

Component Summary

- Login/Register: redirects to homepage or login depending on input (which submit button)
- Homepage: Redirects the user to different stories that are marked public or private
- Users can also create new stories
 - Public stories show up on the "suggested" portion of the user's homepage
- Homepage also shows the stories that the user has joined
- Both of these (suggested stories and joined stories) are divided into two sections split horizontally across the screen
 - Suggested stories only displays the story names and the latest update to that story
 - User can click a button below to join that story
 - Joined stories will feature a link where you can go read the full story created so far which includes the latest update to that story
- Buttons (located at the top of homepage):
 - Manage joined and suggested stories
 - "Create a new story" button

- Manage account settings

Templates: Loads stories based using data from a database into a template. This is loaded when users click on the related joined story link mentioned above

Suggest/Joined Story Template:

- Header as the title of the story taken from database
- Location of where stories displayed depends on whether the user has joined
- Buttons appear depending on whether the user wants to remove stories

Login Template (same as previous project)

Account Settings Template:

- Retrieves account info from database

Story Template (creation of story):

- Additional textbox for title
- Textbox for contribution

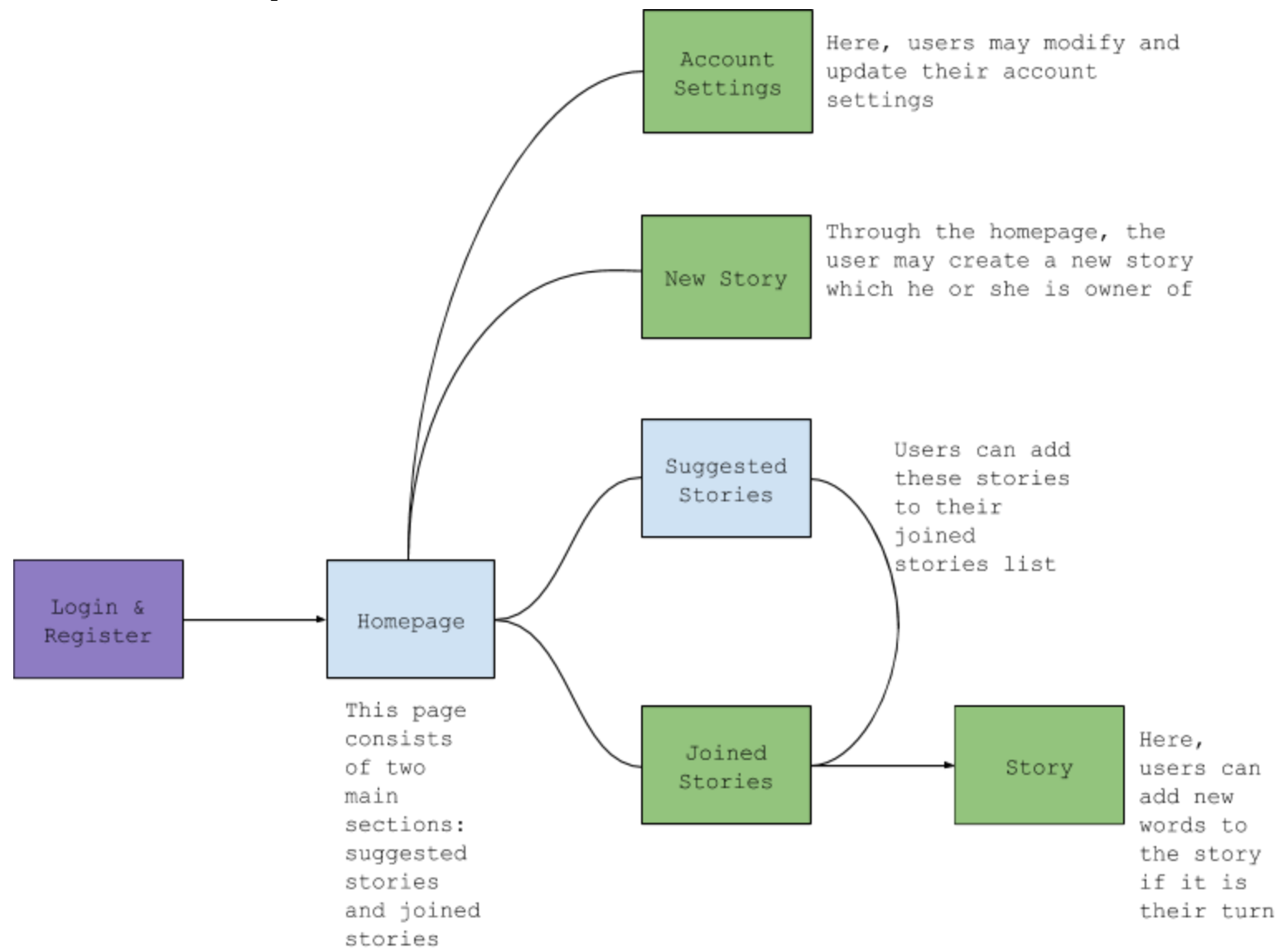
Story Template (addition to the story):

- Retrieves last contribution and whether the user has contributed yet
- Textbox for contribution
- Submit button depends on whether the user has contributed

Submission page: Users go here when they want to contribute to a story

- This page will have the latest update to all the stories currently created and a small textbox underneath where users can contribute.
- There will be a submit button underneath that will only send info if the user has not contributed to that story already

Site Map:



Database Schema:

- Contributors table
 - This would have the contributor username and ID
- Story table
 - This table would have the initial story text, the name of the story, an array of user IDs, integer to represent whose turn it is (this will revert to 0 once the user has surpassed the boundaries of the array) who have joined the story, along with the contributor ID of who created it
- Reach goals: rating for story

- Contribution table
 - Individual entry would have a contributor ID, story ID, timestamp, text contributed
 - Reach goals: rating for contribution, character limit for contributions
 - Will check before allowing entries to be made if it is contributor's turn to add text
 - Last entry should be readily accessible so that it can be quickly and easily displayed on the home page
- Comment table
 - This would have the contributor ID of the commenter, story ID the comment is on, the comment text, and timestamp
- Log-in/Registry table
 - Standard log-in that we've already experienced in doing
 - Username included
 - Hashed passwords

Optional Tasks - Ambitious Features we would like to implement but cannot guarantee

- Automatic refreshing in the story pages once in every few seconds to check whether another user has added a word to a story
- Upvotes, downvotes for stories (can be viewed in the suggest stories page)
- Private stories that owner can create
 - Other users can be added by the user by their username
 - Boolean would be added to the story database stating whether it is private or public (true for public)
- Limit on the amount of contributors in a given story
 - Set by the user (would be stored as an integer in story database)
- User receives notifications whenever they are added to another story
 - Requires another database

Project assignments:

Lorenz (Project Manager): Responsible for the functionality of stories and their contributions

Richard: Responsible for commenting functions, login/registration functionality and anything related to sessions

Will: In charge of front-end tasks such as html templates, css styling and possible javascript functionality