

Sami Souayah

Agile:

- Theme: Get GigggleGit 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 GigggleGit before, I want to make sure the layout is user friendly and I am able to navigate through the GigggleGit app easily
- User story 2: As a team lead onboarding an experienced GigggleGit user, I want to make sure that regular users have their previous data reliably stored and accessible.
- User story 3: As an experienced GigggleGit user, I would like to be able to access my old memes, both unsent and sent ones with the unsent ones being in a draft section of some sort.

- Task: Make users' previous memes accessible

- Ticket 1: Design and create 2 separate sections in the frontend for posted and unposted memes respectively

Update the frontend to accommodate a drafts section and previously posted section that calls to a database to retrieve all previous memes

- Ticket 2: Create a database to store all users' memes

Create an appropriate database on mongoDB that stores memes by user and in two sections. Ensure the database is called from backend to the frontend correctly and efficiently.

3. 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
  - i. This isn't a user story because it is missing the who and the why.  
There isn't a very specific "who" other than the user referring to themselves as a "user" but nothing more specific. There also isn't a why/rationale behind the use/need for the desired feature.

#### Formal Requirements:

1. Goal: Create a merging tool that utilizes GigggleGit packages guided by user studies.
  - a. Non-goal: Utilize user studies to understand what functions are necessary
2. Non-functional requirement 1: Testability
  - a. Functional requirements:
    - i. Ensure that all merging behaviors can be easily identified for users that are testing when they are completed.
    - ii. Ensure that user testing errors, inputs, and preferences are able to be tracked in a secure testing environment
  - b. Non-functional requirement 2: Useability
    - i. Functional requirements:
      1. Ensure the frontend is uncrowded, vibrant/attractive for users, and each section is clearly and accurately labeled for users
      2. Ensure that the process of merging doesn't require many extra and complicated steps.