Proposal Template:

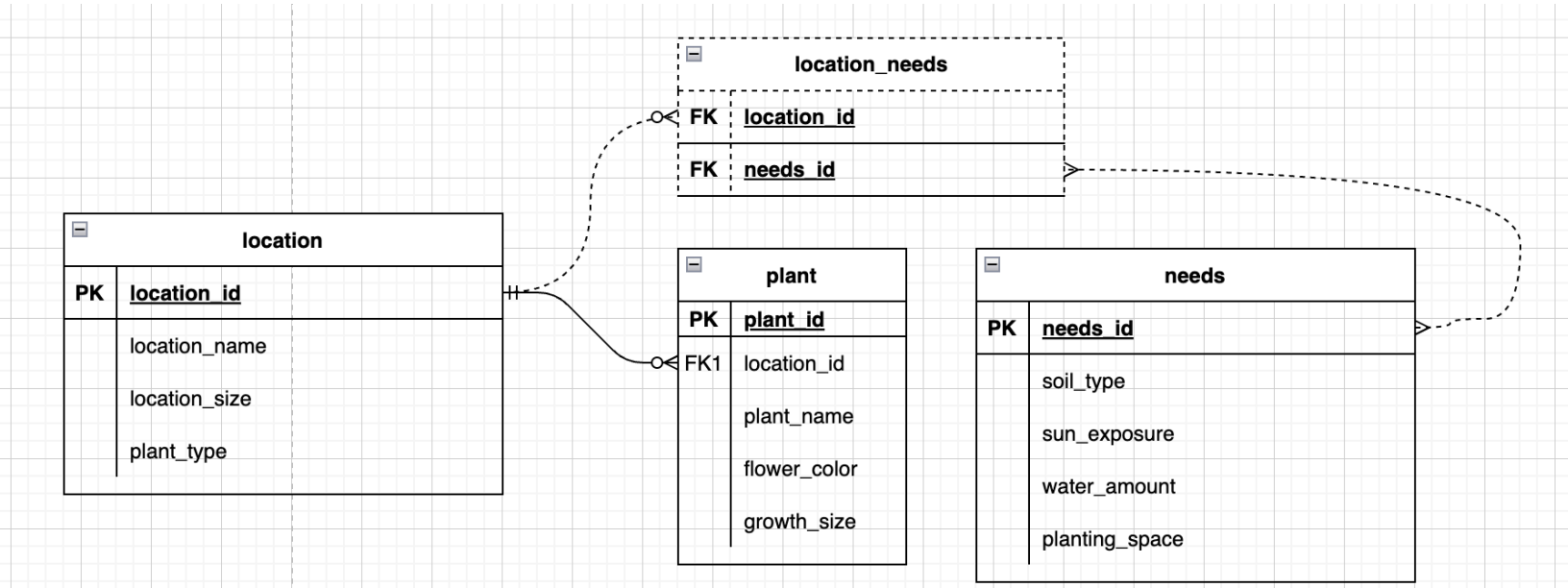
Project Participants: Amanda Dean

Title: Garden final project

Executive Summary:

I will have a **location** table that represents each garden bed. It will have a location name, size and plant type and has a one to many relationship with the **plant** table. This table has the plant name, flower color (if any) and the growth size. There will also be a **needs** table that consist of soil type, sun exposure and water amount. The **plant** and **needs** table will have a many to many relationship with the **plant needs** table.

Initial Features:



Class	Relationship Variable	Annotation
Needs	Set<Location>location	@ManyToMany(mappedBy = "needs", cascade = CascadeType.PERSIST)
Plant	Location location	@ManyToOne(cascade = CascadeTypeALL) @JoinColumn(name = "location_id")
Location	Set<Needs>needs	@ManyToMany(mappedBy = CascadeType.PERSIST) @JoinTable(name = "location_needs", joinColumns = @JoinColumn(name = "needs_id", inverseJoinColumns = @JoinColumn(name = "location_id")
Location	Set<Plant>plants	@OneToMany(mappedBy = "plant", cascade = CascadeType.ALL, orphanRemoval = true)

Create a bulleted list of Initial Features. This will be a list of features that meet the requirements of the final project.

- Database Design -- maybe an ERD which demonstrates the design.
- A bulleted list of planned features you will complete in your project by the deadline.
- A list of API endpoints for each feature. *If you are working in a group, please note which team member will be assigned to each feature.*

- View all locations(GET locations)
- Create location(POST location)
- Update location(PUT location/{locationId})
- Delete location by ID(DELETE location/{locationId)
- View location by ID(GET location/{locationId}
- Create plant(POST plant)
- Create needs(POST needs)

Example of list of Features/Endpoints for a Library API:

- *Entity Ideas: Users, Administration, Books, Genre, Checkout, BookReviews, etc.*
- *Operation (Endpoint) Ideas: (Think: what can a User (or Client Application) do when they use this REST API)*
- *Login, and use system*
- *Browse all Books (GET on Books)*
- *Browse Books by genre (GET on Books with genre specified)*
- *View all details about a specific Book (GET on Books by primary key)*
- *Leave review on a Book (POST in BookReviews)*
- *Read reviews on a Book (GET on BookReviews)*
- *Checkout X amount of books at a time (POST in Checkout), set dueDate for 2 weeks from today (e.g. Checkout Date) and (PUT Change Status of Book to Unavailable)*
- *Return a Book (PUT Change Status of Book to Available, PUT Checkout to RETURNED status, POST Fee on Users, by primary key, if the book is past due, etc.)*

Stretch Goals (to be completed if time allows, or after graduation):

Create a bulleted list of Stretch Goals. These are Enhancements that you will add to your REST API after you get your initial features

implemented. These should be features that may require more research in how to implement or features that would take longer than the allotted time frame.

- Companion planting.. EX- tomatoes are good next to cucumbers but not onions.

Final Reminders:

- This project should be written following the pattern implemented in one of the following projects: **pet-parks**, **dog-rescue** or **pet-store**.
- **Use JPA to create a REST API Server. Test your REST API Server using Postman, Swagger, or AdvancedRestClient (ARC).**