Manny Wang

CS411 HW 2 10/8/2024

Requirements:

Agile:

1. Complete these user stories:
   - As a vanilla git power-user that has never seen GiggleGit before, I want to… have access to a manual to explain the functionality of each meme and its corresponding git command so that I can quickly become familiar with GiggleGit.

   - As a team lead onboarding an experienced GiggleGit user, I want to… be able to add the user to the team's GiggleGit repository in order for them to begin contributing

2. Create a third user story, one task for this user story, and two associated tickets.
   - Tasks should be a single phrase. (As should themes and epics. See those provided.)
   - User stories should be one to three sentences.
   - Tickets should have a title consisting of a single phrase and details that are long enough to sufficiently describe what needs to be done. You do not need to assign points to the tickets

     User story: As a repeat user, I want to be able to save my settings

     Task: Store and use user settings for repeat users

     Ticket: Design and create a settings file that can appropriately store user settings

     Create a JSON document that stores all GiggleGit appropriate user settings.

     Ticket: Implement settings file parser

     Parse JSON document so that GiggleGit properly executes commands with proper arguments and flags set in user settings.

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

     This is not a user story because there is no "benefit." The user just expresses a desire and does not provide a reason (benefit) for why this should be done. Instead it is a request for a feature.

Formal Requirements:

1. List one goal and one non-goal
   Goal: Collect accurate and meaningful data from user studies on SnickerSync
   Non-goal: The software will not alter or analyze the user data
2. Create two non-functional requirements. Here are suggestions of things to think about:
   - Who has access to what
   - PMs need to be able to maintain the different snickering concepts base
   - A user study needs to have random assignments of users between control groups and variants

     Non-functional Requirement #1: Data Validity

     Non-functional Requirement #2: Confidentiality

3. For each non-functional requirement, create two functional requirements (for a grand total of four functional requirements).

#1:

Functional Requirement: Store whether a user has taken the study and only allow them to take the study once

Functional Requirement: The system must produce random assignments of users between control groups and variants

#2:

Functional Requirement: Employees can only access the data that they have the designated clearance for

Functional Requirement: All data stored of study participants must be encrypted