

CS 411 HW #2

Agile

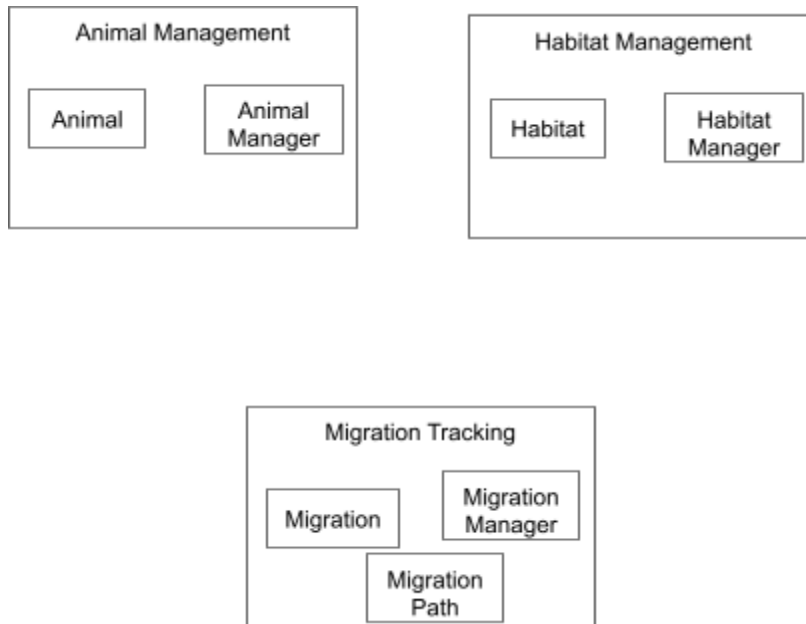
- Complete these user stories:
 - As a vanilla git power-user that has never seen GiggleGit before, I want to...
 - Easily grasp the unique features provided by GiggleGit
 - As a team lead onboarding an experienced GiggleGit user, I want to...
 - Be able to access their previous works in GiggleGit
- 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
 - As a meme enthusiast, I want to be able to customize the memes I use for version control.
 - Task: Enable meme customizability
 - Ticket 1: Make a meme creator
 - Be able to upload an image (both still image & GIF support)
 - Provide options for text creation, image cropping, etc.
 - Ticket 2: Store memes both in and out of use
 - Probably use influxDB
 - Be able to toggle what memes you do and don't want to use
 - Add created memes to the database
- 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's not a user story because it is not a benefit, it's simply a feature.

Formal Requirements -

- **Goal:** Gather information about areas of development for SnickerSynch
- **Non-Goal:** For now we will not be taking suggestions on areas of improvement outside of the main merging functionality of SnickerSynch
- **Non-Functional Requirement #1:** Randomization of User Study Participants
 - **Functional Requirement #1:** The system should automatically assign new participants to either the control or variant groups using a random number generator.
 - **Functional Requirement #2:** The system should log the group assignments to ensure reproducibility of the study for further analysis.

- **Non-Functional Requirement #2:** Access Control for PMs to Manage Snickering Concepts
 - **Functional Requirement #1:** PMs should have a dedicated interface to manage and update snickering concepts, with options for adding, editing, or deleting concepts.
 - **Functional Requirement #2:** The interface should require PM authentication and should restrict access so that only authorized users can modify snickering-related content.

Modules & Classes Diagram



Module Dependencies Diagrams

