

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 similar Git commands and not need to relearn too much about it.

Task: Implement standard git command support

- Ticket 1: Make sure GiggleGit is able to support basic git commands
 - Implement standard git commands by researching and implementing commands from <https://www.atlassian.com/git/glossary#commands>
- Ticket 2: Create documentation for functionality
 - Create documentation for what each command does, highlighting the difference between GiggleGit commands and standard git commands.

User story 2: As a team lead onboarding an experienced GiggleGit user, I want to be able to manage permissions and access control so I can easily delegate roles to specific team members based on their work.

Task: Implement role based permission and access control

- Ticket 1: Design role types and permissions system
- Ticket 2: Implement a UI for easy management of permissions and access

User story 3: As a new user who has no experience with standard git commands, I want an easy onboarding guide to understand core features

Task: Create user friendly onboarding guide

- Ticket 1: Design easy to understand onboarding guide detailing key features(merges are managed by memes)
- Ticket 2: Implement a tour for first time users with option to skip

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

- It is not a user story because it's more of a technical capability of the product(feature request) rather than a user story where the user's goal/reason for wanting it is framed around the requirement.

Project requirements

Goal: Enable users to sync their changes with added element of a snicker, making it more playful

Non-Goal: Incorporate optional AI meme image generation in place of memes

Non-functional requirement 1: Access control for PM and devs to modify snicker concepts

Functional requirements:

Implement role based access control that allows certain devs and PM to modify and maintain different snicker concepts

Make sure only people with authorized access(privileges) are able to access management features to assign others roles by integrating a secure authentication system.

Non-functional requirement 2: Random user assignment for user study

Functional requirements:

Implement backend system to randomly assign user into control group or variant group when being put into the study

Log information and feedback from the users in different groups for later analysis