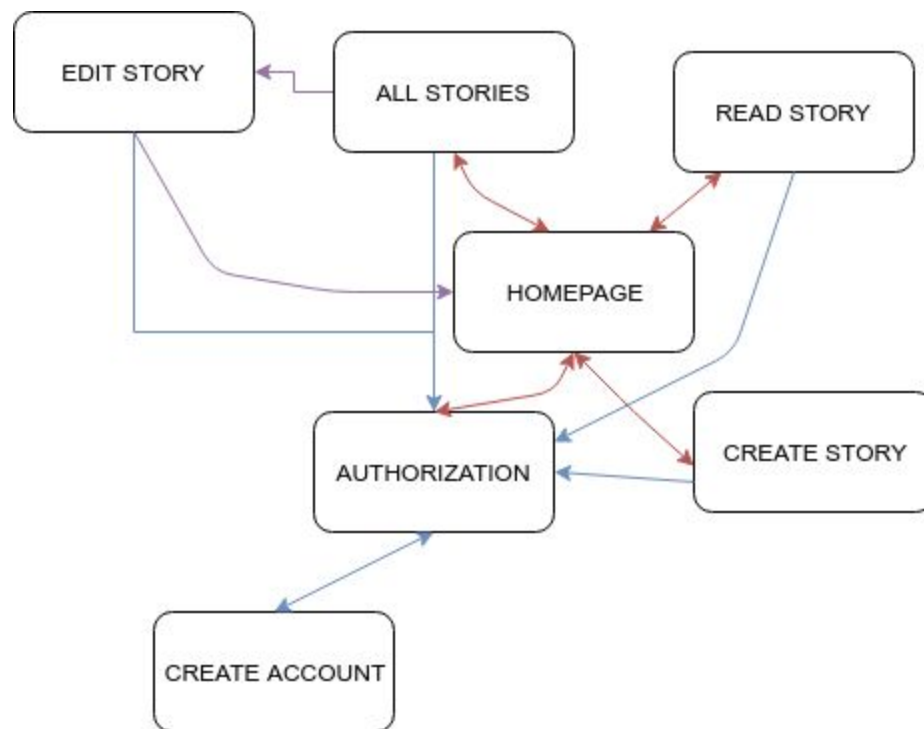


Explanation of DB schema:

- *users* is a table that contains the login information of each user. Password and username is checked when a user attempts to login. Passwords are stored in their hashed forms.
- Every time an edit is made, the *edits* table is updated.
 - No past edits are deleted
 - User field exists so we can check what stories they have contributed to and, therefore, can be viewed
 - There is a separate field for edit made and all content so far
 - Edit made is exclusively what was appended to the story
 - Edit exists for when a new person wants to edit so they can just see the last edit
 - Content is all of the story (all past edits and edit made)
 - Content exists so that we don't have to look at every last edit in order to display the current state of the story
- *recent* contains unique titles, associated user, and the update time.

- *recent* holds only one entry per title
 - When a new story is made, there is a new line that holds the title, user, and the time at creation (crtime).
 - When an existing story is edited, the old line of that story is deleted and a new one with the user who just edited is added. The timestamp section is also updated.
- Stories are identified by their titles.
- Editor is stored because this is how we reference the most recent edit. Content is not stored in this table.
 - View *edits* table for way in which the content is managed.

Site Map:



Clarification of Site Map:

1. If the user is not logged in, then they can only access the Authorization page and the Create Account page. If the user is logged in, they can access Homepage and all of its

successive pages labelled in the sitemap, along with being able to logout. The Authorization page is the first page with which the user interacts.

2. Homepage allows logged in user to a) create a new story (Create Story page), b) add to existing stories (All Stories and Edit Story), and c) click on links that generate pages containing stories the user has already contributed to (Read Story). The user can also interact with a search bar that produces the appropriate page depending on user input.
3. If Authorization is successful, the user is led to Homepage. Else, user is redirected back to Authorization and an error message is flashed. Creating a new account leads the user back to the Authorization page.
4. The All Stories page displays all of the stories the user has not yet contributed to. Upon clicking on a story title link, the user is sent to the Edit Story page in which they can only see the latest update to the story. The user's input is then incorporated into the story and that particular title is removed from the All Stories page for that user. User is redirected to Homepage upon submission of their edit.
5. Homepage also allows user to go to the Create Story page in which the user can input a title and a body of text for a new story. User is redirected to Homepage upon submission of their new story.
6. Homepage also allows user to go to the Read Story page of a particular story they have contributed to. From the Read Story page they can return to the Homepage.
7. All of the arrows pointing back to Authorization represents logout. The user can logout from all pages except for Authorization and Create Account.

A breakdown of the different tasks required to complete this project:

- Creation of database to house three tables (edits, recent, users)
- Creation of a basic template page and specific pages such as All Stories, Create Story, Home Page, Authorization, Create Account, and Read Story.
- Creation of flask app that serves pages and facilitates the creation of new ones. Tasks include:
 - 1) checking within database for user authorization
 - 2) adding new usernames and passwords to database
 - 3) updating All Stories, Homepage, and Read Story components based on user input and all associated database changes
 - 4) logging out and redirecting to the Login page
 - 5) processing user input when creating a new story and updating database accordingly
 - 6) processing user input when submitting an edit to an existing story and making sure that user can only see the most recent change
 - 7) creating a search bar that lets the user access a specific story

Break Down of Roles:

Aleksandra Koroza: (PM) Monitored devlog and updated design doc as necessary. Completed initial database work, worked on Authentication and Create Story functionality, added timestamp feature and hashed passwords.

Maryann Foley: Completed Edit Story, All Stories, and Read Story functionality. Formatted pages to facilitate editing and reading. Migrated db commands to separate file in util and created logo.

Robin Han: Worked on Create Account and Authentication functionality. Also created search bar component on Homepage. Fixed errors for other team members.

Mohtasim Howlader: Setup all templates and agreed upon pages with basic flask functionality. Worked on Edit and Authorization components.

Timeline:

10/16 Planned design of site and assigned roles

10/17 Set up of basic pages and templates and flask app.

10/18 Set up database and work on Authentication/ Create Account

10/19 Continued work on Authentication

10/20 Create Story component completed

10/21 Edit Story, Read Story, and All Stories components completed. Edited Create Account component.

10/23 Search bar component completed, added time stamp feature, and hashed passwords.

10/24 Front end (aesthetic) work

10/25 Database commands migrated to separate location, code optimized, all links turned to buttons, creation of logo.

10/26 Comments reviewed, cleaned up base template for logged in pages, bug fixes.

10/29 Demo!!