Christian Gonzalez

Professor Golbus

CS411

Oct 6, 2024

Agile

1.

    a. As a vanilla git power-user that has never seen GiggleGit before, I want to easily integrate GiggleGit into my existing workflow so that I can manage my repositories with GiggleGit without disrupting my current processes.
    b. As a team lead onboarding an experienced GiggleGit user, I want to assign specific roles and permissions within GiggleGit so that each team member has appropriate access levels tailored to their responsibilities.

2.

    a. As a new user, I want to quickly set up GiggleGit on my development environment so that I can start using its features without unnecessary delays.
        i. **Task:** Set up GiggleGit installation wizard
            1. **Ticket 1:** Design Installation Wizard Interface
                a. Create a user-friendly interface for the GiggleGit installation wizard that guides users through the installation process step-by-step.
            2. **Ticket 2:** Implement Installation Wizard Functionality
                a. Develop the backend functionality for the installation wizard, including environment checks, dependency installations, and configuration settings. Ensure the wizard handles errors correctly and provides clear feedback to the user.

3. The statement provided is not a user story because it lacks the context and the desired outcome. A user story must follow a format similar to “As [a user persona], I want [to perform this action] so that [I can accomplish this goal]”. In the statement, there is no clear reason for wanting to authenticate on a new machine. This is instead more of a functional requirement or a feature rather than a user story

Formal Requirements

1. **Goal/Non-Goal**
    a. **Goal:** Create a reliable and easy-to-use SnickerSync interface that allows users to merge code with memes and supports user studies to test its effectiveness.
    b. **Non-Goal:** Develop advanced features like automatic meme generation during the merge process.
2. **Non-Functional Requirements**
    a. **Access Control**
        i. **Functional Requirements**
            1. Implement role based access control (RBAC)
            2. Create an administrative interface that allows administrators to add, remove, and modify user roles and permissions.
    b. **User Study Randomization**
        i. **Functional Requirements**
            1. Develop a module that assigns participants randomly to either the control group (using standard Git merges) or the variant group (using SnickerSync)
            2. Implement a way to log which group each user is assigned to for later analysis.