

Team Name: git\_kermmmit (Aleksandra Koroza, Maryann Foley, Mohtasim Howlader, Robin Han)

SoftDev1 pd8

**P #00:** Da Art of Storytelling'

2018-10-15

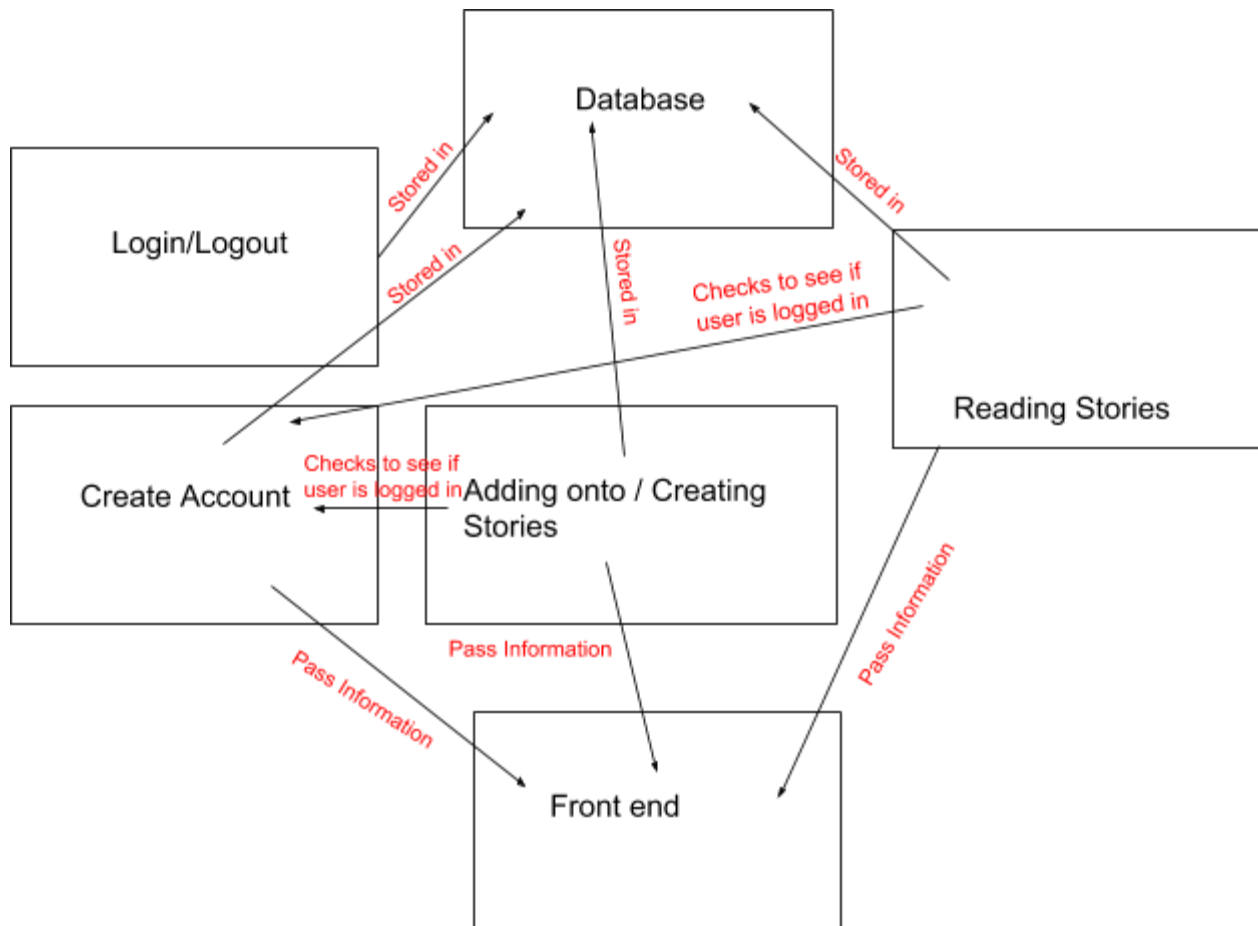
### **Program Components:**

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

### **Explanation of how each component relates to the others.**

1. The Database holds all the information of the storytelling game. It will store all the created accounts, and the stored, edited and added pages made by the users.
2. The Login/Logout related 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 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, it needs to be checked that the user is logged in or not.
7. The Front end would relate to every other component because the user could have the option to login/logout/create account and search keywords. 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, but the user will not have direct access to the database.

**Component map visualizing relationships between components.**



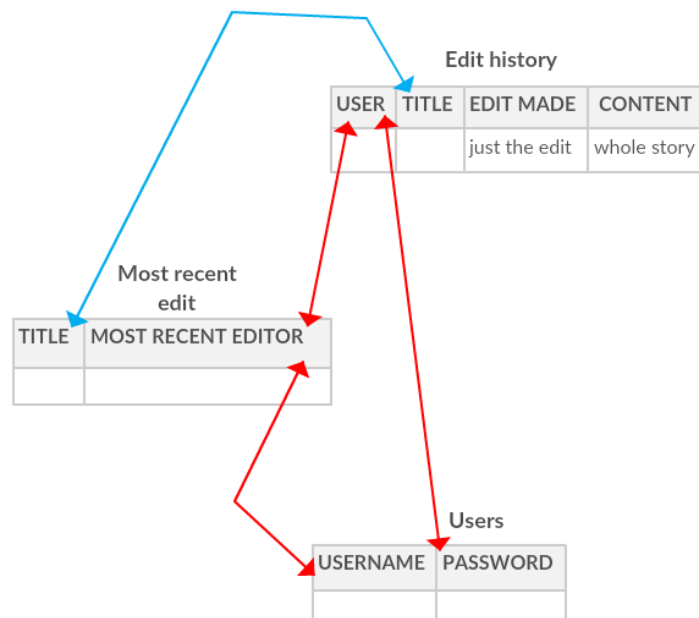
### Database Schema:

Names of tables

“Edit history”= edits

“Most recent edit”= recent

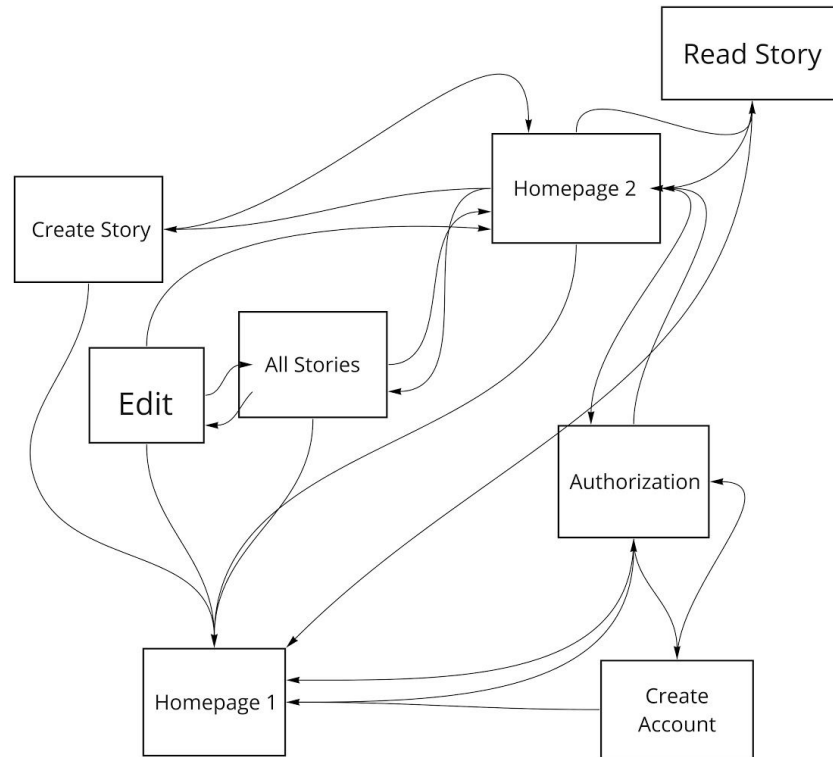
“Users”= users



Explanation of DB schema:

- Users is a DB that contains the login information of each user, password and username checked when a user attempts to login
- Edit history is appended to every time an edit is made
  - No past edits are deleted
  - User field exists so we can check what stories have been contributed to and therefore can be viewed
  - There is a separate field for edit made and content
    - Edit made is exclusively what was appended to the story
      - Edit exists for when a new person wants 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 see the current state of the story
- Most recent edit
  - Only one entry per title
    - When a new story is made, there is a new line
    - 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
  - Contains only title and editor
  - Title because that is how the story is identified
  - Editor is there because this is how we reference the most recent edit
    - View edit history table for way in which the content is managed
  - Content not contained here because we want to know all past editors and we do not want to hold the content in both of these places.

## Site Map:



\*Conditional flow specified below\*

## Clarification of Site Map:

1. If the user is not logged in, then they can only access Homepage 1, the Authorization page, and the Create Account page. If the user is logged in, they can access Homepage 2 and all of its successive pages labelled in the sitemap, along with being able to logout.
2. We envision two home pages. One for a guest user (Homepage 1) and one for a logged in user (Homepage 2). Homepage 1 would only have access to the Authorization page. Homepage 2 would allow user to a) create a new story (Create Story page), b) add to existing stories (All Stories and Edit pages), and c) click on links that generate pages containing stories the user has already contributed to (Read Story).
3. If Authorization is successful, the user is led to Homepage 2. Else, user is redirected to Homepage 1 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 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 2 upon submission of their edit.
5. Homepage 2 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 2 upon submission of their new story.
6. Homepage 2 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 (or logout, which can be done on any page).
7. All of the arrows pointing back to Homepage1 (aside from the Authorization and Create Account page) represent logout.

**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 1, Home Page 2, 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 2, and Read Story components based on user input and all associated database changes.
  - 4) logging out and redirecting to Homepage 1.
  - 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 they only see most recent change

**Break Down:**

Aleksandra Koroza: Set up templates and all basic pages. Complete New Story functionality.

Maryann Foley: (PM) Write editing and selecting story from All Pages component

Robin Han: Complete Read Story component.

Mohtasim Howlader: Set up database. Facilitate logging in, logging out, creating account, and differentiating between logged in user experience and that of a guest user.

**Timeline:**

10/16 Set up pages and templates. Whip up database file. Set up flask app to handle authorization and creating a new account.

10/17 Complete All Stories and Editing functionality

10/18 Complete Read Stories and Create Story functionality.

10/19 Test and present project!