Team ROCZ: Adeebur **R**ahman, Levi **O**levsky, Cynthia **C**heng, & Jake **Z**aia

Databases

Table	**Values**
Users	(username PRIMARY KEY String, hashedPass String, editedStories String)
Stories	(id PRIMARY KEY Integer, content String, revision String)

- **Users:**
  - Username: A name selected to represent a user (no overlaps)
  - hashedPass: The user's password (which has already been hashed)
  - editedStories: A string containing integers separated by commas corresponding the the ids of the stories they have edited
- **Stories:**
  - id: A number assigned to a given story (starts from 0 and counts up)
  - content: The plaintext content of the story, including the last revision.
  - revision: The most recent addition to the story (will be limited to a short number of characters)

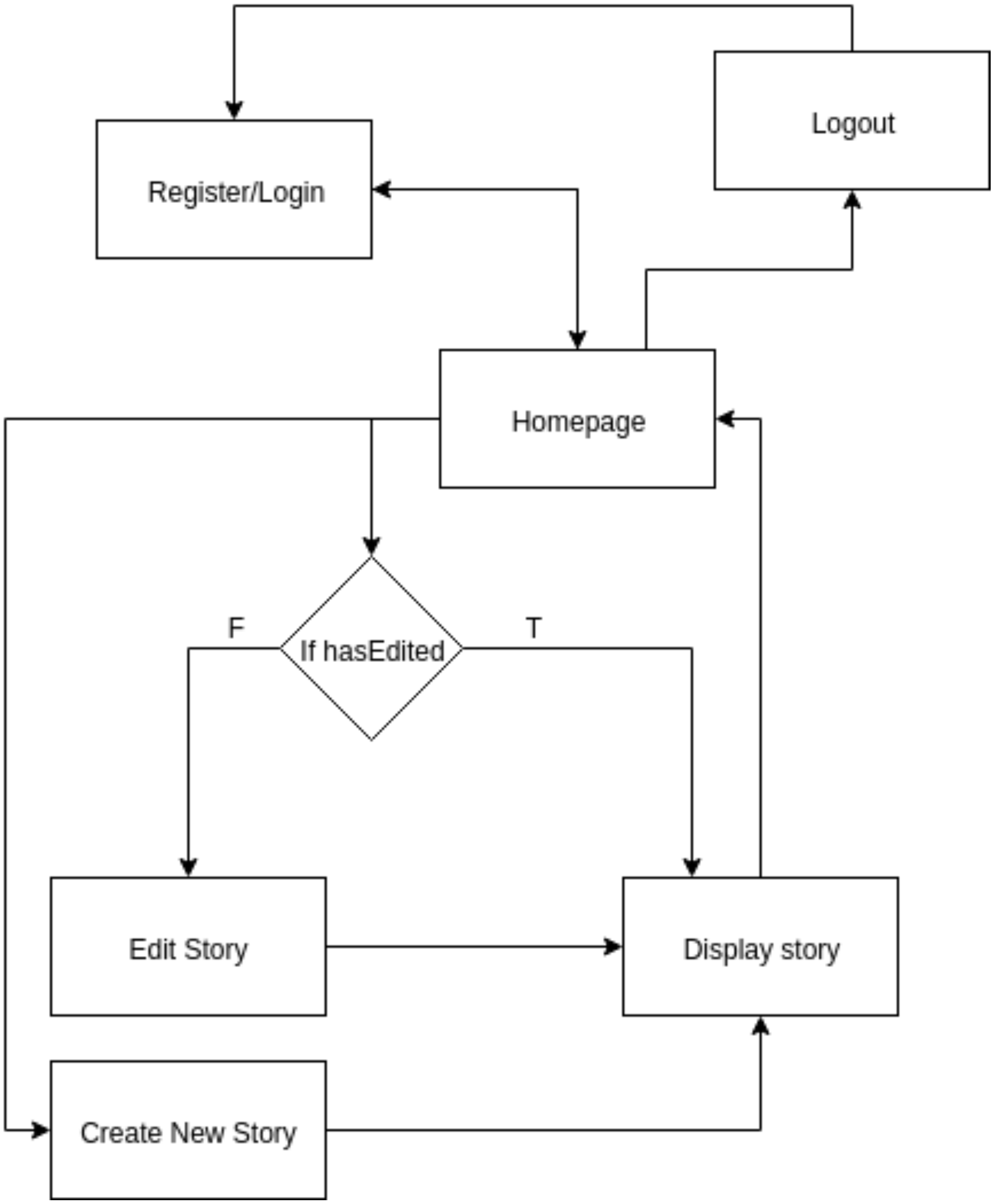
Components

- **Home Page**: Forces the user to log in or register if not already logged in. Contains a list of all stories, sorted by the edited status and then by recency. The user can click on these to go to the edit/display story page.
- **Register / Login**: A simple HTML form that will log the user in/ create their account and redirect to the homepage.
- **Story Creator**: An HTML form to start a new story, which is then added onto the homepage. The user will add the title and first few lines. Once they create the story they will not be able to edit it, however can view it.
- **Story Editor**: If the user has not edited the story, they are shown the last lines of the story in process and can add their own excerpt.
- **Story Display:** If the user has edited the story, they are shown the most updated version of the story.
- **Log Out:** User is logged out and taken back to the Register/ Login page.

Component Map

- **Home Page**: Will occupy the root route, should check session for cookies to see if the user is logged in and redirect if not. Otherwise it will pull data from the database and display a list of all stories that can be edited and a list of all stories that have been edited by the user and can now be viewed in whole.
- **Register**: Uses an HTML form to get data to create an account. Will add this data to the database as well as creating a cookie for the user.
- **Login**: Uses HTML form to get data. Will check this data against the database to determine if the login was a success.
- **Story Creator**: Uses HTML form to get data from user. A new story is added to the database, with a content string, and ID. The user's editedStories is also updated to include the story's ID.
- **Story Editor**: Uses HTML form to get data from user. Using the story's ID, it is found in the database where the story's revision and content is updated. The user's editedStories is also updated to include the story's ID.
- **Story Display:** Depending on the story's ID, the page will pull information from the database and display the content string.
- **Log Out:** User is logged out and taken back to the Register/ Login page.

Sitemap


Logout
Register/Login
Homepage
If hasEdited
F
T
Edit Story
Display story
Create New Story

Tasks

- **PM-** Cynthia
  - Management as well as resolving miscellaneous issues.
  - Possibly some front-end work?
- **Home Page** - Jake
  - The Home Page
  - The table in sqlite that makes the home page work
- **Logins/Accounts** - Adeebur
  - Register/Login Functionality
  - Log-Out Functionality
  - The sqlite table behind these features
- **Story Functionality** - Levi
  - Story Creation Page
  - Story Editing Page
  - Story Display Page