Software Requirements Document for [Group 4]

TEAM: 4

AUTHORS: Hunter Stwalley, Mario Pantaleo, Jake Garnett, William Schulte, Spencer Kane

1. REWRITE SECTIONS IN THIS DOCUMENT WITH YOUR OWN DESCRIPTIONS
2. OMIT SECTIONS MARKED OMIT

Version	Date	Author	Change
0.1		SM	
0.2	10/17	Team4	2.1.1, 2.3, 2.4, 3.2.1, 3.2.2, 3.2.3, 3.5
0.3	10/18	Team4	1.3, 2.1.2, 2.1.2.1
	10/19	Team4	

Table of Contents

[redacted]

1.1 Purpose

The purpose of this document is to establish how the application should interact with the end user, and establish all application requirements functional, and non functional. Once finalized, this document will state what must be accomplished for the application to be considered finished.

1.2 Scope

This SRS covers a number of potential use cases that users may encounter, as well as an overview of the project and its intended uses. It also includes information on the project's UI sketches, but the primary purpose is to give detailed descriptions of anticipated use cases.

1.3 Definitions, acronyms, abbreviations

Term	Description
Facility	A facility is the facility or organization that monitors/tracks butterflies. For example, the facility we are working with is Reiman Gardens in Ames, IA. Organizing by facility allows privacy and personalization for that facility's needs.
Forms	The data entry forms, in which a user fills out with their relevant data, is referred to here as simply "forms". There are several form types that will be filled out by a user. The differences mostly have to do with growth stages and where the specimen is kept. Each form has a model that corresponds to its table in the database.
User	Any person who has registered an account and uses the web application to track data or view other data.

1.4 References

[redacted]

Reiman Gardens would like a better way to log their butterfly growth data. Currently they do this on paper which can be messy and inefficient. With this web-app we will be providing a fast and uniform way to collect his data. The goal is to provide Reiman Gardens and other facilities a standardized method of logging their research data and even allow facilities to share that data.

2.1 Product Perspective

“Unified Butterfly Tracker”, an easy to use data tracking web-app to be used by research facilities and individuals around the US.

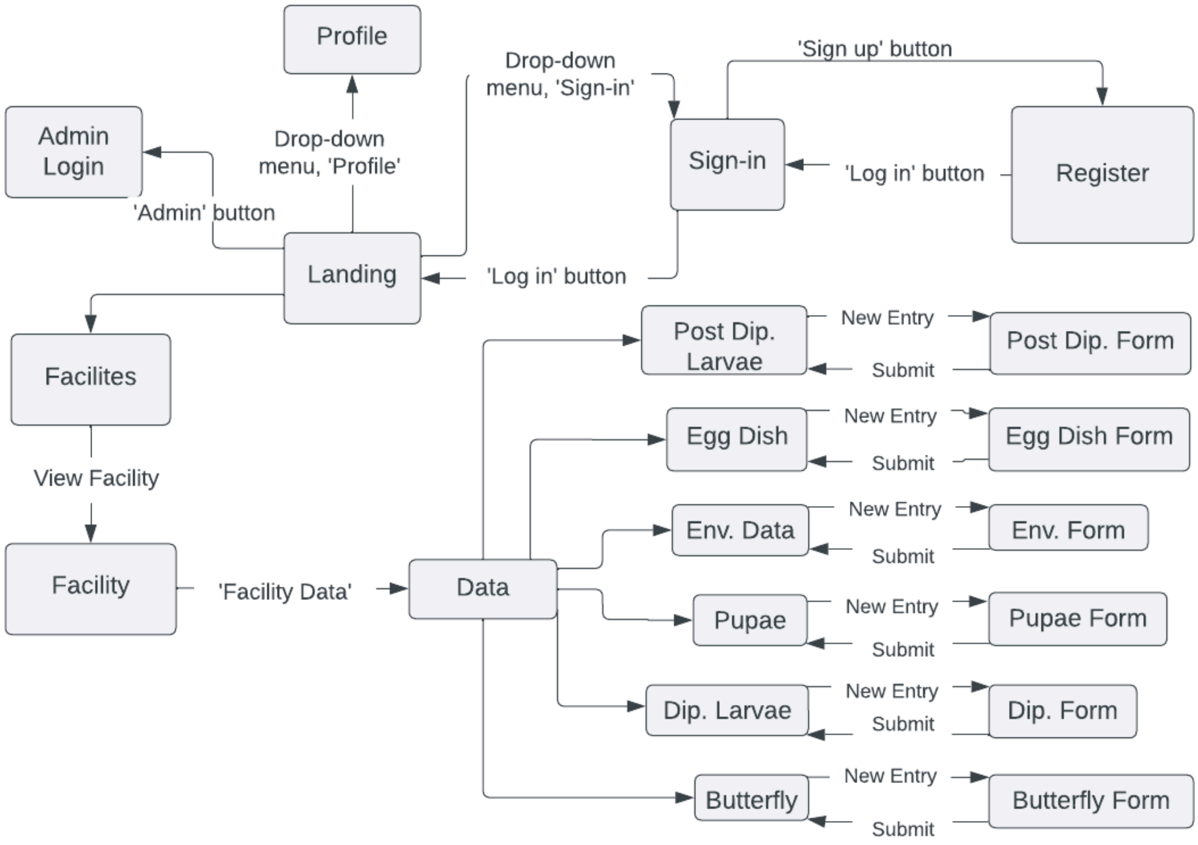
2.1.1 Concept of Operations

This web-based application allows users in a facility to collect data. A user can create an account, view their facility, add data by filling out forms for their facility and edit data. A facility admin grants access to their facility, creates groups/collaborations and manages all users in that facility. A systems admin manages both facility admin and standard users.

A database server will support the system. Here we store the various tables (Butterfly, Egg Dish, Dip. Larvae, Facility Data, Post-Dip. Larvae, Pupae and Users). This will be the uniform framework for the various facilities. There will be the option to include or not include certain entrees to allow facilities to log data how they prefer.

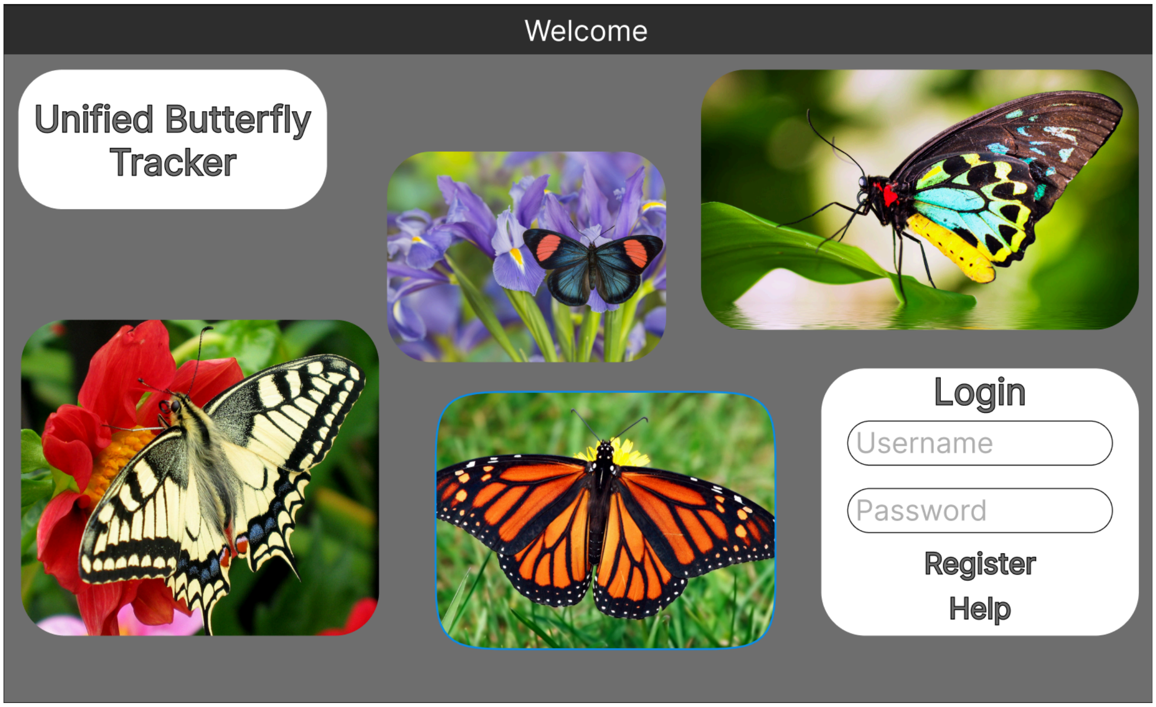
2.1.2 Major User Interfaces

Screen flow Diagram

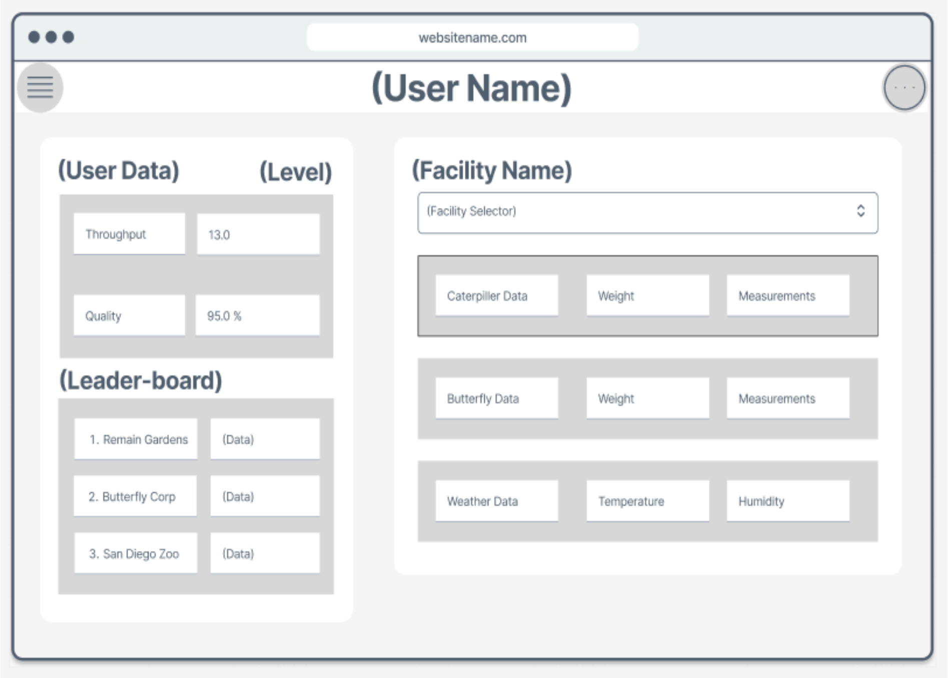


2.1.2.1 Example Screenshot and description

Login



Landing Page



(User Name)

(User Data) (Level)

Throughput	13.0
Quality	95.0 %

(Leader-board)

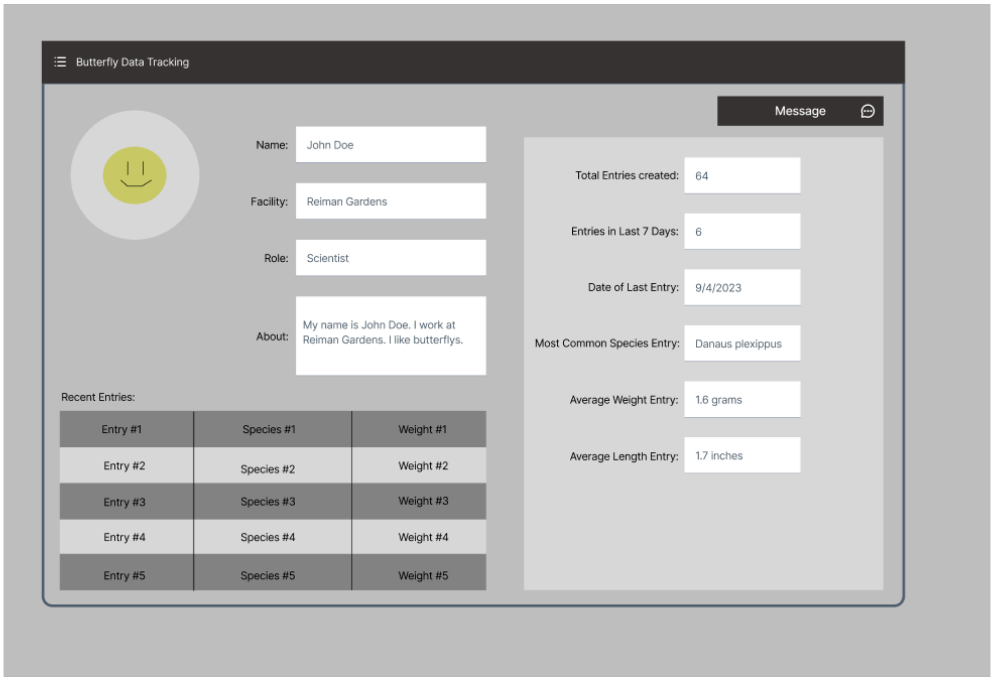
1. Remain Gardens	(Data)
2. Butterfly Corp	(Data)
3. San Diego Zoo	(Data)

(Facility Name)

(Facility Selector)

Caterpiller Data	Weight	Measurements
Butterfly Data	Weight	Measurements
Weather Data	Temperature	Humidity

User Profile



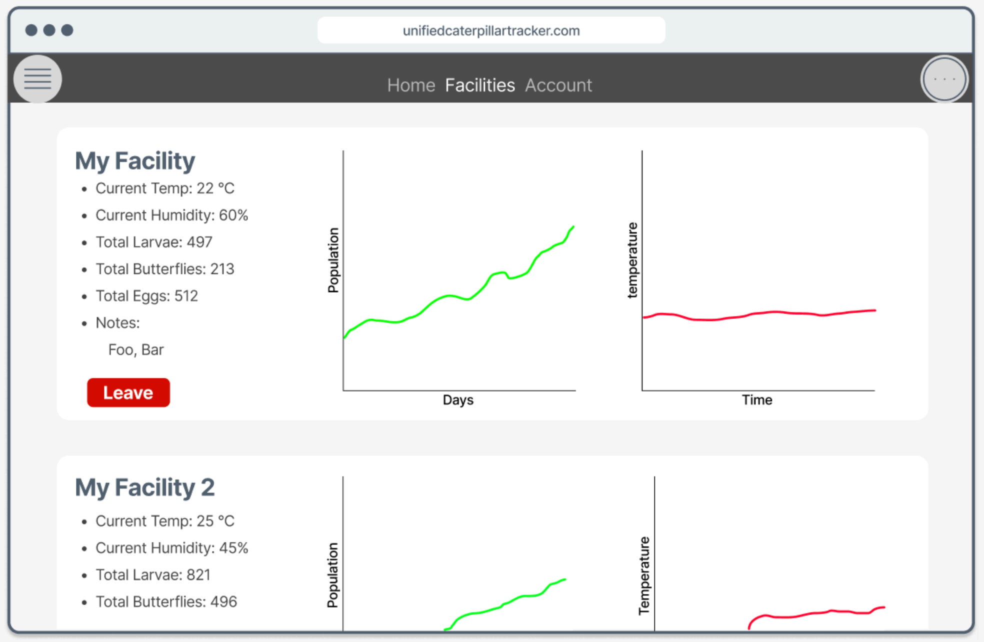
Name:	John Doe
Facility:	Reiman Gardens
Role:	Scientist
About:	My name is John Doe. I work at Reiman Gardens. I like butterflys.

Recent Entries:

Entry #1	Species #1	Weight #1
Entry #2	Species #2	Weight #2
Entry #3	Species #3	Weight #3
Entry #4	Species #4	Weight #4
Entry #5	Species #5	Weight #5

Total Entries created:	64
Entries in Last 7 Days:	6
Date of Last Entry:	9/4/2023
Most Common Species Entry:	Danaus plexippus
Average Weight Entry:	1.6 grams
Average Length Entry:	1.7 inches

Facility Page



My Facility

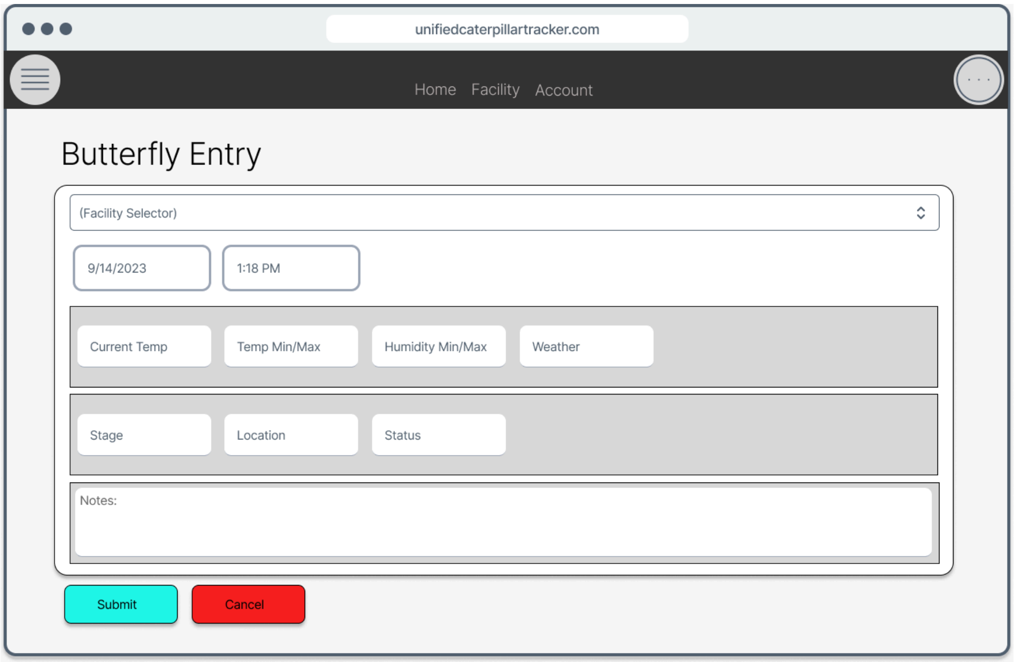
- Current Temp: 22 °C
- Current Humidity: 60%
- Total Larvae: 497
- Total Butterflies: 213
- Total Eggs: 512
- Notes:

Foo, Bar

My Facility 2

- Current Temp: 25 °C
- Current Humidity: 45%
- Total Larvae: 821
- Total Butterflies: 496

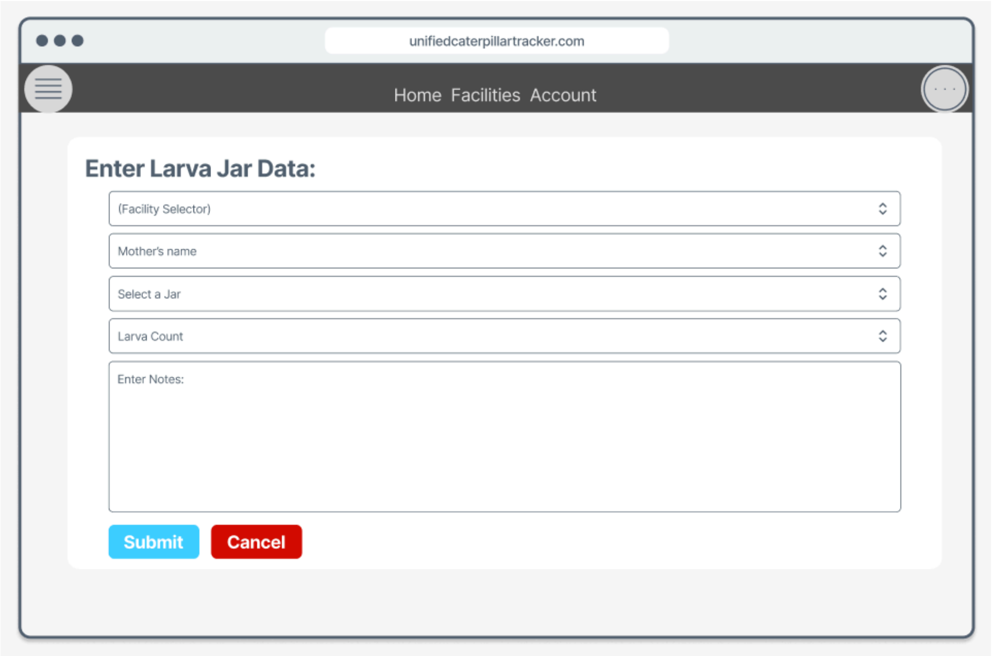
Butterfly Entry Page



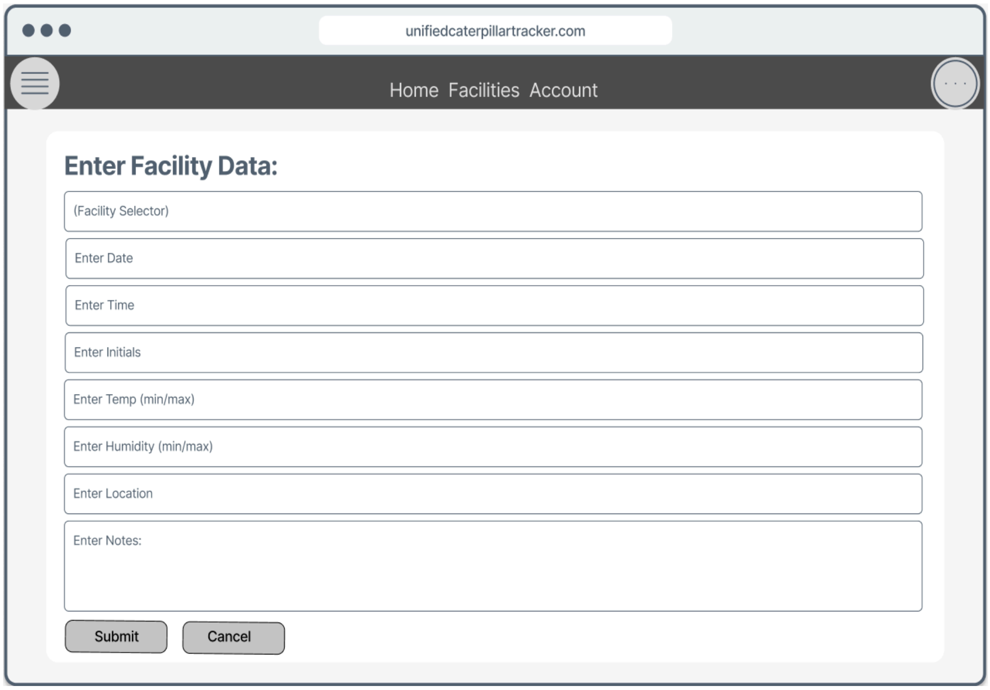
9/14/2023

1:18 PM

Larva Entry Page



Facility Environment Entry


Enter Facility Data:

2.1.3 Hardware Interfaces

Any device that supports a web browser and current standards of HTML, CSS, etc

2.1.4 Software Interfaces

The web app is built using Django as frontend, backend, and database management. Django works by creating (as necessary) and sending HTML from the server to the browser. It also allows you to submit and retrieve information from the database.

2.1.5 Communication Interfaces

// example: modem etc (OMIT for now)

2.1.6 Memory Constraints

Constrained by the amount of memory on the server where the data is stored.

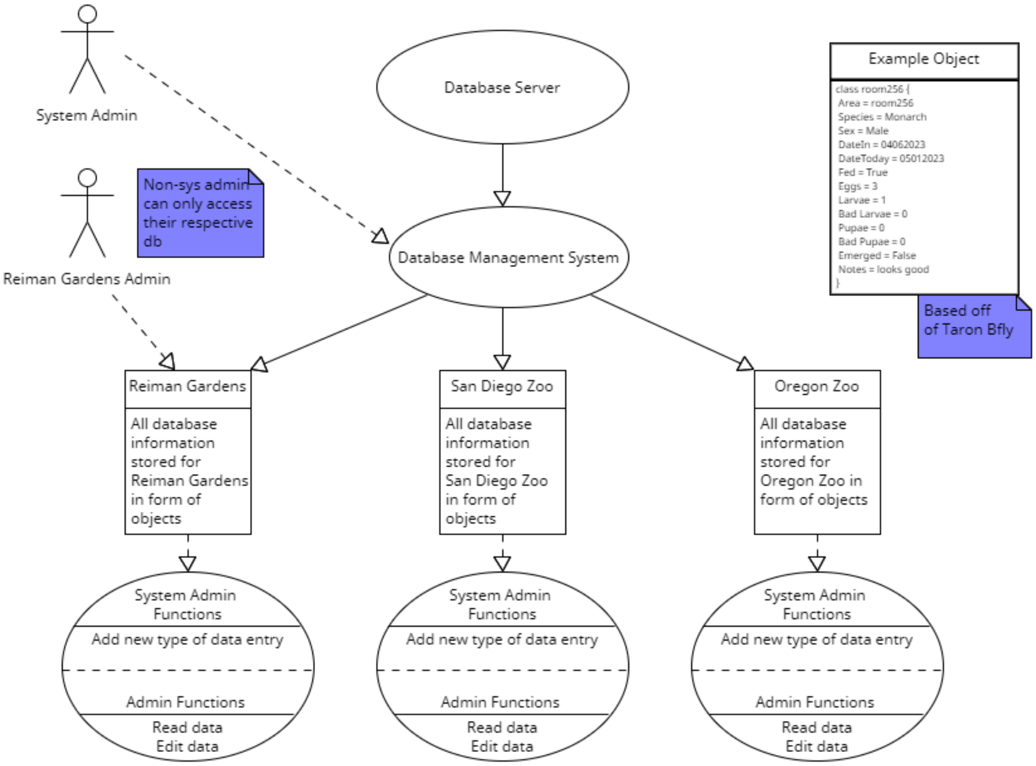
2.1.7 Operations

// special operations (if any) (OMIT for now)

2.1.8 Site Adaptation Requirements

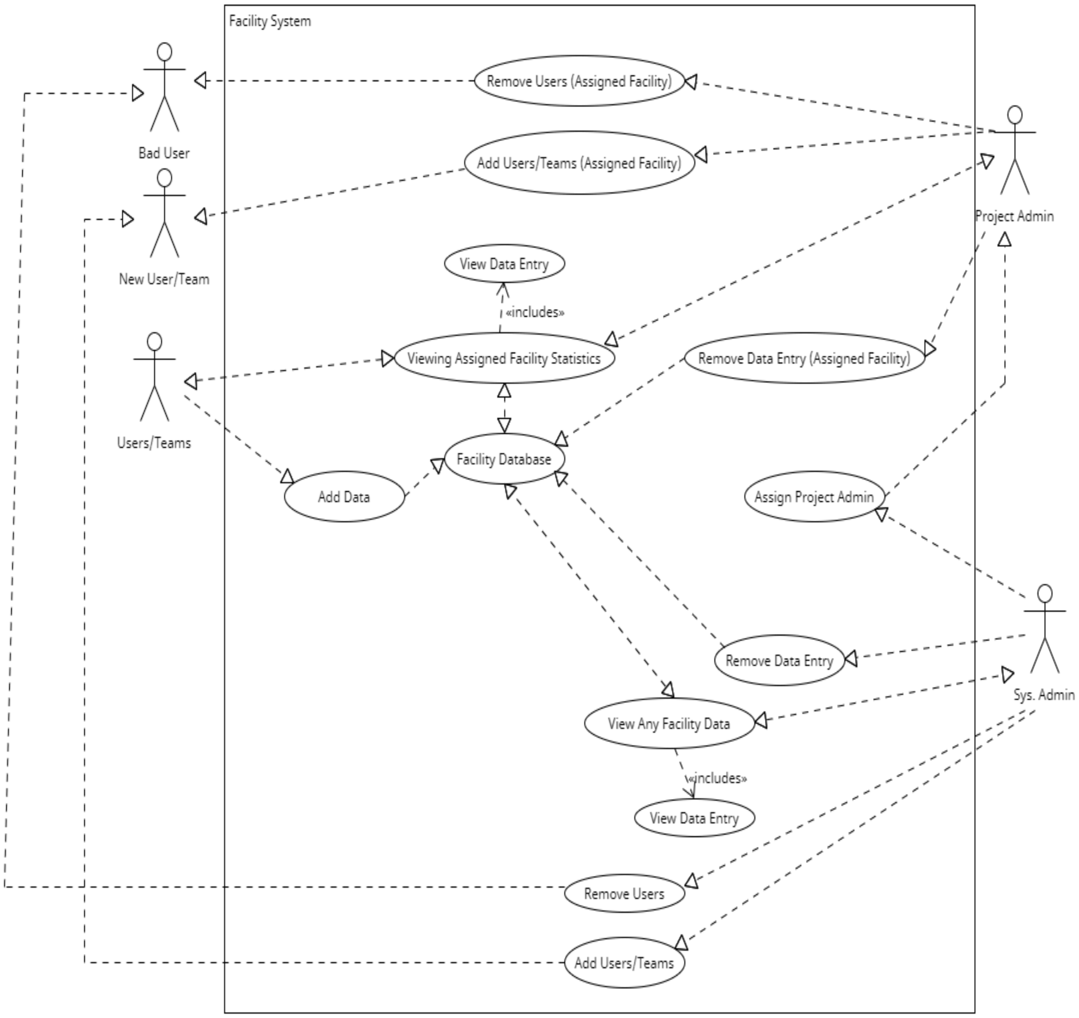
The users will have an option to decide what other users in different groups can see. It also allows them to choose what fields they want to include in the form submissions.

2.2 Product Functions

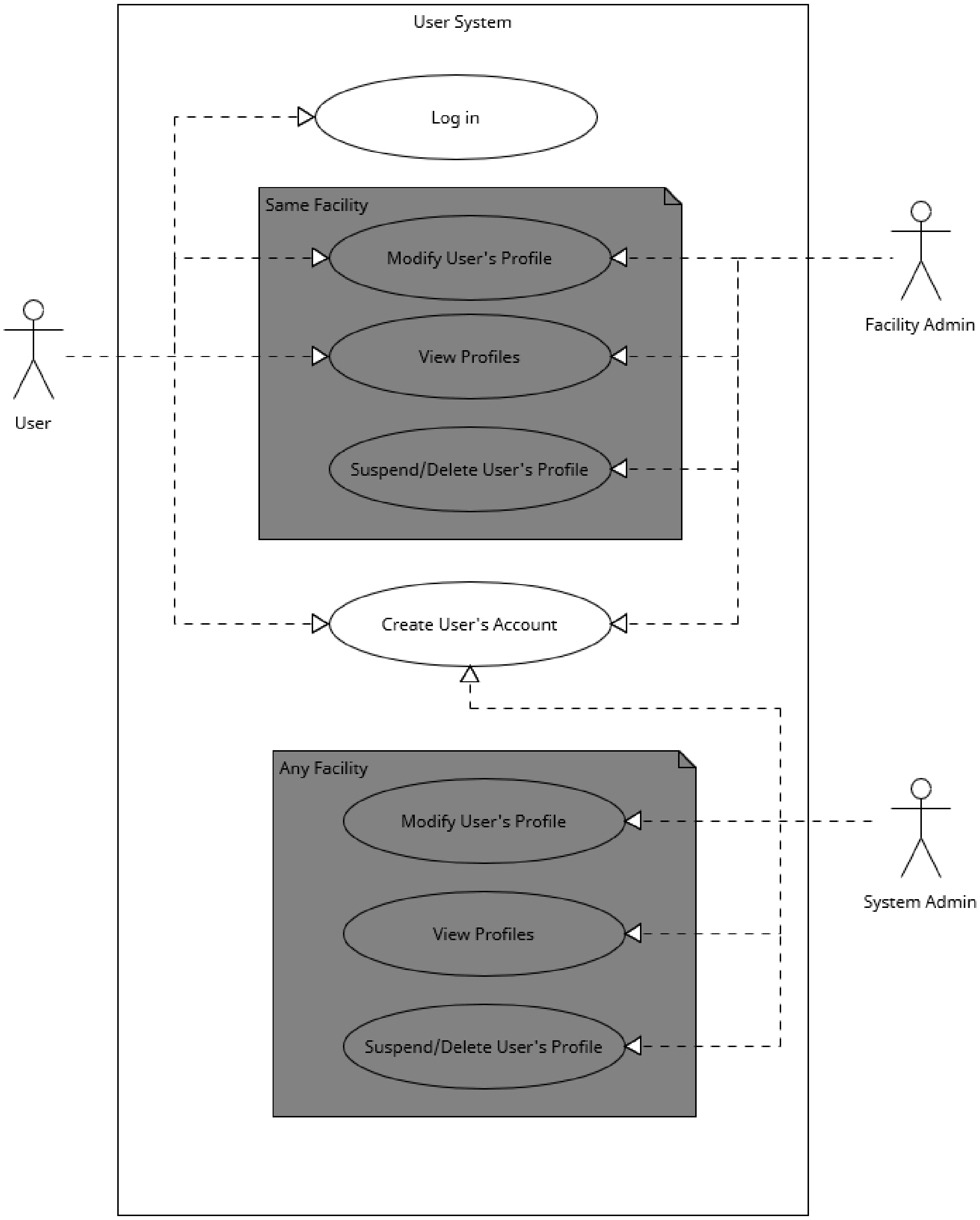

System Admin
Database Server
Example Object
Non-sys admin can only access their respective db
Reiman Gardens Admin
Database Management System
Based off of Taron Bfly
Reiman Gardens
All database information stored for Reiman Gardens in form of objects
San Diego Zoo
All database information stored for San Diego Zoo in form of objects
Oregon Zoo
All database information stored for Oregon Zoo in form of objects
System Admin Functions
Add new type of data entry
Admin Functions
Read data
Edit data
System Admin Functions
Add new type of data entry
Admin Functions
Read data
Edit data
System Admin Functions
Add new type of data entry
Admin Functions
Read data
Edit data

```
class room256 {
 Area = room256
 Species = Monarch
 Sex = Male
 DateIn = 04062023
 DateToday = 05012023
 Fed = True
 Eggs = 3
 Larvae = 1
 Bad Larvae = 0
 Pupae = 0
 Bad Pupae = 0
 Emerged = False
 Notes = looks good
}
```

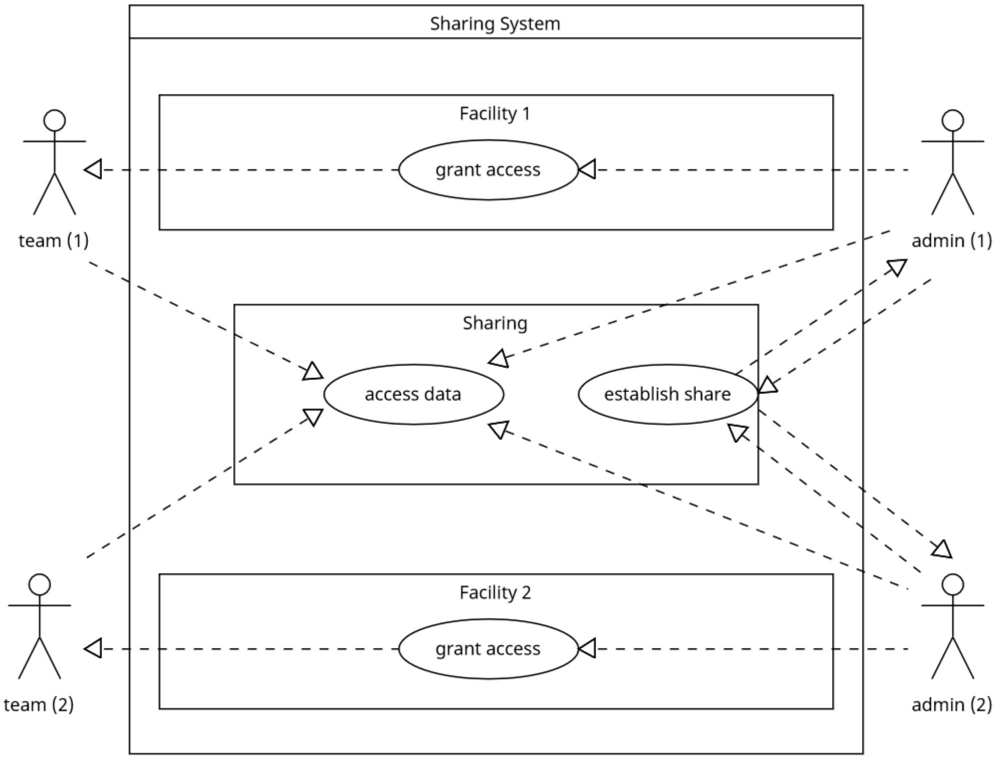
The above use case diagram models the interactions between the system administrators, facility administrators (Reiman Gardens Admin), and the database.



The above use case diagram models the interactions between users, administrators, and the respective facility.



The above use case diagram models the interactions between users, facility admins, system admins when managing user accounts.



The above use case diagram shows the interactions between the facility sharing system in which facilities can share data with each other.

2.3 User characteristics

The web-app will be used to enter or access data. This may be multiple times a day or less so based on the user and their tasks. They will be able to do this through the creation of a new form or editing of a previous one. Admins will be able to invite other users to a facility or remove users from a facility. This will happen much less frequently.

2.4 Constraints

// all conditions that may limit design options (INCLUDE NON FUNCTIONAL CONSTRAINTS)
Client is after something fast but is also interested in gamification. The gamification aspect may work against the speed with its bloat. If using the camera functionality to determine size of specimen we must be aware of the variables that could mess with results. Maintaining height and stability will require some physical rig. From a non-functional standpoint the main constraints are time and maintainability. We have a limited time to develop this meaning we must prioritize. When our semester is up it is out of our hands.

2.5 Assumptions and Dependencies

// hardware and software assumptions and dependencies
It is assumed that the hardware being used is able to access the website via a browser. It is assumed that the software will be accessible from the web. The dependencies will be on the Django models.

[redacted]

// Here you need to put in details (if any). Mark items [None] if you do not have any information.

HERE instead of looking at users and user stories, look at features of the system.

For example, You can think of a car and view it in terms of features. For example, steering, cruise-control, air-bags, 4-wheel-drive etc.

DO ENTER NONFUNCTIONAL REQUIREMENTS (like maintainability, extensibility etc)

3.1 External Interface Requirements [OMIT THIS SECTION]

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communications Interfaces

3.2 FEATURES

3.2.1 Form Data Entry

3.2.1.1 Butterfly Table

3.2.1.2 Egg Dish table

3.2.1.3 Diapause Larvae table

3.2.1.4 Post-Diapause Larvae table

3.2.1.5 Pupae table

3.2.2 Facilities

3.2.2.1 Unique facility information

3.2.2.2 Facility admin

3.2.2.3 Data private to that facility

3.2.2.4 Messaging between users in facility

3.2.3 User classifications

3.2.3.1 Standard user

1. create account
2. add/edit/view data
3. send messages
4. view facility
5. view profile

3.2.3.2 Facility admin

1. create/delete standard users
2. add users to group
3. edit/delete data
4. confirm collaborations

3.2.3.3 System admin

1. confirm facility request
2. add/edit/delete facility

3.3 Performance requirements

Must be accessible from any mobile device or computer. Data must be stored securely and separate from other facilities.

3.4 Design Constraints

Client is after something fast but is also interested in gamification. The gamification aspect may work against the speed with its bloat. It is difficult to include every new feature that is thought up during meetings. We have a limited time to develop this meaning we must prioritize. When our semester is up it is out of our hands.

3.5 Software System Attributes (THESE ARE NON-FUNCTIONAL REQUIREMENTS)

3.5.1 Speed

1. Load data from previous form into current form to minimize retyping common data.
2. Use facility sensors to partially automate data entry.

3.5.2 Engagement

1. Gamify data entry to make it less monotonous.
2. Include facility leaderboard.

3.5.3 Standardization

1. Provide standardized forms to all butterfly husbandry facilities.
2. Standardize collaboration methods between facilities.

3.5.4 Security

1. Implement user, facility admin, and system admin permissions
2. Require a user to be admitted to a facility before being able to access husbandry data.
3. Data is private, unless a facility admin chooses to share.

3.5.5 Portability

1. Web application is accessible on desktop, laptop, or mobile device.

3.6 Other Requirements

// ADD Appendices (if any)

// Regenerate Table of Contents