Emily Pessina
CS411 Homework 2

Agile
User Stories:

1. As a vanilla git power-user that has never seen GiggleGit before, I want to explore how GiggleGit differs from my current experiences with Git, with ease and minimal confusion in setup, so that I can use the extra features without running into issues or losing efficiency.

2. As a team lead onboarding an experienced GiggleGit user, I want to quickly set up a new repository and invite my team members to collaborate, and feel confident that they can join smoothly, merge work, and find joy in using the meme based version control system.

3. As a new user unfamiliar with version control systems, I want to have a step by step, interactive tutorial in the platform that introduces the major version control concepts and the platforms meme oriented merge workflow, so I can easily manage my code and collaborate with others.

Task: Design an interactive tutorial for new users unfamiliar with version control.

Tickets for User Story 3:

1. Ticket Title: Design interactive tutorial flow
   Details: Create a detailed flowchart for the tutorial that covers the basics of version control and introduces GiggleGit's system of a meme oriented merge feature. The flow should include user prompts, action steps, and visual aids to help guide users through their first repository setup and commit.

2. Ticket Title: Develop tutorial UI/UX mockups
   Details: Create mockups for the tutorial screens, making sure they are visually pleasing, easy to navigate, and aligned with GiggleGit's theme. Incorporate feedback from initial user tests to edit the design for clarity and effectiveness.

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 a requirement rather than a user story because it doesn't specify the value for the user or the goal for the user. User stories emphasize all aspects of what is being effected or catered to, this phrase focuses only on what is specifically being requested (it is a feature request). An actual user story would be

As a user, I want to authenticate on a new machine quickly and securely so that I can continue my work seamlessly on multiple devices.

Formal Requirements

- Goal: Ensure that users find SnickerSync enjoyable and simple to use, enhancing the overall GiggleGit experience by making merges more engaging and funny through humor.

-Non-Goal: Optimize the efficiency of the SnickerSync algorithm for large-scale enterprise repositories. This would not be a main focus for the current phase of user studies.

Non-Functional Requirements:

Non-functional Requirement: The project managers need access to manage the different concepts used in SnickerSync, making sure they can update and test new snickers without messing up the base functionality for all users.

Functional Requirement 1: Implement a user role management system where project managers have extra access to create, edit, delete, etc. snickering concepts within the tool.
Functional Requirement 2: Create an interface for project managers to preview and test snickering concepts in a controlled testing environment, where they can experiment and play around with new ideas and features before making them available to all user groups.

Non-functional Requirement 2: SnickerSync should maintain consistent performance and response times during merging, despite of the size or level complexity of the codebase being merged.

Functional Requirement 1: Implement performance monitoring tools to track the response times during different stages of the merge process, while notifying the team about any major differences from the expected performance levels.
Functional Requirement 2: Make sure that SnickerSync handles codebases of different sizes without very obvious worsener in speed or functionality. Do this by optimizing the base algorithms and resources used in the merge process.