HW1

Class	CS411
Date	@October 8, 2024

Agile:

1. Complete these user stories:
    - As a vanilla git power-user that has never seen GiggleGit before, I want to be able to be able to use git commands I'm familiar with to streamline the acclimation process to GiggleGit.
    - As a team lead onboarding an experienced GiggleGit user, I want to be able to manage user profile imports. Thus, allowing the new team member to import their own settings without my manual set up.
2. Create a third user story:
    - As a developer using GiggleGit, I want to customize which types of memes appear during different git merge outcomes, so that I can add a team specific touch to the project.
        - Task:
            - Create a customizable meme selection feature for each team project.
        - Ticket 1:
            - Add meme customization settings to user preferences/settings.
        - Ticket 2:
            - Add tags for memes so the control system knows which memes are associated with which outcome.
3. 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

        - This is more of a request for a feature than a user story. User stories are more focused on a user’s perspective on what they’ll gain.

Formal Requirements:

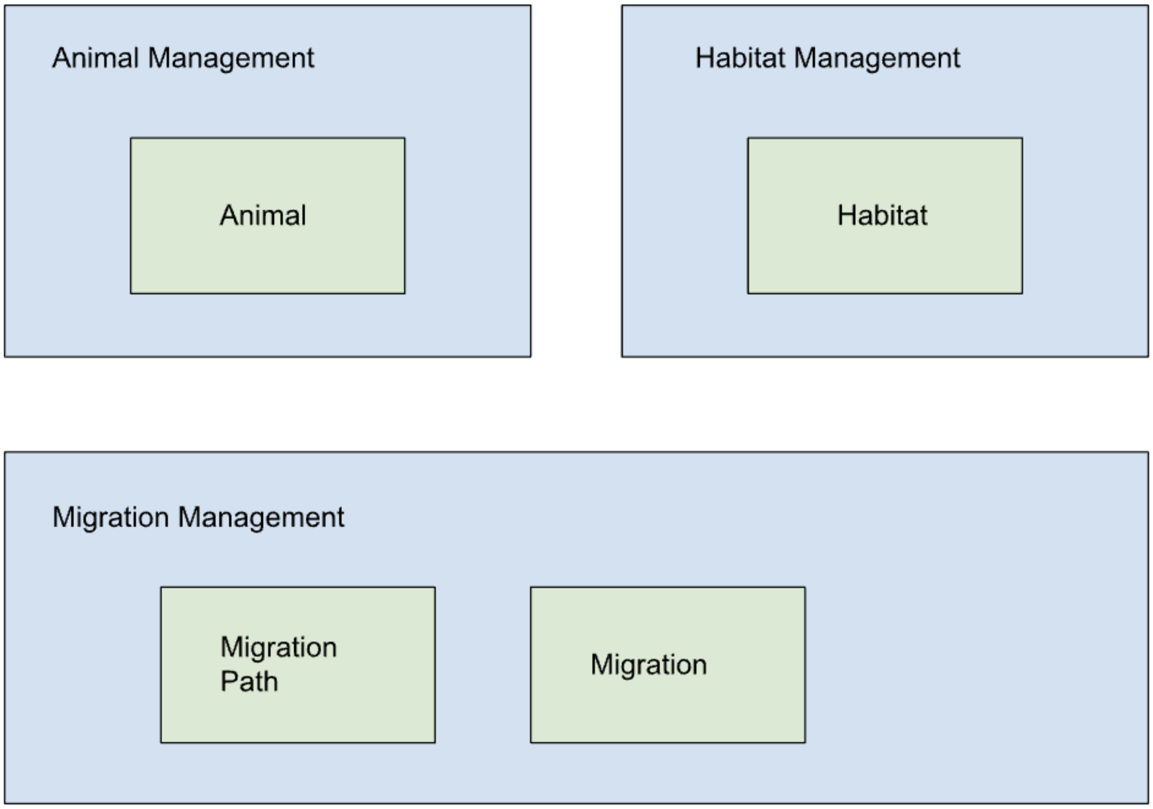
- Goal: Understand how SnickerSync impacts the user experience of merging.
- Non-Goal: Automatically suggest different types of snickers based on the situation.

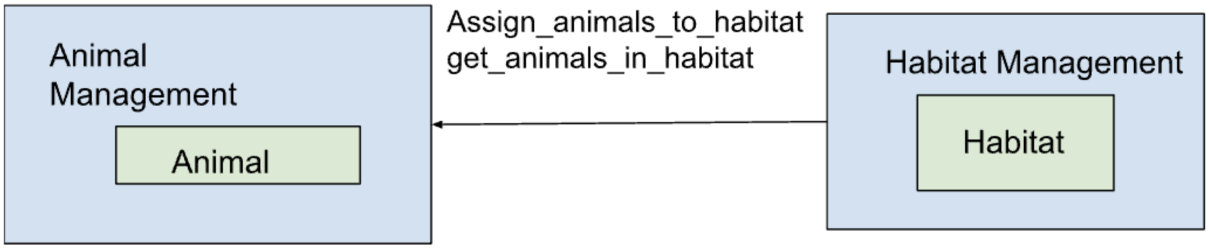
1. Create two non-functional requirements. Here are suggestions of things to think about:
    - A user study needs to have random assignments of users between control groups and variants
2. For each non-functional requirement, create two functional requirements (for a grand total of four functional requirements).

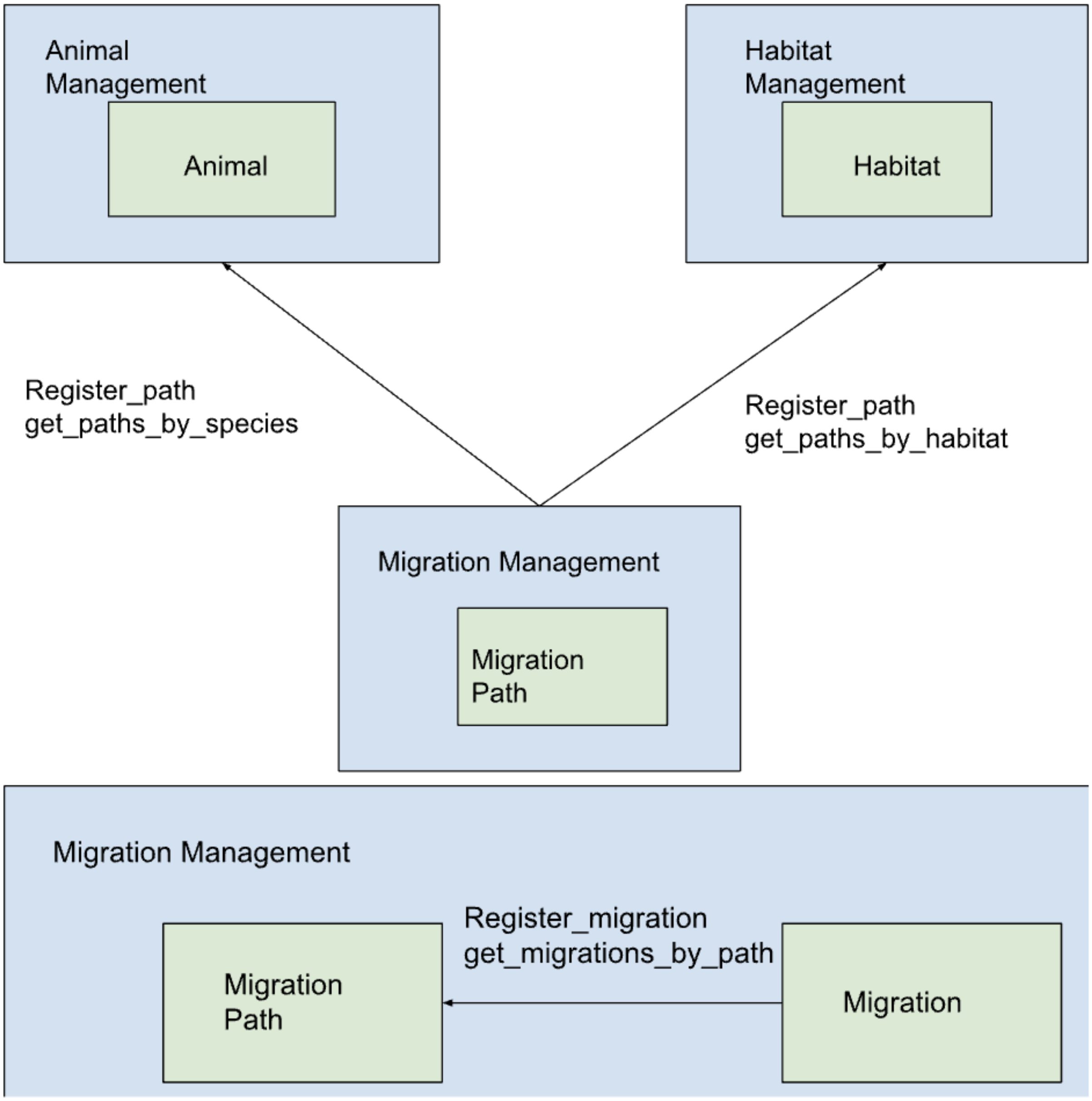
- Non Functional Requirements and their Functional Requirements:
    - Access Control Users:
        - PMs will have specific admin permissions within each project to set the various snickering concepts. Insuring that only authorized people can change snickers.
        - Developers would have access to choose from the available options of snickers but not modify them. Thus, maintaining the format defined by the PM.
    - User Study Assignments:
        - The randomly selected control group would have access to GiggleGit without the addition of the SnickerSync packages to find data on the user experience without SnickerSync.
        - The other half of the group would have access to GiggleGit with the SnickerSync packages to document their experience.

- Ideally, both groups would come in without any preconceived notions of GiggleGit and SnickerSync.

Dependencies:






Animal
Management
Animal
Habitat
Management
Habitat
Register_path
get_paths_by_species
Register_path
get_paths_by_habitat
Migration Management
Migration
Path
Migration Management
Register_migration
get_migrations_by_path
Migration
Path
Migration