Agile

Theme: Get GiggleGit demo into a stable enough alpha to start onboarding some adventurous clients

Epic: Onboarding experience

User Story 1: As a vanilla git power-user that has never seen GiggleGit before, I want to be able to use GiggleGit in the same way I use git with the skills I already have to carry over.

- Task: Implement basic git support to GiggleGit
    - Ticket 1: Make GiggleGit able to take basic git commands
    - Ticket 2: Document GiggleGit and git's similarities and differences

User Story 2: As a team lead onboarding an experienced GiggleGit user, I want to be able to manage my team and their work using GiggleGit so that I can effectively see what they are doing and who can access what.

- Task: Implement the ability to grant permissions and require access
    - Ticket 1: Make a role hierarchy that limits access
    - Ticket 2: Make a UI that clearly displays who can do what and how to manage roles

User Story 3: As someone who only enjoys specific types of humor, I want to personalize my GiggleGit experience so that the memes I come across cater to what I find funny.

- Task: Add personalized content for users in GiggleGit
    - Ticket 1: Add the option to "see more like this" or "not interested" for each meme
    - Ticket 2: Gather information based on a user's demographic enjoys and recommend memes to that user

This is not a user story. Why not? What is it?

- As a user I want to be able to authenticate on a new machine

This is not a user story, it's more of a request. A user story should have insight and give background as to who the user is and it should provide an explanation as to why they want the features they wrote about.

Formal Requirements

Goal: Create a diff tool for GiggleGit that adds a playful snicker to when they sync

Non-goal: Try to make SnickerSync compatible with products outside of GiggleGit

Non-Functional Requirement 1: Access control

- Functional requirements 1: People with permission such as PMs should be able to configure the settings of the snickering concepts
- Functional requirement 2: The data inside should be secure and random people should not be able to access it without some sort of authentication'

Non-Functional Requirement 2: User Study

- Functional requirement 1: People should be able to be randomly assigned to control or test groups
- Functional requirement 2: Engagement and feedback should be tracked and collected in order to see what the differences are