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

It is not a user story because it does not specify a kind of user that wants to do this, it simply says “a user”. This is a functional requirement because it is something that would assist in some non-functional requirement like flexibility.

* 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 to quickly learn to use GiggleGit without having to relearn basic concepts.
  + Task: Create a quick start guide that highlights GiggleGit’s functionality
    - Ticket 1: Document essential GiggleGit commands for new users

Survey current users on which commands they find the most useful and describe how these add functionality to vanilla.

* + - Ticket 2: Define scenarios where GiggleGit would be useful instead of simply using vanilla git.

Take photos of people laughing when they see a meme when they merge using GiggleGit

* User story 2: As a team lead onboarding an experienced GiggleGit user, I want to seamlessly integrate them into the team without upsetting the previous workflow.
  + Task: Have a standardized onboarding process to highlight team conventions and how they use GiggleGit.
    - Ticket 1: Provide a command to display the roles and permissions of each team member

Gather data on the each member’s roles and permissions to store in a database so a new member would know who to go to for a specific issue.

* + - Ticket 2: Maintain a history of team projects, tasks, and tickets so a new member could see examples of how the team operates

Have a queue for future tickets and a store project history in a database so a new member can quickly find a task to work on and examples of how the team operates in the past.

## 2. Formal Requirements

* Goal: Run user studies which investigate the ways potential users would want to use SnickerSync and how our team can improve its functionality to optimize for certain users.
* Non-Goal: Create a more developed interface that can sync with select third party packages.
  + Non-Functional Requirement 1: Usability
    - Functional Requirements:
      * Have a concise list of use cases that every user should be able to execute in the user study.
      * Include a “--help” command that will allow users to discover how to use SnickerSync.
  + Non-Functional Requirement 2: Reliability
    - Functional Requirements:
      * Make unit tests which should be passed consistently during the user study.
      * If there is an error, there should be an error message which correctly and uniquely identifies how and why an error occurred so the user can fix their issue.