

CSHW2

CS411

Roman Bukhovko

Agile

- Theme: Get GiggieGit 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 GiggieGit before, I want to understand how to perform basic operations in GiggieGit with a minimal learning curve, so that I can use it without lots of training.
 - Task: Create simple user documentation for basic Git operations in GiggieGit.
 - Ticket 1: Write clear, step-by-step instructions for committing, branching, and merging in GiggieGit.
 - The documentation should include at least three basic operations with examples that relate to Git equivalents.
 - Ticket 2: Create a get-started guide integrated into the UI.
 - Add a help icon linking to the guide.
- User Story 2: As a team lead onboarding an experienced GiggieGit user, I want to be able to create roles within my team so they can have access to the correct features.
 - Task: Implement a role-based control system.
 - Ticket 1: Design and develop the role-based system.
 - Define roles for admins, contributors, etc. with varying levels of permissions.
 - Ticket 2: Add role management features to the settings page.
 - Enable team leaders to assign roles and adjust permissions for members.
- User Story 3: As a user interested in experimenting with meme merges, I want to see the merge history in a fun, visual way, so I can track how memes influenced my merge decisions.
 - Task: Develop visual merge history tracking.
 - Ticket 1: Implement a visual merge history.
 - Create a graph or log that shows past merges, with memes and their effects on the merge.
 - Ticket 2: Allow users to replay past merges.
 - Enable users to replay previous merges to see how meme-based merges impacted the final result.

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's not a user story because it lacks a specific goal that the user wants to achieve. This is more of a functional requirement.

Project requirements

- Goal: Implement SnickerSync in a way that enhances the user experience by adding jokes during the merging process.
- Non-Goal: Do not replace the whole Git workflow with SnickerSync, only enhance the merge experience with jokes.
- Non-functional requirement 1: Security
 - Functional requirement 1: Use OAuth authentication for secure access.
 - Functional requirement 2: Encrypt all data sent over the network during the syncing process.
- Non-functional requirement 2: Usability
 - Functional requirement 1: Simplify onboarding with a tutorial for new users.
 - Functional requirement 2: Build a GUI for the merging process.

Design

