

CodeChuckle is introducing a new diff tool: SnickerSync—why merge in silence when you can sync with a snicker? The PMs have a solid understanding of what it means to "sync with a snicker" and now they want to run some user studies. Your team has already created a vanilla interface capable of syncing with the base GiggieGit packages.

Complete the following tasks:

1. List one goal and one non-goal
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
 - A user study needs to have random assignments of users between control groups and variants
3. For each non-functional requirement, create two functional requirements (for a grand total of four functional requirements).

Goal:

To conduct user studies that gather feedback on the usability of the SnickerSync tool

Non-Goal:

To validate the technical performance or integration of SnickerSync with other tools outside of the user experience context.

Non-Functional Requirement 1: User Access Control

Description: The system must ensure that only the authorized users have access to the different functionalities of SnickerSync.

Functional Requirement 1.1: Role-Based Access

- Details: Implement a role-based access control system that defines user roles (e.g., admin, project manager, user) and permissions. Each role should have specific access rights to features in the SnickerSync tool, ensuring sensitive functionalities are restricted to authorized users only.

Functional Requirement 1.2: User Authentication

- Details: Develop a secure user authentication system requiring users to log in before accessing SnickerSync. This system should support features like multi-factor authentication (MFA) to restrict SnickerSync access to authorized users only.

Non-Functional Requirement 2: Random Assignment for User Studies

Description: The user study must implement random assignment of participants to control groups and variants to ensure unbiased results.

Functional Requirement 2.1: Randomization Allocation

- Details: Randomly allocate participants to different groups for user studies. One method to do this is to randomly assign each participant a unique number. Based on this number, assign each participant into either the control or variant groups.

Functional Requirement 2.2: Tracking and Reporting of Assignments

- Details: Develop a tracking system that logs each participant's assignment to a group. The system should allow administrators to generate reports on participant distributions, ensuring transparency and aiding in the analysis of user study results.