# Small Pupae Big Data

**sdmay20-01b**

Manthan, Matthew Markose, Mark O'Meara

Advisors: Mohamed Y. Selim, Md Maruf Ahamed, Namrata Vaswani

Client: Reiman Gardens

http://sdmay20-01.sd.ece.iastate.edu/

# Project Overview

# Team Introduction

Manthan: Lead Developer/Project Manager

Matthew Markose: Project Organizer

Mark O’Meara: Project Planner

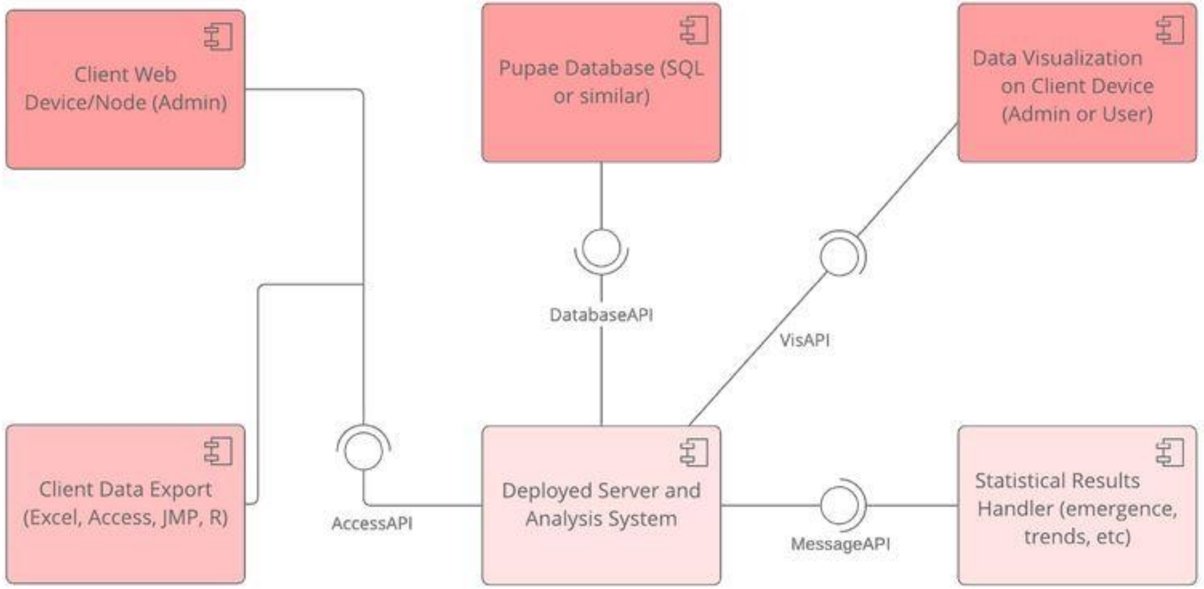
# Project Background

- Client: Reiman Gardens
- Objective: To make a repository for butterfly pupae emergence data
- Existing App: The existing application did not support multiple gardens. It did not have SSL encryption and could not handle large amounts of data to support multiple gardens.
- Issues: Our group was split into two because of internal issues.
- Requirements: Requirements were changed for us to manage the project after the split.

# Problem Statement

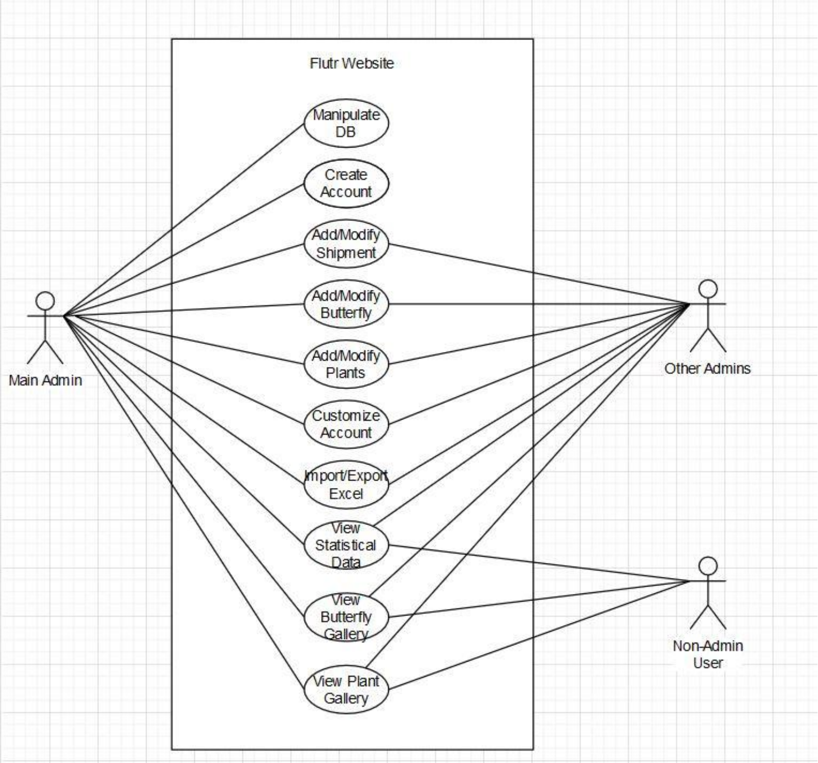
- **Problem**:
  1. Reiman Gardens needs a way to store butterfly emergence data
  2. Current software only supports one garden
  3. New features need to be added for statistical analysis
- **Solution**:
  1. We made a platform for storing emergence data and used a cloud-based database which is meant for big data projects.
  2. The new software will support multiple gardens as authentication was developed using OKTA
  3. New features were added using Vue.js front-end framework and connected to the database

# Conceptual Sketch

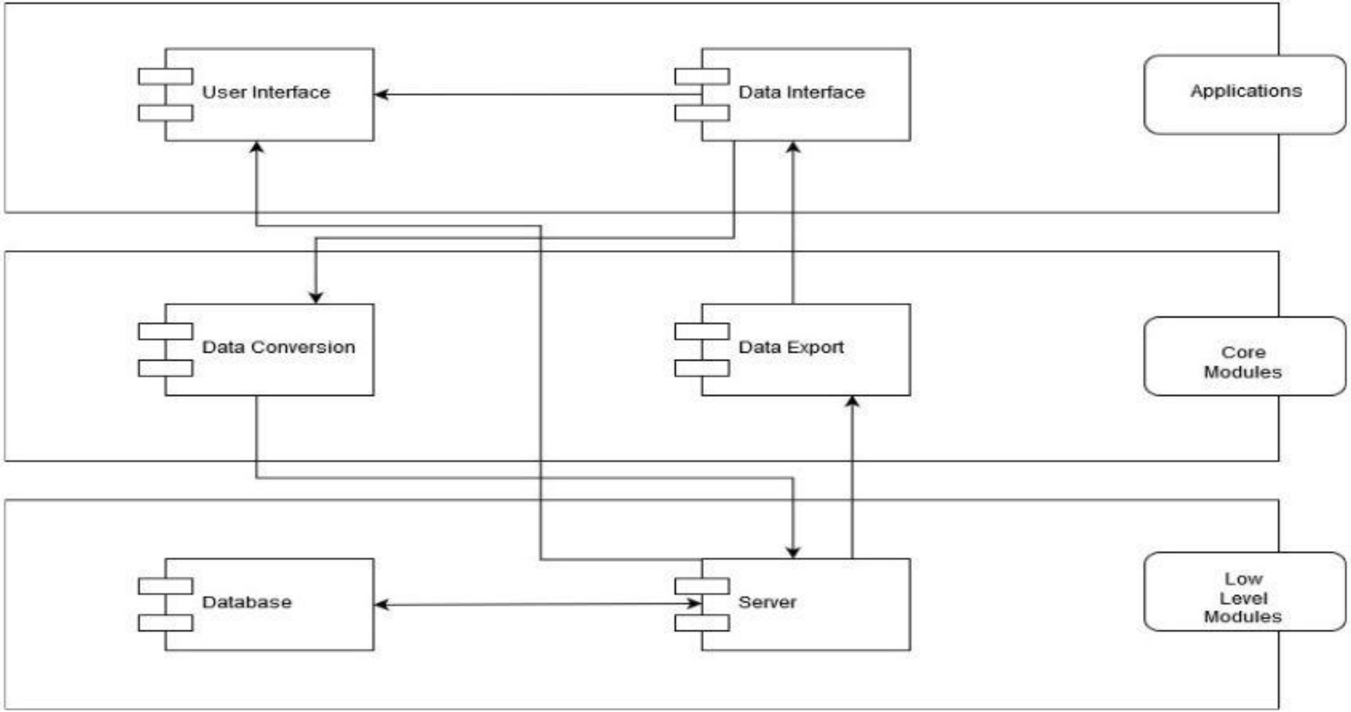

Client Web Device/Node (Admin)
Pupae Database (SQL or similar)
Data Visualization on Client Device (Admin or User)
DatabaseAPI
VisAPI
Client Data Export (Excel, Access, JMP, R)
AccessAPI
Deployed Server and Analysis System
MessageAPI
Statistical Results Handler (emergence, trends, etc)

# Use Cases

- 3 actors
  - Main Admin
    - Create/Manage Accounts
    - Modify Database
    - View Gallery
  - Other Admins
    - Modify Database
    - View Gallery
  - Non-Admin User
    - View Gallery



# Module Diagram



# System Design

# Functional Requirements

- Excel Import/Export
- Butterfly/Plant gallery
- Multiple Users
- Butterfly Search
- Statistics page
- Admin Note and fact of the day

# Non-Functional Requirements

- Admin user should be able to change their background
- Ability of change emergence data using button instead of drop downs
- Butterfly Showcase on front page
- Kiosk View
- Ability to modify plant gallery
- Shipment Search

# What makes our project unique?

- Use of cloud-based database
- Multiple user implementation
- Secure authentication using OKTA
- Customization for each user
- Streamlined data input

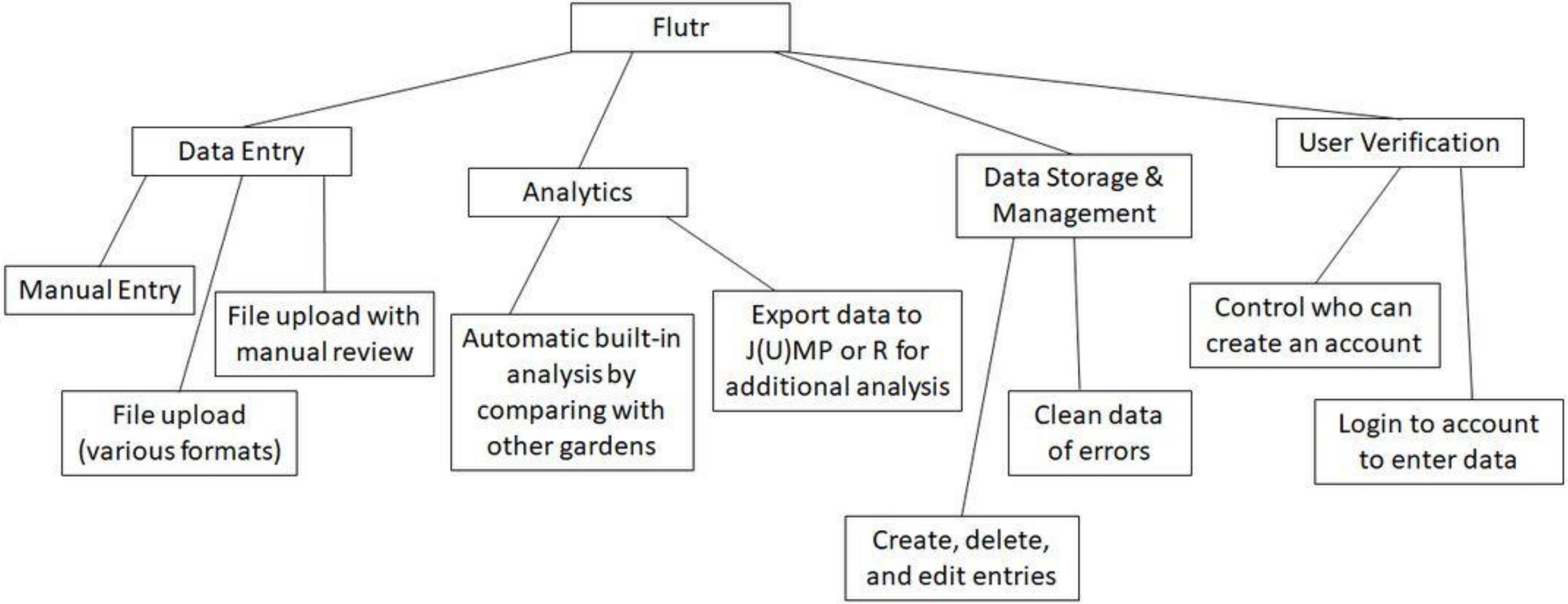
# Resources

- Firebase
  - Firebase gives free data for developers
- OKTA Authentication Framework
- Developers
  - Manthan
  - Matthew Markose
  - Mark O’Meara

# Risks & Mitigation

- Security Risks
  - Unauthorized Login – We have a secure industry trusted framework in place
  - Data Leak – Ajax requests to the database could possibly be interrupted, firebase cloud functions provide a secure channel to prevent that
- App slowdown because of large amounts of data processing
  - Use of firebase cloud database will ensure all data is processed on a remote computer
  - Possible statistical glitches would also be mitigated because of firebase

# Functional Decomposition



Flutr

Data Entry

Manual Entry

File upload (various formats)

File upload with manual review

Analytics

Automatic built-in analysis by comparing with other gardens

Export data to J(U)MP or R for additional analysis

Data Storage & Management

Create, delete, and edit entries

Clean data of errors

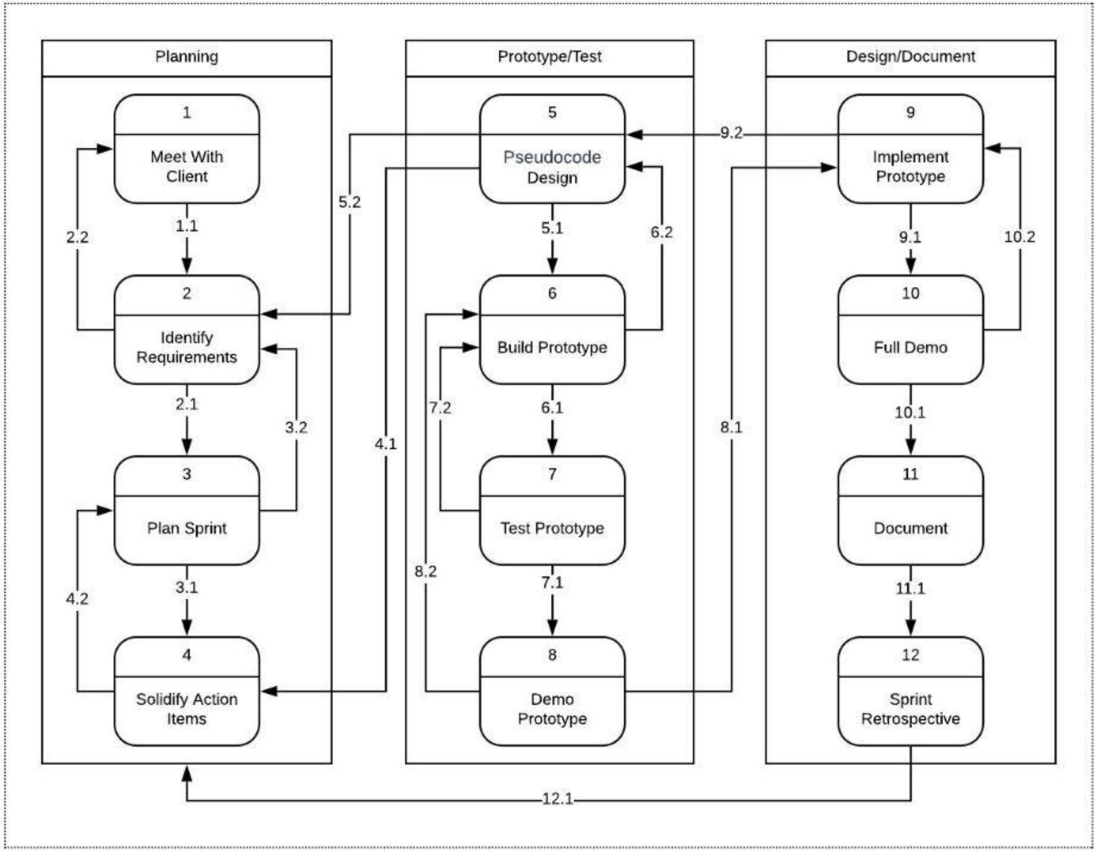
User Verification

Control who can create an account

Login to account to enter data

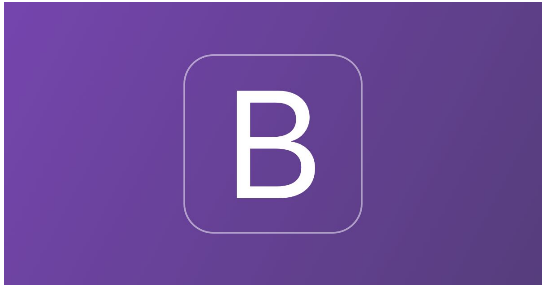
# Design in Detail

# Design Process



# Technology Used

node JS ™
Node js Development



# Design Decisions

- Cloud vs Local Storage
    - Existing site uses local storage
    - Firebase is cloud based storage that is free for a certain amount
    - Firebase is usually used for projects involving big data
- Vue.js
    - New framework with long term support
    - Simple and easy to use front-end framework
    - Based on JavaScript and easy to use with Firebase
- Firebase vs Amazon Web Services
    - Firebase is easier to integrate with Vue.js
    - Firebase has a free plan for developers
    - AWS requires you to pay from the beginning
- Node
    - Vue.js just uses Node by default
    - Node provides a lot of useful plugins like excel import/export plugin

# Testing

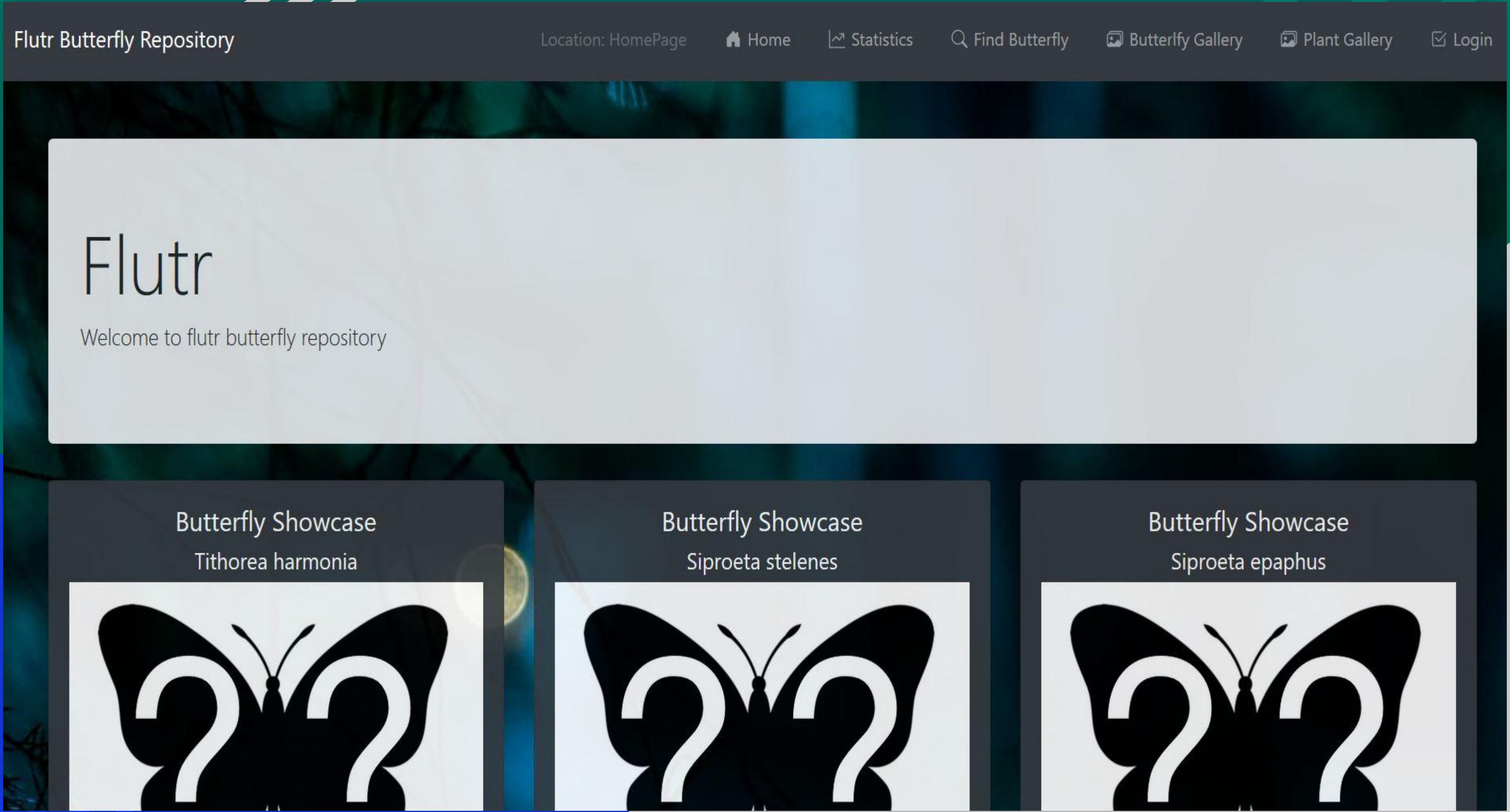
# Initial Test Plan

- Unit testing with built-in framework
- Work with Jest
- Utilize Nielsen and Molich’s User Interface guidelines
- Manual testing to match up with guidelines
- Hybrid/ Sandwich approach for integration testing

# Things we learned from this project

- To fulfill the requirements of the project we needed to a learn a cloud-based database
- Other new frameworks such as Vue.js and OKTA API
- Team-work in agile environment
- Getting requirements from the client with face to face interactions
- Designing a software from the ground-up with implementing a design document and working on the design process

# Demo



# Flutr

Welcome to flutr butterfly repository

## Butterfly Showcase

Tithorea harmonia

## Butterfly Showcase

Siproeta stelenes

## Butterfly Showcase

Siproeta epaphus

# Summary

- Created an application for storing and analyzing butterfly pupae emergence data
- Built with Vue.js, Firebase and Node.js
- Most of the requirements were fulfilled besides the requirements that were changed based on team split.
- Main purpose of our application is complete although improvements can always be made

# Engineering Standards and Design Practices

- Hardware is minimal in purpose, mainly a client facing web device and temperature/environment probe to create a standard of easy access and analysis
- Software practices planned are for easy adaptability for multiple potential clients, thorough data analysis presented in a simple way for clients, and cohesive design.
- Standards: all were considered; mainly ethics for pupae growth and competition

# Thank you