Software Requirements Document for [Group 7]

TEAM: ____7___

AUTHORS: On Wave Tiong, Tze Yik Ong, Kaden Wehrenberg, Richard Bach, Joseph Jennings

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

Version	Date	Author	Change
0.1		SM	
1.0	5/2/24	Group7	

Table of Contents

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, acronymns, abbreviations

Term	Description
4-year Plan / Flowchart/Schedule	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.
Course / Class	COMS 402C
Prerequisite	COMS 309, COMS 311, COMS 321, COMS 331
User	Any person who has registered an account and uses the web application to track data or view other data.

1.4 References

1.5 Overview

[OMIT]

[redacted]

Reiman Gardens would like a better way to log their butterfly longevity 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 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

"Butterfly Longevity" is an easy to use butterfly tracking web-app that is to be used by research facilities.

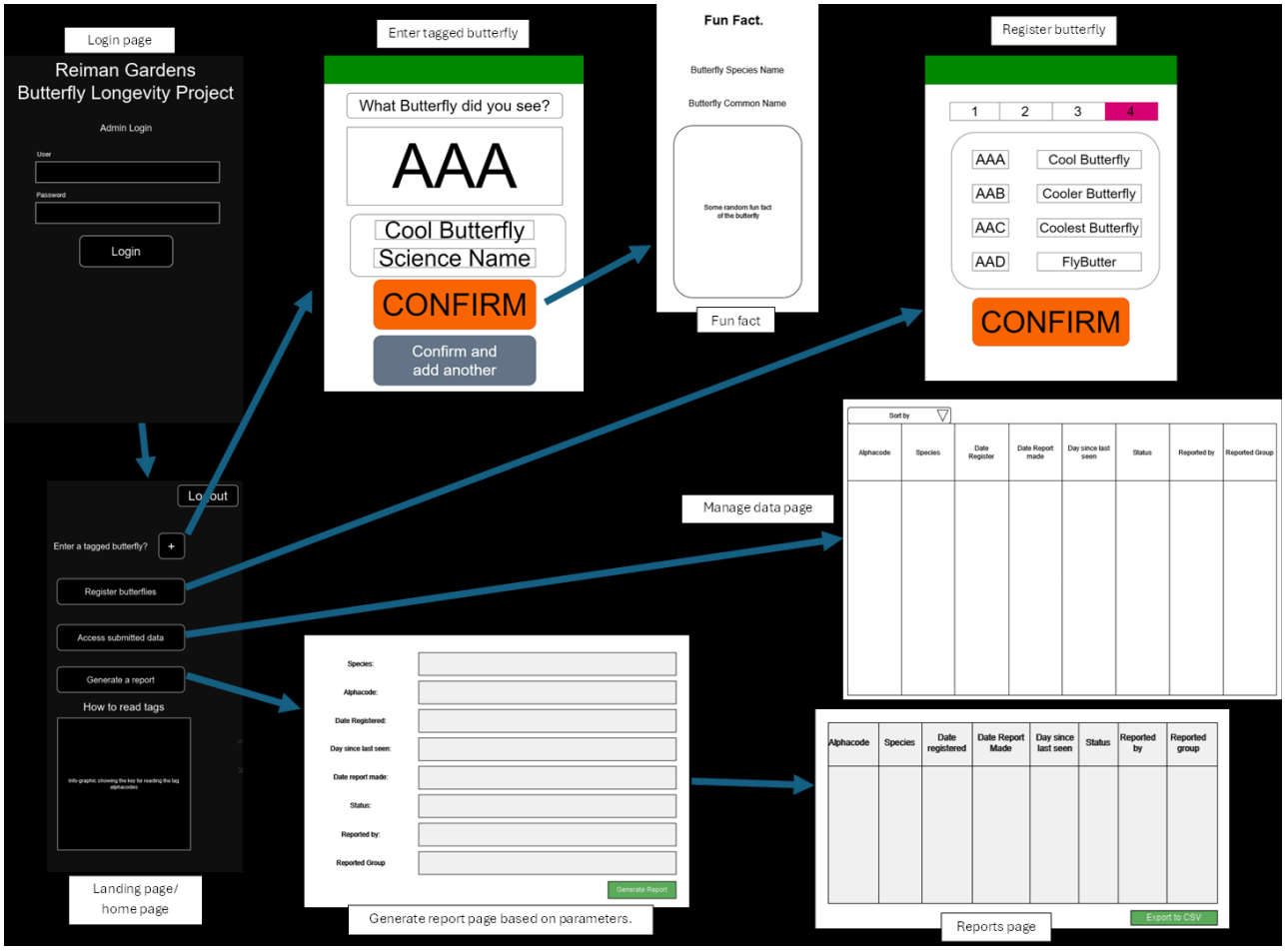
2.1.1 Concept of Operations

This web-based application allows users in a facility to collect butterfly data. An admin can create an account, view their butterfly, add butterfly for their facility and edit data. A facility admin grants A systems admin manages both data from docent and the public. A public user can tag butterfly by inserting alpha code into the system, same as docents.

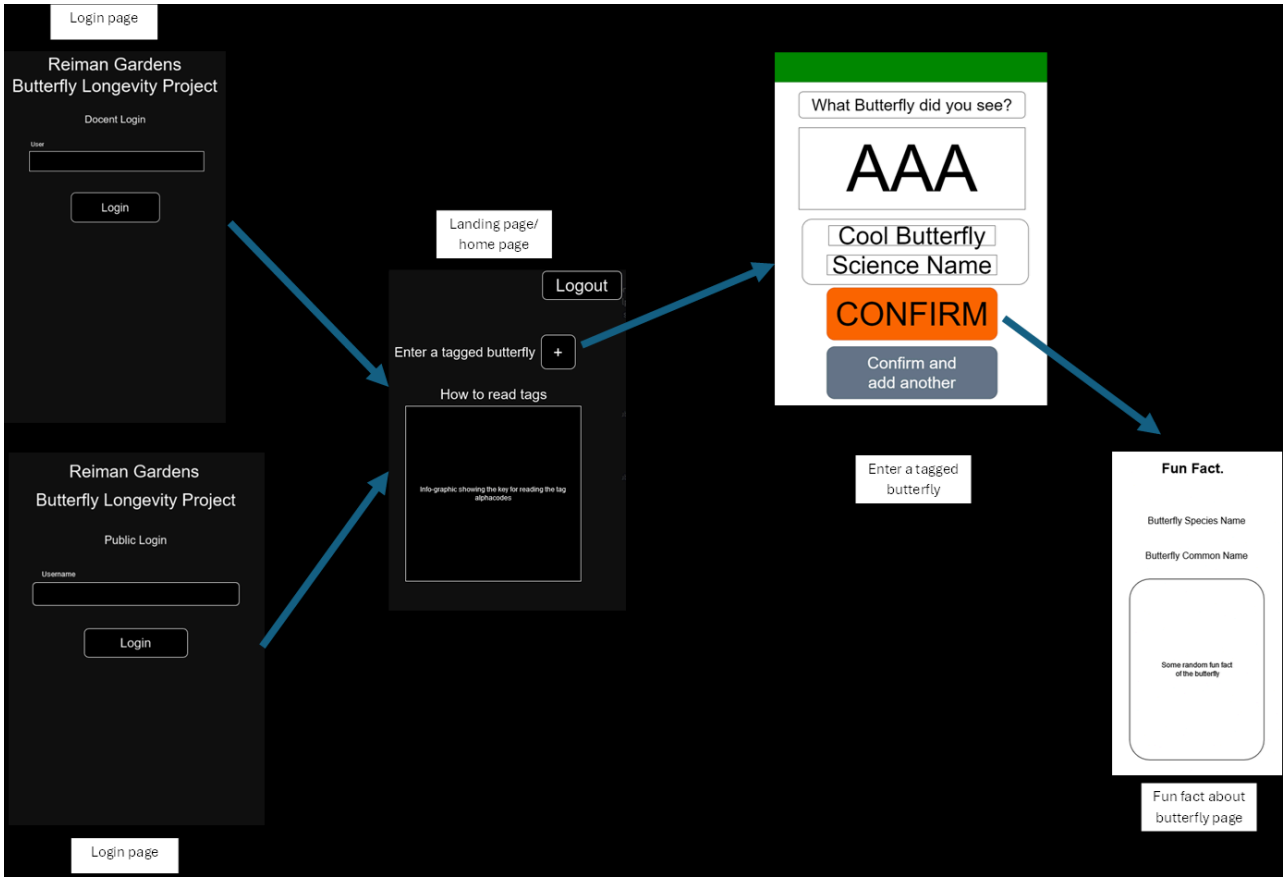
A database server will support the system. Here we store the various tables to keep track of data about each individual butterfly and who tagged them.

2.1.2 Major User Interfaces

Admin Screen Flow

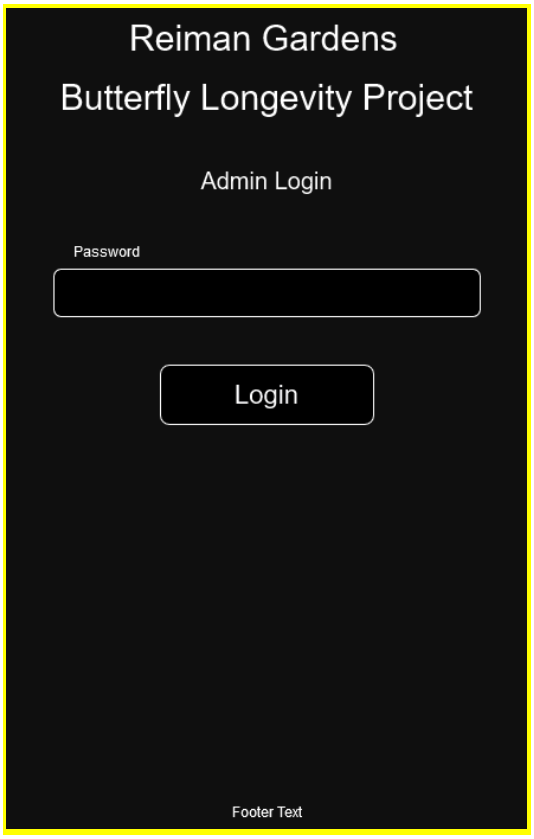


Public/Docent Screen Flow

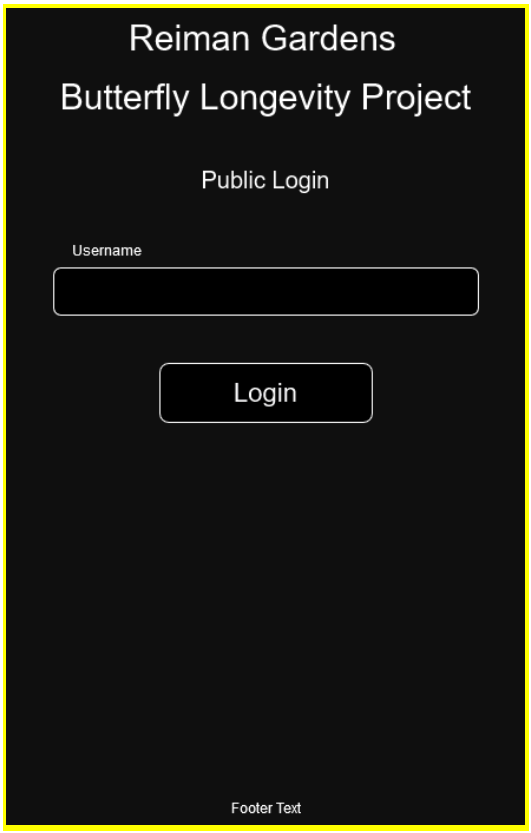


2.1.2.1 Example Screenshot and description

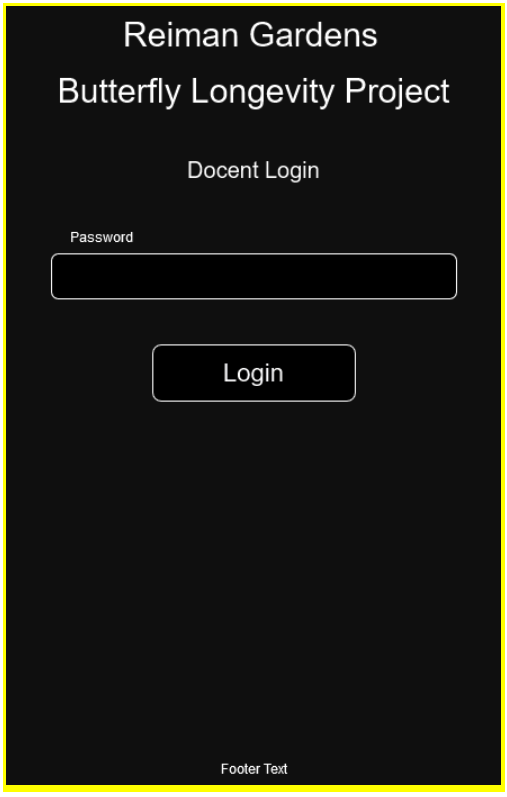
Admin Login



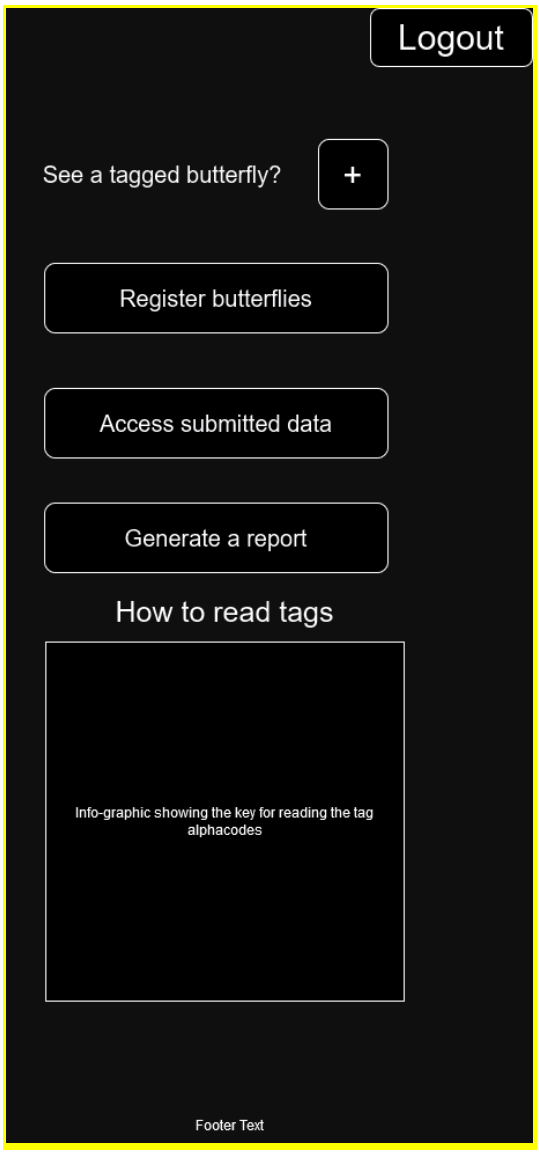
Public Login



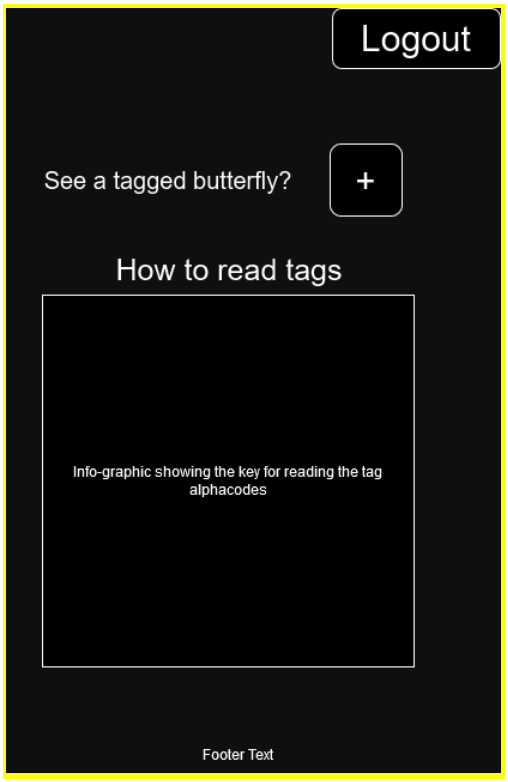
Docent Login



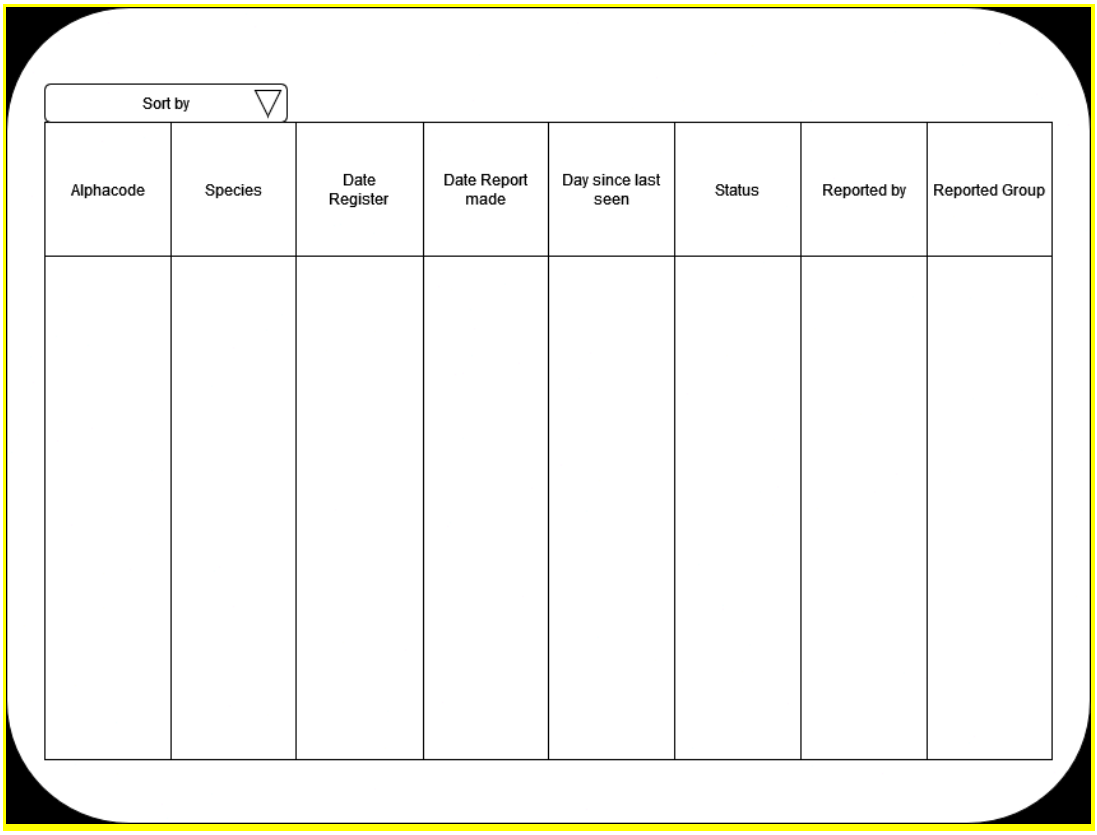
Admin Landing Page



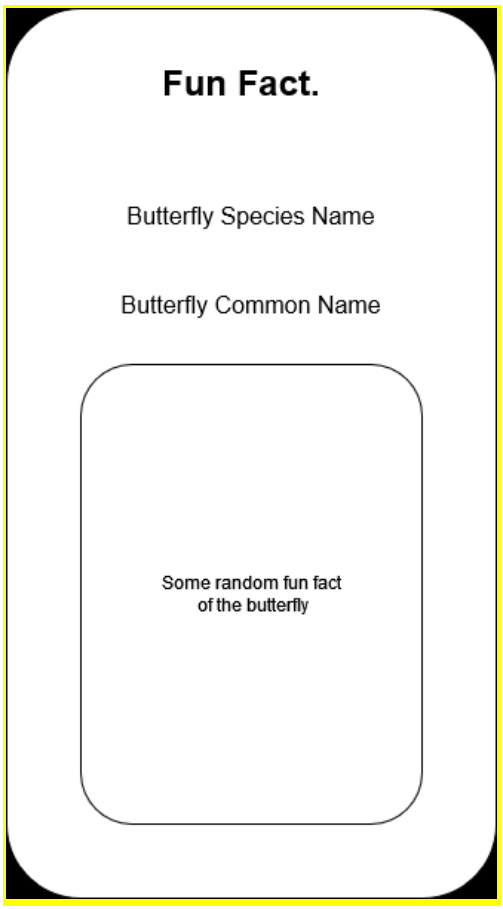
Public/ Docent Landing Page



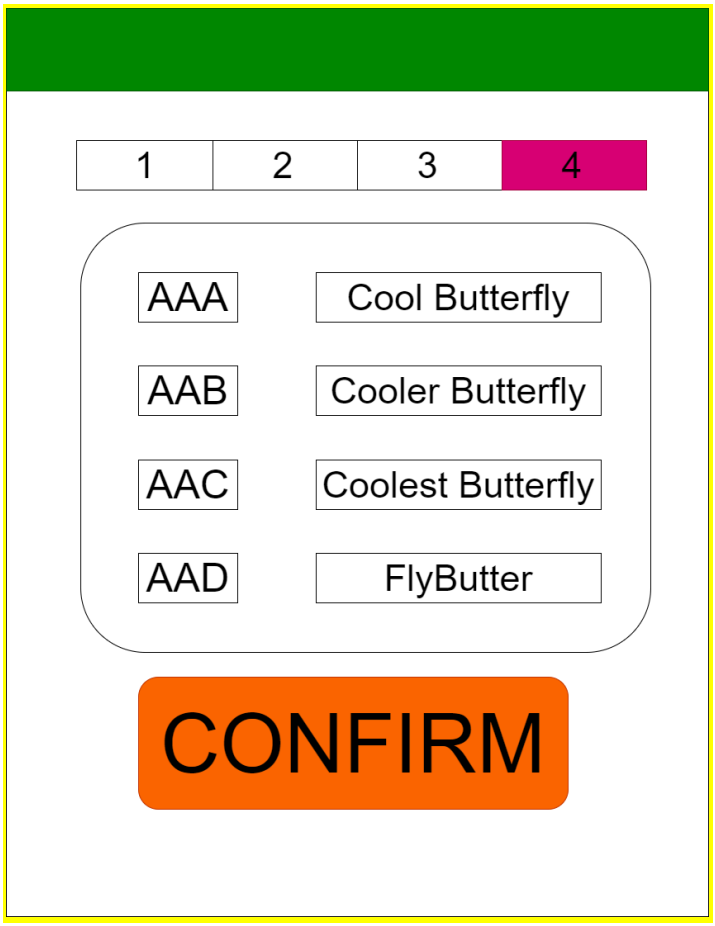
Manage Data page



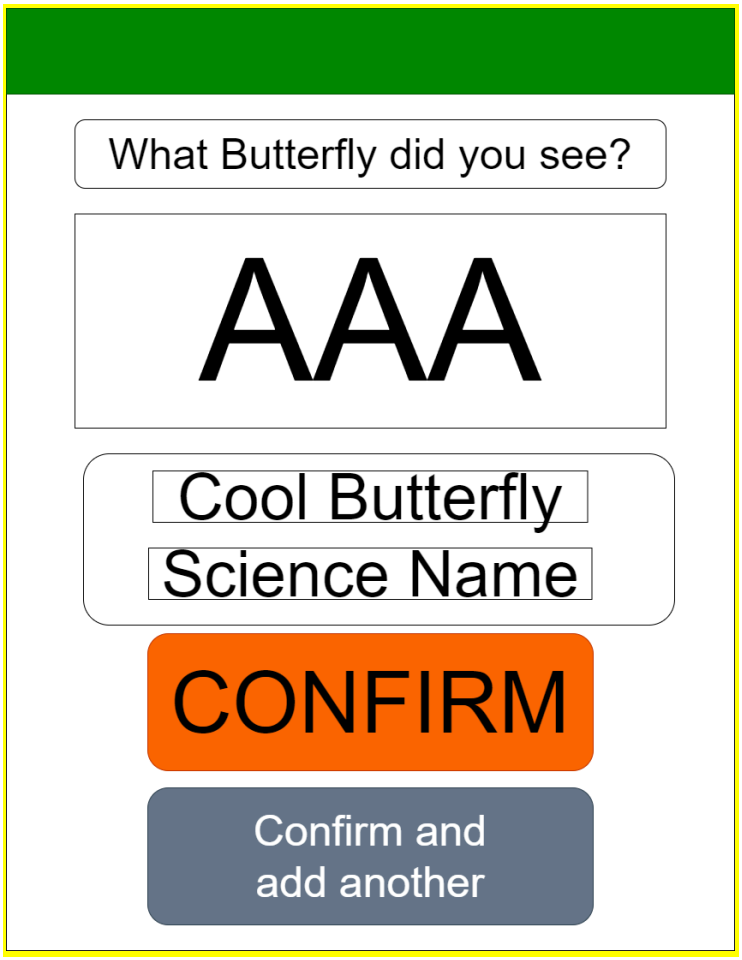
Fun fact Page



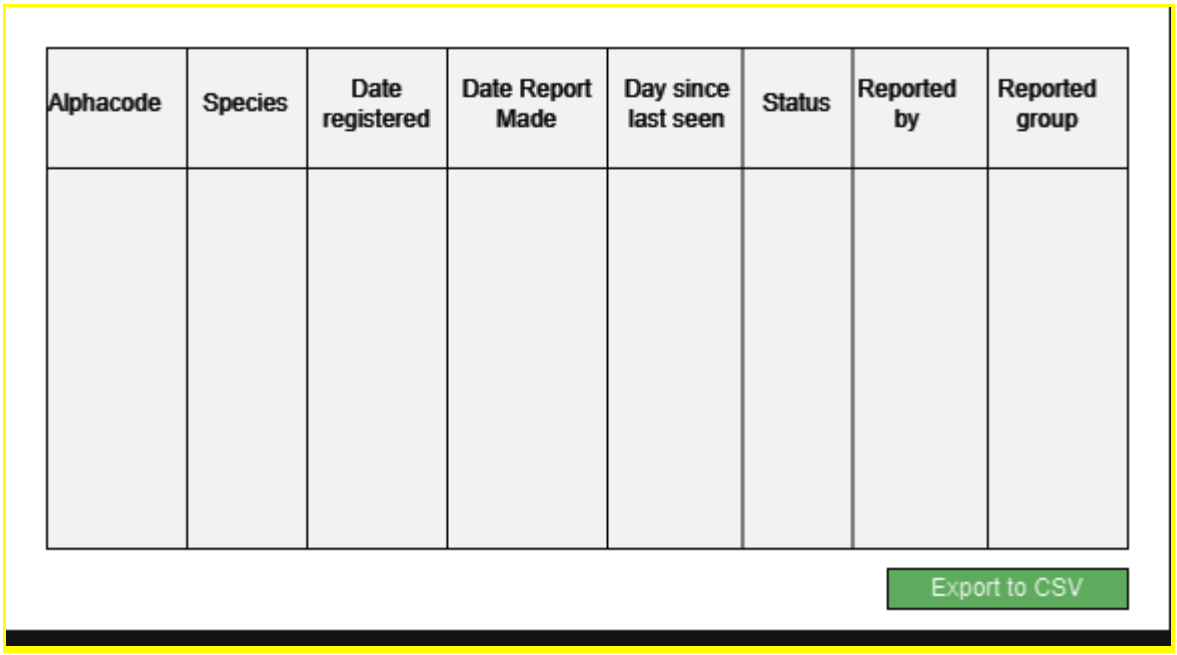
Admin Register Butterfly Page



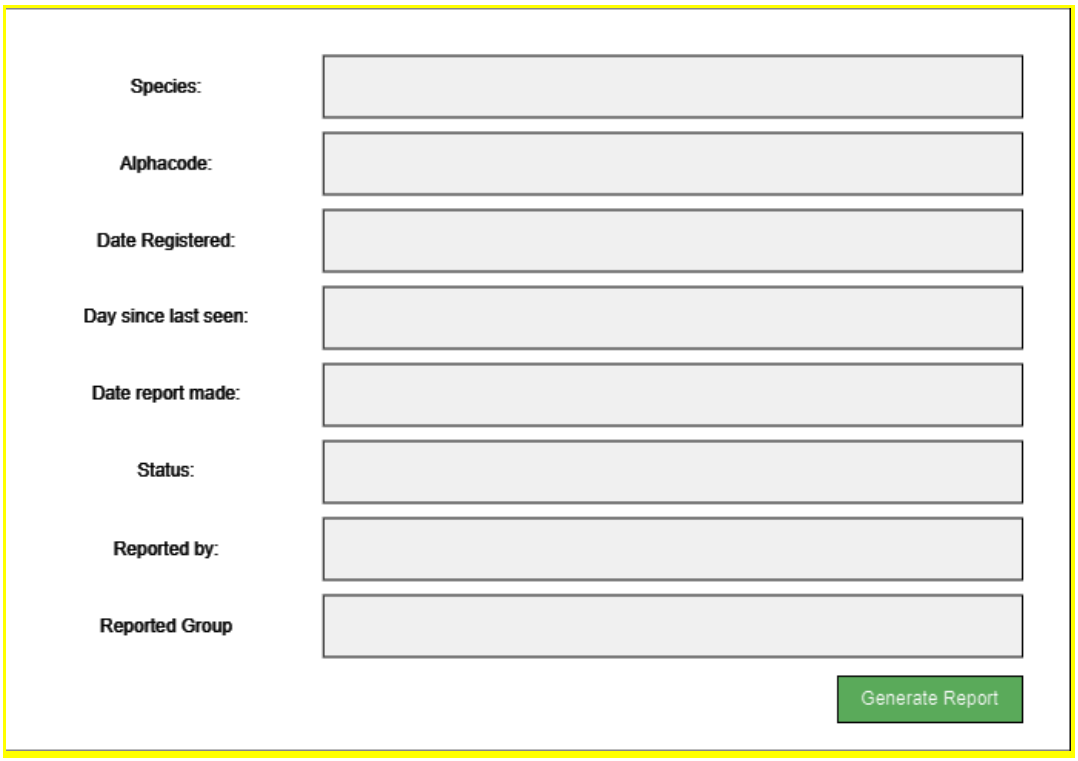
Public/Docent Tag Butterfly Page



Data Report Page



Generate Report Page



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 React JS as frontend, Java Spring Boot backend, and AWS MySQL database management.

2.1.5 Communication Interfaces

// example: modem etc (OMIT for now)

2.1.6 Memory Constraints

// RAM, and other storage constraints (OMIT for now)

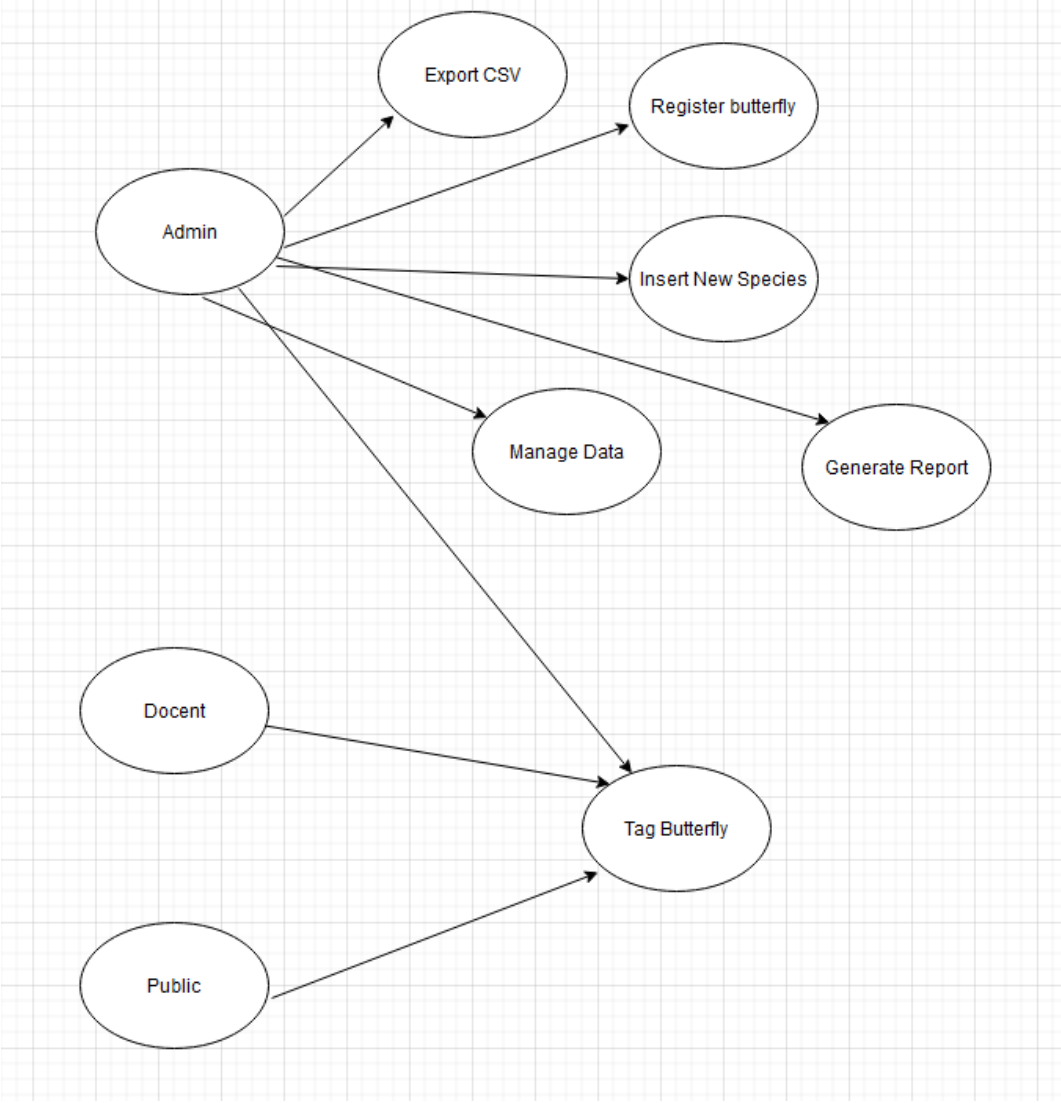
2.1.7 Operations

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

2.1.8 Site Adaptation Requirements

//ex: Japanese language etc (OMIT for now)

2.2 PRODUCT FUNCTIONS



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

2.3 User characteristics

The web-app will be used to enter or access data by admin. They may input hundreds of butterfly data a day depending on the situation. Public and Docents are also able to enter data into the app which can be managed by the admin.

2.4 Constraints

// all conditions that may limit design options (INCLUDE NON FUNCTIONAL CONSTRAINTS)
Client is wants the web application to be used on a tablet like device and smaller handheld device. We have to set the website so it adjust automatically according to the size of the user screen no matter how big or small.

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 on mobile devices like an iPad or any type of smartphones.

// 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 Register New Butterfly

3.2.1.1 Butterfly Table

3.2.1.2 Logs table

3.2.1.3 bfLookup table

3.2.1.4 User Table

3.2.2 User Classification

3.2.2.1 Admin User

1. register new butterfly
2. manage data
3. register new butterfly species
4. Generate Report
5. Export Report to CSV

6. Tag Butterfly

3.2.2.2 Publics User

1. Tag Butterfly

3.2.2.3 Docents User

1. Tag Butterfly

3.3 Performance requirements

Must be accessible from any mobile device or computer. Data must be stored securely. Must be fast and data input ideally can be perform with only one hand

3.4 Design Constraints

Client wants the web application to be used on a tablet like device and smaller handheld device. We have to set the website so it adjust automatically according to the size of the user screen no matter how big or small.

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

3.5.1 Reliability

3.5.2 Security

3.5.3 Maintainability

3.5.4 Portability

3.5.5 Speed

3.6 Other Requirements

// ADD Appendices (if any)

// Regenerate Table of Contents