Project Requirements

- Goal: Implement a user study platform for SnickerSync to gather feedback and data, enabling PMs to analyze user interactions with the new snickering interface.
- Non-Goal: Develop new snickering concepts beyond those already defined by the PMs.
- Non-Functional Requirement 1: Access Control
    - Only authorized personnel should have the ability to manage and modify snickering concepts.
    - Functional Requirements:
        i. User Authentication:
            - The system shall require PMs to log in using secure corporate credentials before accessing the snickering concept management interface.
        ii. Role-Based Access Control:
            - The system shall implement role-based permissions, allowing only users with the 'PM' role to create, edit, or delete snickering concepts.
- Non-Functional Requirement 2: Randomized User Assignment
    - Ensure unbiased results by randomly assigning users to control and variant groups during the user study.
    - Functional Requirements:
        i. Random Assignment Algorithm:
            - The system shall automatically and randomly assign new users to either the control group (standard GiggleGit interface) or the variant group (SnickerSync interface) upon their first use.
        ii. Balanced Group Distribution:
            - The system shall monitor group sizes and adjust the randomization process to maintain an approximately equal number of users in both the control and variant groups over time.

Agile

- Theme: Get GiggleGit 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 GiggleGit before, I want an interface that feels familiar so that I can quickly adapt without a steep learning curve.
    - Task: Create a familiar user interface
    - Ticket 1: Design UI Similar to Standard Git Clients
        - We need to design UI mockups that closely resemble traditional Git interfaces to make new users feel comfortable.
    - Ticket 2: Implement the Familiar UI Design
        - Develop the frontend based on the mockups, ensuring functionality aligns with user expectations and standard Git operations.
- User Story 2:
    - As a team lead onboarding an experienced GiggleGit user, I want to integrate GiggleGit into our team's workflow easily so that collaboration is seamless from the start.
    - Task: Simplify team integration
    - Ticket 1: Develop Team Onboarding Documentation
        - Create comprehensive guides and tutorials to help teams set up GiggleGit within their existing workflows without friction.
    - Ticket 2: Implement Integration with Existing Tools
        - Ensure compatibility with popular development tools (e.g., CI/CD pipelines, code editors) and test these integrations thoroughly.
- User Story 3:
    - As a developer interested in innovative tools, I want to experience how GiggleGit manages merges with memes so that merging code becomes a more engaging and enjoyable process.
    - Task: Implement meme-based merge conflicts
    - Ticket 1: Design Meme-Based Merge Conflict Visualization
        - Design how merge conflicts are displayed using memes, selecting appropriate memes that correspond to different conflict types while keeping the interface intuitive.
    - Ticket 2: Integrate Memes into Merge Process
        - Code the functionality to display memes during merge conflicts and ensure the merge process remains stable and efficient.