Agile:

1. Complete these user stories:

- As a vanilla git power-user that has never seen GiggleGit before, I want to be able to use GiggleGit without having to take a lot of time to learn it so it does not affect my workflow, and I can use it effectively.
- As a team lead onboarding an experienced GiggleGit user, I want to ensure our team processes and activities align with GiggleGit's features so that the new members can integrate seamlessly into our projects.

2. Create a third user story, one task for this user story, and two associated tickets.

- As a product manager of a large team, I want to know GiggleGit's suitability and scalability for large scale projects and cooperation.
- Ticket 1: Case Study for large projects team dynamics and productivity.
  Perform a case study where we test the capabilities of GiggleGit in large scale environments and projects and how it effects
- Ticket 2: Stress Test for large amounts of users
  Perform stress tests to evaluate GiggleGit's performance under heavy load and with lots of simultaneous users. Record the results and evaluate and optimize the system to ensure it remains responsive and reliable during large scale environments.

3. This is not a user story. Why not? What is it?
   a. As a user I want to be able to authenticate on a new machine
      This is not a user story because it does not really have a goal or benefit to explain why the user wants the feature, it should be in the format: "As an X I want Y so I can Z". It is closer to a functional requirement.

Formal Requirements:

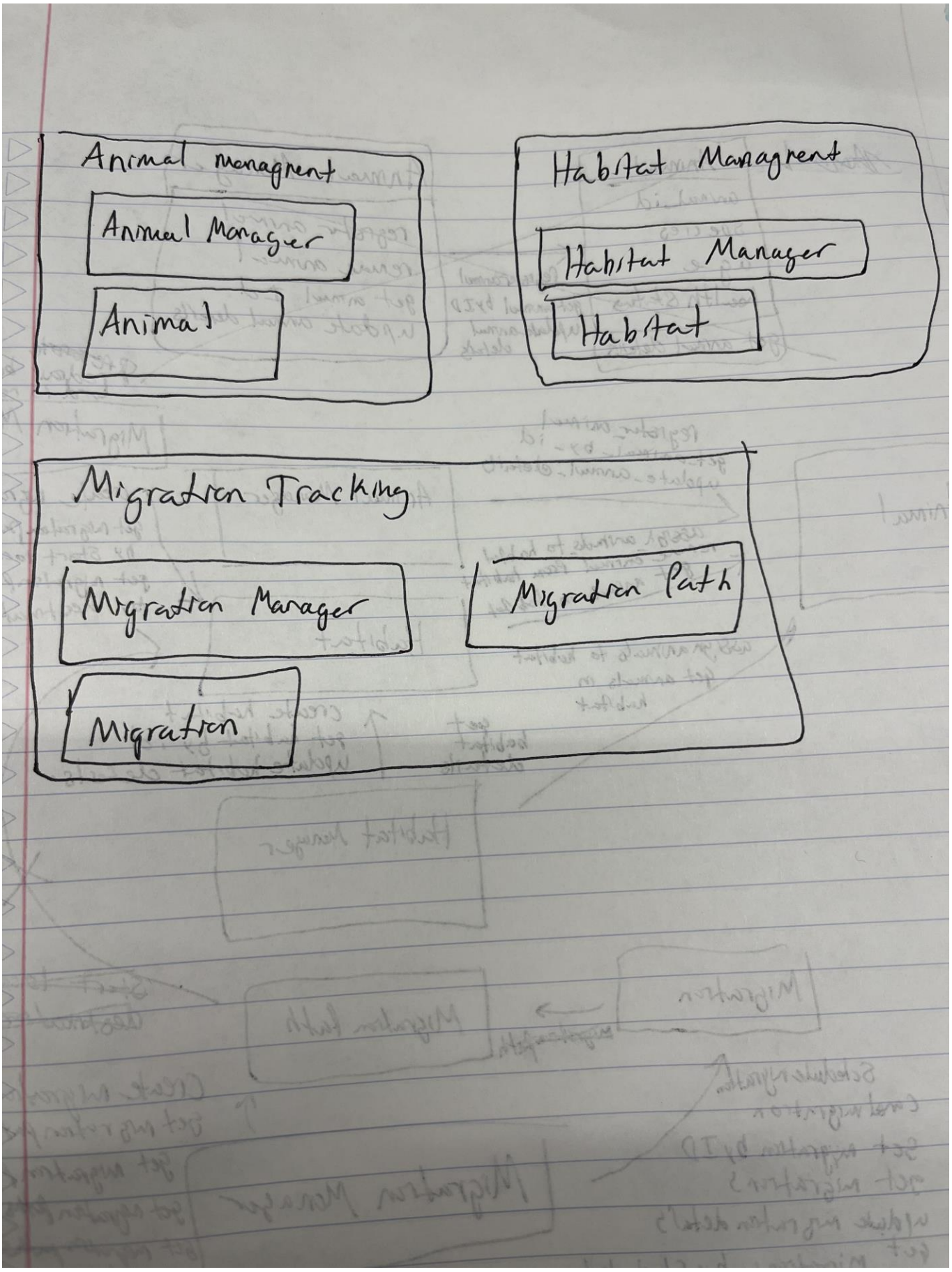
1. List one goal and one non-goal
   Goal: Allow PMs to conduct user studies on SnickerSync to evaluate its effectiveness and gather user feedback.
   Non-Goal: Fully integrating SnickerSync into the main GiggleGit production environment for all users at this stage.

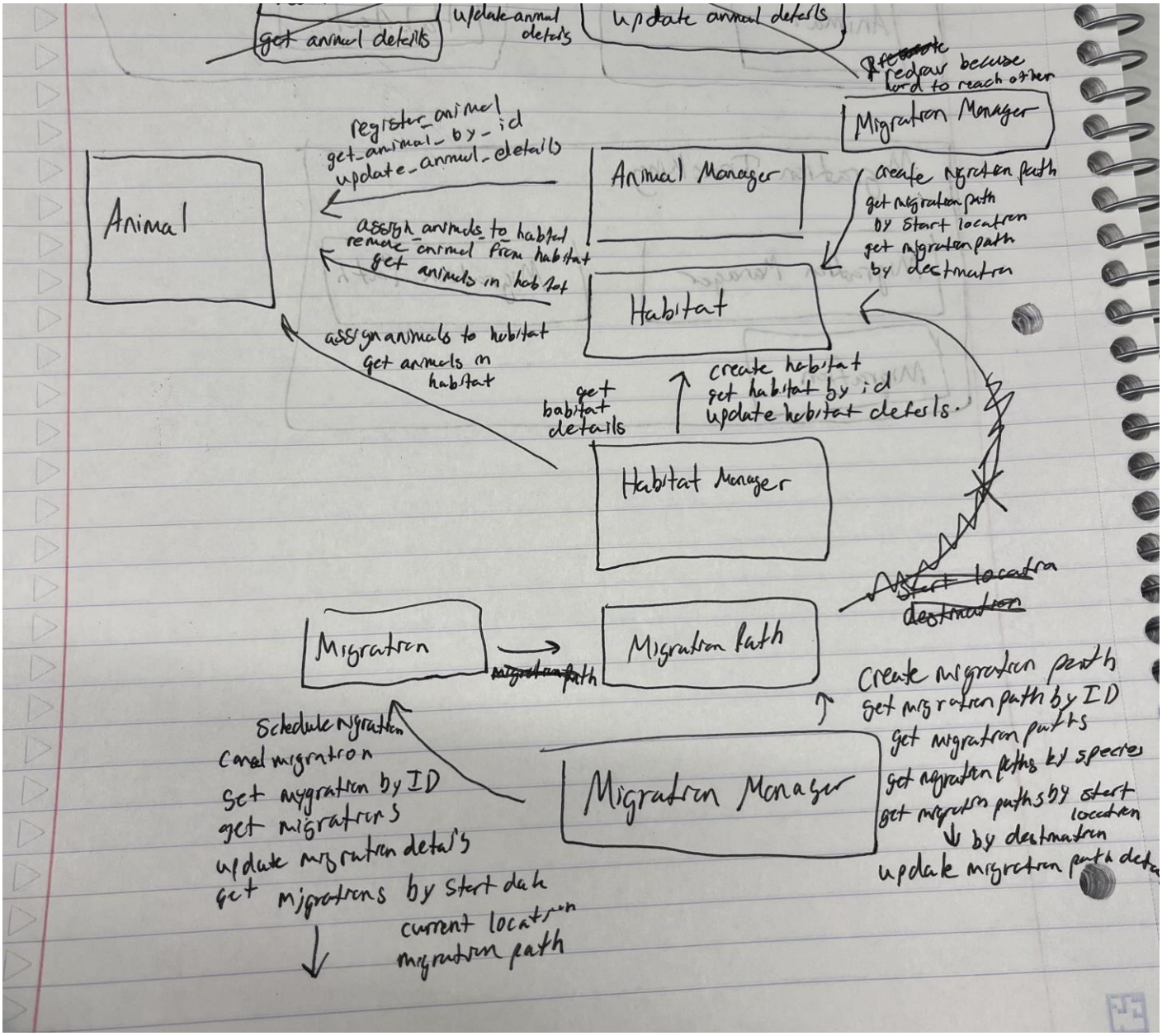
2. Create two non-functional requirements. Here are suggestions of things to think about:

1. Create a secure login system that allows only authorized user study participants and PMs can access SnickerSync. These can have defined roles so they can access specific controls or data.

2. Make sure that all user data collected during the study complies with relevant data protection regulations and inform users that they consent to data collection and can withdraw from the study at any time to stop data collection.

3. For each non-functional requirement, create two functional requirements (for a grand total of four functional requirements).
    1. Develop a system that randomly assigns participants to control and variant groups to ensure unbiased results.
    2. Create a system where group assignments securely to maintain the integrity of the user study.
    3. Create an interface that allows PMs to add, change or remove different snickering concepts without developer approval.
    4. Ensure that changes made by PMs are reflected quickly in the user study environment for real-time testing.


Animal managment
Animal Manager
Animal
Habitat Managment
Habitat Manager
Habitat
Migration Tracking
Migration Manager
Migration Path
Migration


get animal details
update animal details
update animal details
redraw because hard to reach other
Migration Manager
register_animal
get_animal_by_id
update_animal_details
Animal
Animal Manager
create migration path
get migration path by start location
get migration path by destination
assign_animals_to_habitat
remove_animal from habitat
get animals in habitat
Habitat
assign animals to habitat
get animals in habitat
get habitat details
create habitat
get habitat by id
update habitat details
Habitat Manager
start location
destination
Migration
migration path
Migration Path
Schedule migration
cancel migration
Set migration by ID
get migrations
update migration details
get migrations by start date
current location
migration path
Migration Manager
Create migration path
get migration path by ID
get migration paths
get migration paths by species
get migration paths by start location
by destination
update migration path