Unified Captive Reared Caterpillar Tracker

Design Document

Team Number:
sdmay24-04

Client:
Nathan Brockman - Reiman Gardens

Advisors:
Dr. Shana Moothedath, Xinyao Li

Team/Roles:

Michael Gradle	-	Client Interaction / Frontend
Ricky Smith	-	Testing / Backend
Rose Druce-Hoffman	-	Scribe / Backend
Jonah Besselievre	-	Design Lead / Frontend
Gabriel Owen	-	Testing / Frontend
Kristen Hawken	-	Database Admin / Backend

Team Email:
sdmay24-04@iastate.edu

Team Website:
https://sdmay24-04.sd.ece.iastate.edu

Revised: 12/03/2023

Executive Summary

Development Standards & Practices Used

- Agile Development Process: Software
- MVC Software Design Pattern

Summary of Requirements

The main goal is to create an application that helps record data on caterpillars and butterflies.

- Must be usable on smartphones, tablets, and laptops
- Can automatically retrieve data such as temperature, weather, and dates
- User friendly
- Easy export of data

Applicable Courses from Iowa State University Curriculum

COM S 309 is very applicable due to its focus on creating an app for both the frontend and backend.

COM S 363 is applicable due to teaching us how to use databases.

COM S 227/228 is applicable for various introductory programming knowledge

COM S 311 is applicable in case we need any more advanced algorithms

S E 329 is applicable due to teaching us how to manage software productions.

New Skills/Knowledge acquired that was not taught in courses

Gained ReactJS and NodeJS skills which are very prevalent in industry projects

Learned various skills for maintaining a positive client relationship

Gained knowledge about hosting a website (AWS)

Table of Contents

1 Team, Problem Statement, Requirements, and Engineering Standards

1.1 Team Members

Jonah Besselievre, Rose Druce-Hoffman, Michael Gradle, Kristen Hawken, Gabriel Owen, and Ricky Smith

1.2 Required Skill Sets for Your Project

1. Database Management
2. Web Design - HTML, CSS
3. Backend Development - NodeJS
4. Project Management
5. Frontend Module Development - ReactJS

1.3 Skill Sets covered by the Team

Michael Gradle - Frontend Development, Web APIs (C#, NodeJS, Java)

Ricky Smith - Backend Development (Primarily Spring Boot, NodeJS),

Agile Development Familiarity, Web Design

Kristen Hawken - Backend Development, Database Management, Agile process familiarity

Joanna Besselievre - Frontend Development, Agile process familiarity, App design

Gabriel Owen - Frontend Development, Backend Development, Web Design

Rose Druce-Hoffman - Frontend development, Agile Development familiarity

1.4 Project Management Style Adopted by the team

Agile Methodology: combination of Scrum and Waterfall Models

1.5 Initial Project Management Roles

Michael Gradle	-	Client Interaction - Frontend
Ricky Smith	-	Testing - Backend
Rose Druce-Hoffman	-	Scribe - Backend
Joanna Besselievre	-	Design Lead - Frontend
Gabriel Owen	-	Testing - Frontend
Kristen Hawken	-	Database Admin - Backend

1.6 Problem Statement

The current way our client tracks caterpillar data involves collecting data on paper, which is then stored on spreadsheets. It is a slow, difficult process. Our clients want a more efficient way of collecting and analyzing data.

1.7 Requirements & Constraints

- Application should be usable on both mobile and desktop; no clunky UI.
- Text should be minimal and accessible for all users.
- Gamified program that leaves users feeling rewarded for contributing data.
- Easy to find and use toggles, for opting in/out of shared data or seeing global platform collections.
- Letting the user decide what they want on their screen.
- Faster than hand-tracked data (Excel sheets, paper documents, etc.)
- Useable on a variety of devices
- Scalability
- Gather data automatically whenever possible (APIs)
- Must be able to scale well with large amounts of data
- Migration should be viable
- Purchase and integration of different sensors for tracking environment
- Entirety of service must be relatively cheap (<$30 Monthly)

1.8 Engineering Standards

Programming Languages: NodeJS, ReactJS, HTML and CSS

1.9 Intended Users and Uses

Reiman Gardens is our client who will use the application to capture data about caterpillars and butterflies. The users will be the researchers in the field collecting data and the scientists analyzing said data. Our client wants our application to be general enough that it can be used nationwide for multiple species. Beyond just Reiman Gardens, our users are researchers all over the country. A notable example of this is the Oregon Zoo.

2 Project Plan

2.1 Task Decomposition

UI/UX Development

Develop the app initially on mobile in an intuitive way that looks good. The client specifically requested that the app is intuitive, easy to use, and not confusing.

Front End Development

Develop different modular intricate functionalities of the app (tables, separation of windows/scrolling, etc.). Along with making it easy to use, the app will provide many features that will make data collection easier (such as auto-populating fields).

Front End Data Collection

Develop a system for creating records. This is the main focus for the app; users must be able to easily collect data from butterflies in all stages of their life cycle.

Back End Data Storage

Fix back end server to store the data gathered from collection. Data collected shouldn't be static and users should be able to look back.

Data Reporting

Create/use existing data reporting methods for a variety of reports to be used by the client, as well as a method to make data turn into CSV. Not only is the data reporting a major part of the app's use, but converting data to CSV allows it to be used in other applications should a transfer be needed.

Confirm Format Works on Laptop/PC Browsers

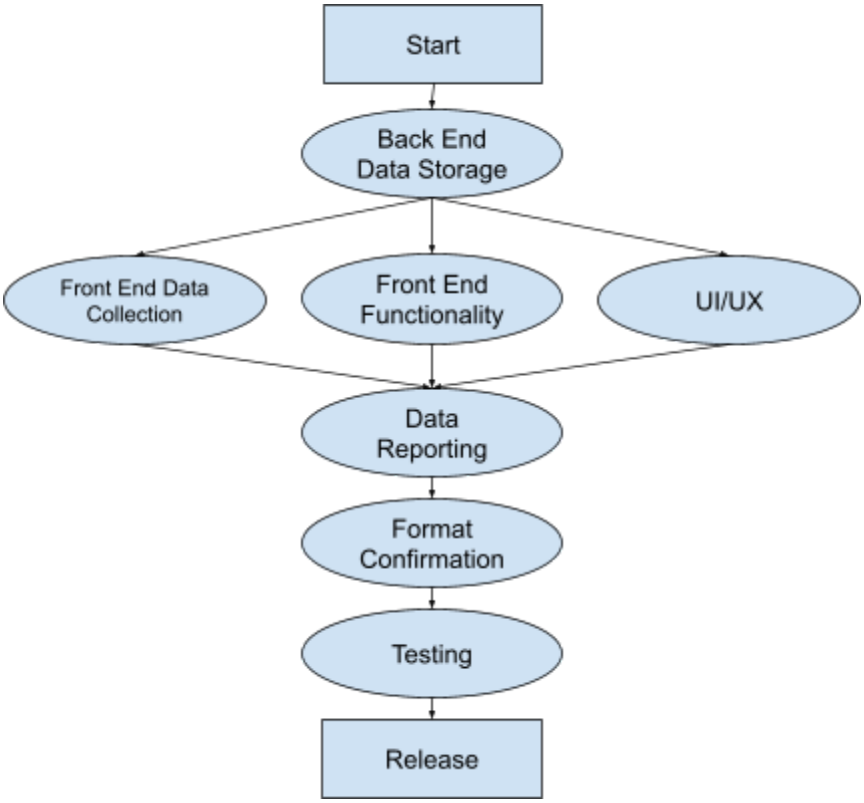
This task involves checking all of the UI/UX on a Laptop/PC and making sure everything looks good. An important step to do before an initial testing phase as we're going to be mainly developing it for use on mobile devices.

Testing

Once functionality is complete, testing must be done to ensure the application will work. The client has assured that they will test it out at their facility. When issues come up during testing, new tasks will be created to fix them.

Release/Finish

A task to release the application in a form that is easily accessible and manageable by the client. Necessary for a smooth hand-off of the app.



2.2 Project Management/Tracking Procedures

We will be following Agile Methodology, and using a combination of Waterfall and Scrum models. We will follow two week sprints and utilize a Kanban board. We will use Git's built in board section to split up and track tasks. Our board will have Backlog, Next Sprint, Current Sprint, Review, and Done columns. We will meet weekly to connect and stay on the same page.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Front End Development

We have a functioning navigable app. All windows are clear to see and use. It looks good on both mobile and desktop. Our client approves of the design.

Front End Data Collection

You can create and populate records faster than when done by hand. It is intuitive to fill in data. Our client approves of the process.

Back End Data Storage

Our data can be found exactly where it's expected to be. Users are able to pull in back end data and view it in a human readable format.

Data Reporting

Our data can be exported to CSV files. Our users can click on a record and see relevant data easily. Our users can choose reports which have certain parameters to output a PDF of organized data for easy visualization. Our client approves of the volume of reports as well as the reporting layout.

Confirm Format Works on Laptop/PC Browsers

The app can be used on both mobile and desktop without looking strange. Buttons should work as intended, and dimensions should look somewhat normal.

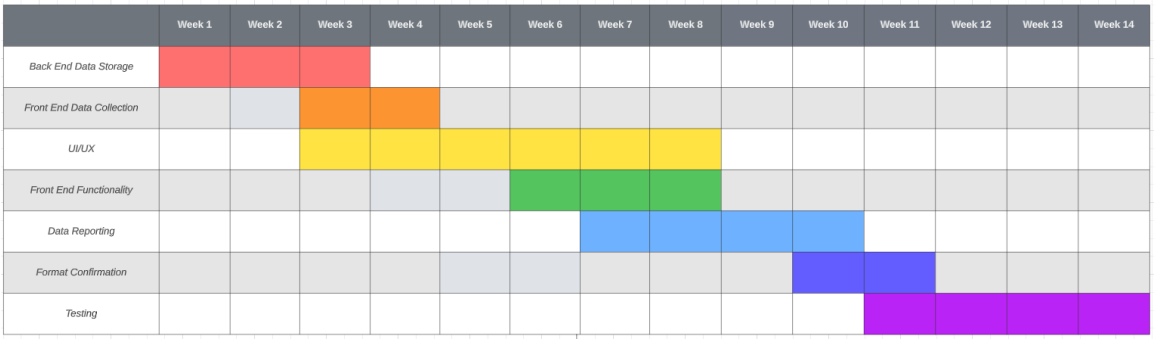
Testing Phase

We feel satisfied having conducted our own tests. We have given users a couple of weeks to use our project to understand what can be improved. Our client approves of a final product.

Release/Finish

We have successfully published the program. Our client has fully taken ownership of the project and understands how to manage it.

2.4 Project Timeline/Schedule



Deliverables:

Back End Data Storage will be delivered at the end of Week 3.

Front End Data Collection will be delivered at the end of Week 4.

UI/UX will be delivered at the end of Week 8.

Front End Functionality will be delivered at the end of Week 8.

Data Reporting will be delivered at the end of Week 10.

Format Confirmation will be delivered at the end of Week 11.

Testing will be delivered at the end of Week 14.

2.5 Risks And Risk Management/Mitigation

UI/UX Development:

- Users do not find the design to be intuitive and have lots of questions. We are working very closely with our client, but we can't anticipate what it's like using it in the field. Risk: 0.3
- Task takes longer than expected. Likely due to the nature of UI/UX, but we've allocated plenty of time for it. Risk: 0.4

Front End Development

- Functionalities do not work as intended. Risk: 0.8 - this risk is mitigated intrinsically through the testing phase.
- Users are confused about how the functionalities work. Risk: 0.3
- Task takes longer than expected. Unlikely due to the priority we've given it. Risk: 0.4

Front End Data Collection

- It is not faster than doing it by hand on paper. There is a chance that the initial learning curve actually makes the collection process slower. Risk: 0.4
- Data auto-populates incorrectly and users have to do many manual overrides. Our client is very excited about this but there is always potential it harms more than helps. Risk: 0.2

Back End Data Storage

- Records get completely wiped. This is pretty much the worst-case scenario, as data storage is the main focus of the app. Risk: 0.6
    - To mitigate this risk, backups might need to be implemented to ensure that if a data wipe occurs, the data is still safe. This can be through offline data reporting, storing the data in an offline storage, storing the data in another online database, et cetera.
- There is no more room for additional records. This can happen by running the commercial database's space out, so we'll have to choose a database that can allow for the amount of data we're storing. Risk: 0.4

Data Reporting

- We cannot export our data into CSV files. This should be simple but who knows. Risk: 0.1
- Our users do not find the data reports to be meaningful. We will have to make sure we get all of our client's priorities straight. Risk: 0.3

Confirm Format Works on Laptop/PC Browsers:

- The UI looks funny on the computer. This is somewhat unlikely as it will probably appear exactly as it does on the phone and have blank space on the sides. Risk: 0.2
- Task takes longer than expected. This depends on how smoothly the UI/UX development goes, and how easy it is to format what we use on a computer's browser. Risk: 0.5 -
    - We should consider finding libraries during UI/UX development to help with dynamic html formatting based on how big the window is. If it does take longer, we may want to find an alternative or repurpose the app for the computer.

Testing:

- Functionality does not work as expected (repeatable). This is the intended purpose for testing, though, so there's not much of a need for mitigation. Risk: 0.99
- Task takes longer than expected. Unlikely given the amount of time we've assigned to it. Risk: 0.4

2.6 Personnel Effort Requirements

Each group member will be expected to exert an equivalent of 1/6 of the person-hours given for each task.

Task	Person-hours (Total Hours for Entire Group)
Back End Data Storage	84
Front End Data Collection	30
Front End UI/UX	118
Front End Functionality	42
Data Reporting	78
Format Confirmation	36
Testing	126

Total: 504

2.7 Other Resource Requirements

Once our app is fully developed and tested, we will need to find a hosting service to keep the app up and running. If possible, we will use a free service, if not, we will find a service for less than $30 a month.

(Potential) If we find an external device that will aid in data collection and is a reasonable price. We will buy it and provide it to the client upon our graduation.

3 Design

3.1 Design Content

Frontend:

- UI that is not too busy and easy to navigate
- Keep unnecessary information hidden
- Allow for use on both mobile and desktop
- Each screen on the app will be its own module
- Keep track of specific caterpillars/butterflies as data Objects

Current UI ideas can be found in section 4.7.1

Backend:

- Limit continuous data accesses for cost efficiency
- Create a separate private and public database
- Most data will be stored as JSON objects

3.2 Design Complexity

1. The problem scope is outside typical project standards. While trackers are common in the market, there aren't many that are specially designed for butterfly care.
2. In making the design, we will be using new technology such as SSMS
3. The project includes creating a professional grade app that is on par with applications already on the market
4. The finished product requires compilation of many different components with respect to functionalities for each of the inputs, as well as Reporting, UI/UX, etc.

3.3 Modern Engineering Tools

For our project, we will be using ReactJS with NodeJS and Express.js to build our application. NodeJS and Express.js will be used to manage the backend routes our application will need to access the database. ReactJS will handle all of the frontend UI that the user will see.

Various APIs will be used when necessary. For example, we plan on using Google Earth/Maps API to gather location data.

3.4 Design Context

Area	Description	Examples
Public health, safety, and welfare	The main stakeholder group for this project is the many groups around the country raising and tracking caterpillars and butterflies. By creating a digitized and centralized system for collecting data, these groups will have a faster, easier to use, and more reliable system than they had before.	There may be a chance that job opportunities may be reduced since they do not need as many people to record data. However, it will free them up to complete other important tasks.
Global, cultural, and social	Taking care of butterflies can be an important societal cornerstone. The health of butterflies can be an indicator on the health of the earth itself, which is important seeing as people live on it. Our project will help in monitoring this health. Our project aims to support the practices of the groups and workplaces that will be using it. Our project is intended to be a digitized version of their current system, improving on what they currently use and centralizing systems across the nation.	There may be a minimal change to existing processes due to this project. Each organization around the nation collects data slightly differently, so by centralizing this data collection some groups may have to change their processes slightly.
Environmental	Our project will help to support the environment by supporting butterfly conservation programs. Our project will be used by butterfly researchers to track caterpillars and butterflies. This will help them to understand more about their life cycle, and in turn work to improve conditions both in the environment and in captivity to improve the number of butterflies raised.	Our project has more benefit to the environment than harm. While there is some energy use that could have been derived from nonrenewable sources, overall these are negligible. Instead, our project is directly contributing to butterfly conservation.
Economic	While not directly helping in an economic way, the project will help determine what is the best conditions for taking care of the butterflies. This will prove useful when making changes to the settings of the butterflies and determining whether the cost of changing an aspect of butterfly care is worth it.	App can help determine whether a new lamp benefits or harms the butterflies; can be a deciding factor on if moving butterflies' location to another building is worth it by monitoring health of already moved butterflies

3.5 Prior Work/Solutions

The current system used by butterfly conservation groups is recording data on data sheets by hand on paper, which is then entered into Microsoft Excel spreadsheets. Each group also currently collects different amounts and types of data on the caterpillars and butterflies they raise. The aim of our project is to improve these current systems.

The advantages of our project over current systems are many. First, it is digital. We are providing a centralized location for researchers to enter data into directly. As long as they have a device and login information, they will be able to enter and/or view any data from their organization. They will not have to worry about losing paper sheets or entering in data onto the computer by hand. In addition, this data will be stored in one database instead of multiple Excel files. This will allow users to access and use the data much more easily, especially when comparing different data sets. Also, each organization across the country records slightly different amounts and types of data. Our system will maximize the amount of different types of measurements that can be taken. While each group can choose which fields they want to focus on, they will be encouraged to take as much data as possible. Finally, with one centralized system being used by many groups around the nation, they will be able to compare data easily with one another.

3.6 Design Decisions

Physical Layout - After taking feedback and suggestions from our client, and from prior reference of the other butterfly reporting app(UBT), which our project is a complement to. We are designing the app in a manner that the client feels most effective for data reporting and showing historical user created forms as well as the biggest concern of keeping the user engaged and feel that they can easily navigate and use the app. Our layout is strongly inspired by the UBT app which has received a lot of praise from the client, but also accounts for additions specifically required by our project and extra desired features. While design is still an ongoing and continually updated process we are confident that we have a good starting point/reference for our initial layout

Features - Since our web app will be used most likely in the field, and for many of the forms requiring field related data. We have found it important that we allow for auto-obtained weather/location information, photo-metadata, and etc, so the user can have a much faster and easier time inputting information into fields. However this data may not always be accessible or wanted by the user so we are figuring out ways to allow the user to selectively populate or choose fields to ignore from auto collection depending on the situation. As well as options to augment data to represent their preferences, eg. celsius or fahrenheit, GPS coordinates or general location description.

Subsystems - 3rd party API integration is still a key design component we are configuring for our application. Most likely we will need to pull data from weather API's, google maps, and possibly other butterfly/caterpillar research organizations to optimize our clients requests. We are attempting to find a good choice between our backend and data storage solutions to best integrate this additional data.

3.7 Proposed Design

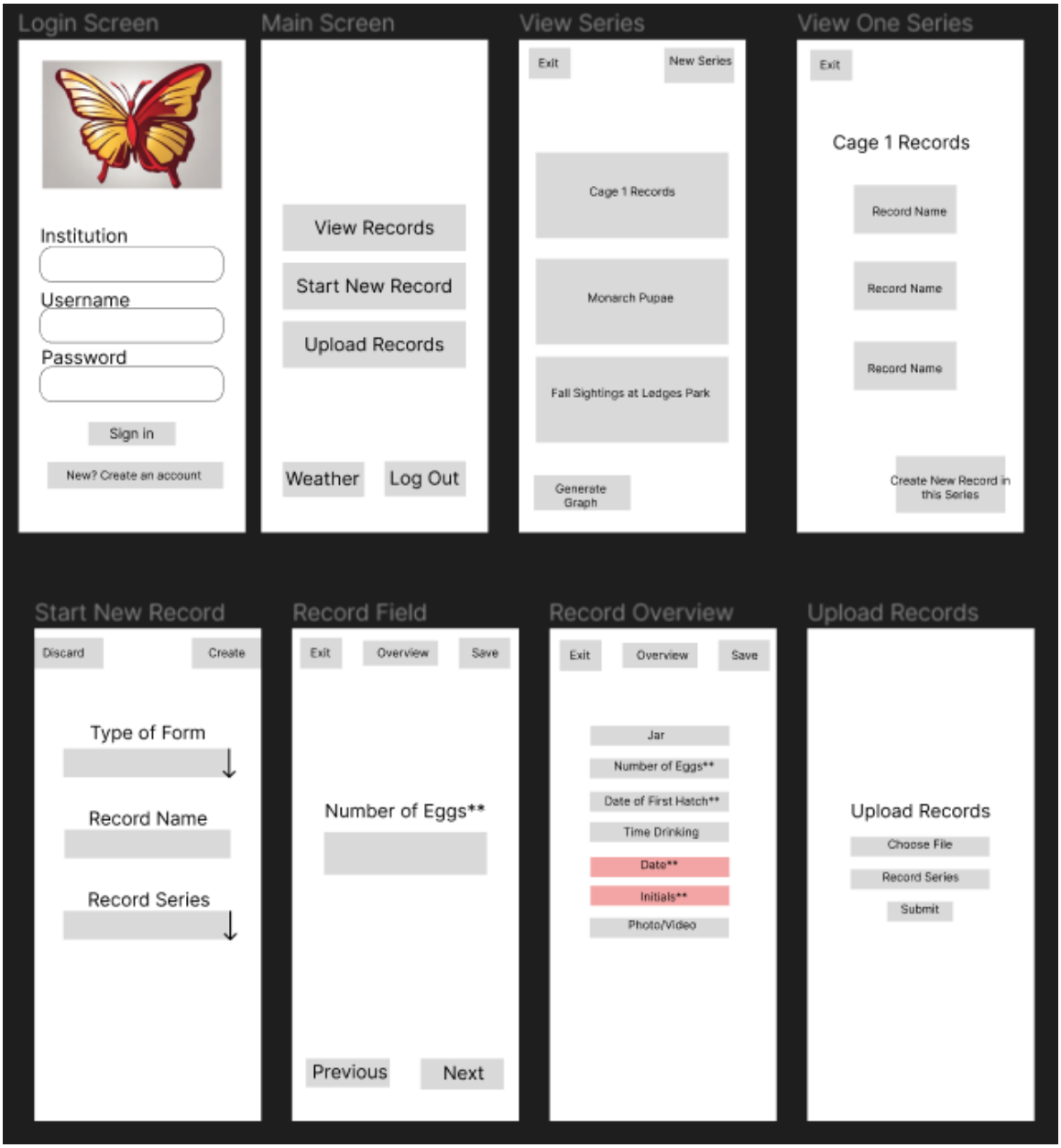
So far, we have implemented Figma as our wireframe creation software and decided on React and Node.js for our respective frontend and backend languages. We plan on using a SQL server with SSMS for the database, but have yet to find a good solution to a cheap hosting service that will allow as many requests as we'd like to be able to handle long-term. For reporting, we are waiting until we get to the point where we have data in our design process to choose which software to use.

Implementation of our actual software has not begun yet, but we have started making wireframes for our mobile web app on Figma. This wireframe includes simple modular pages that we plan on using as the basis for further development.

3.7.1 Design 0 (Initial Design)

Design Visual and Description

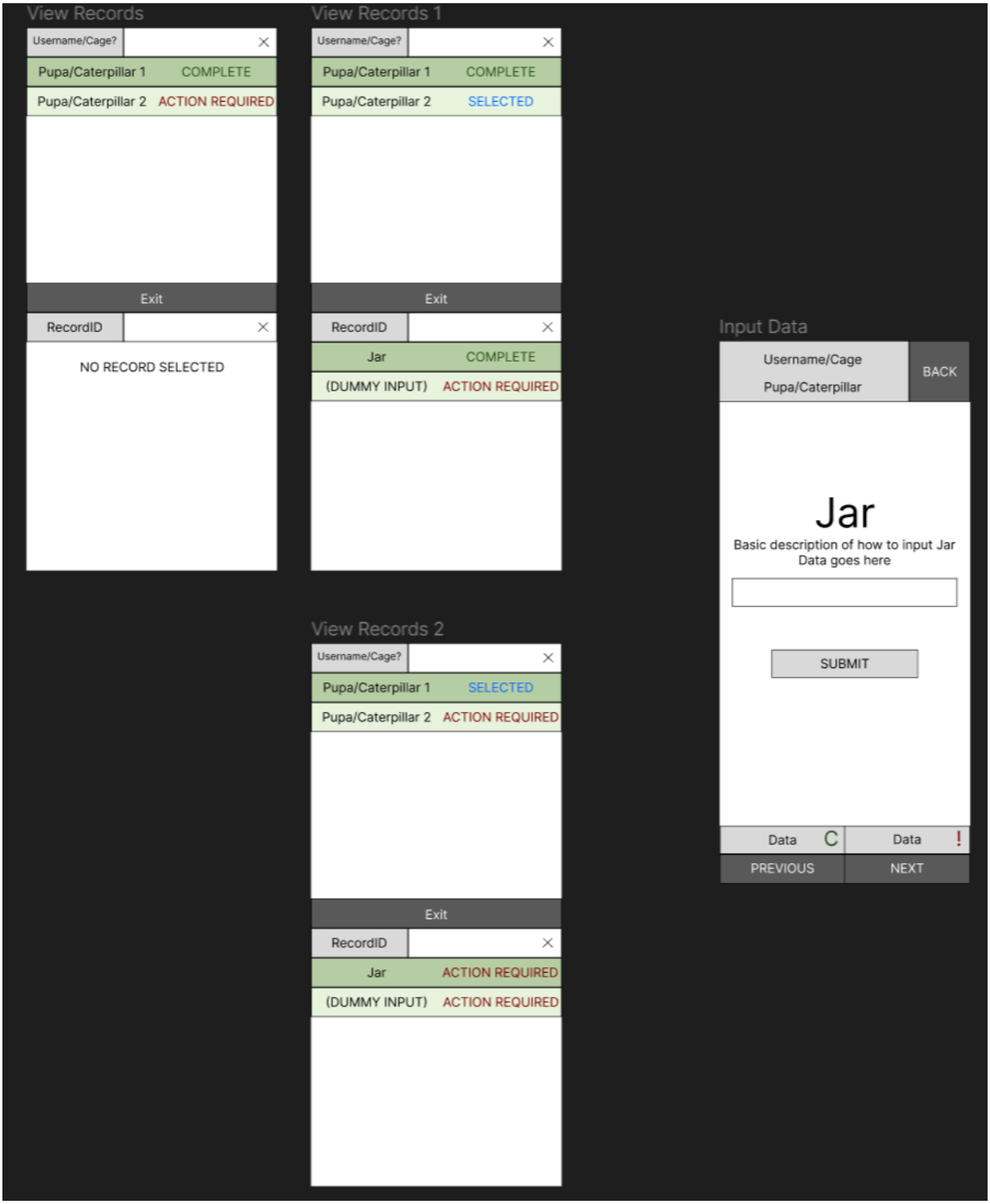
Visuals of our simple UI foundation wireframes are below.



This visual shows a basic path that a user would take to input information. The **Login** screen covers the “Useable on a variety of devices” requirement.

The **Upload Record** screen allows for more easy Migration of data. The **Main** screen lets the user decide what they want on their screen - the **View series** screen does the same.

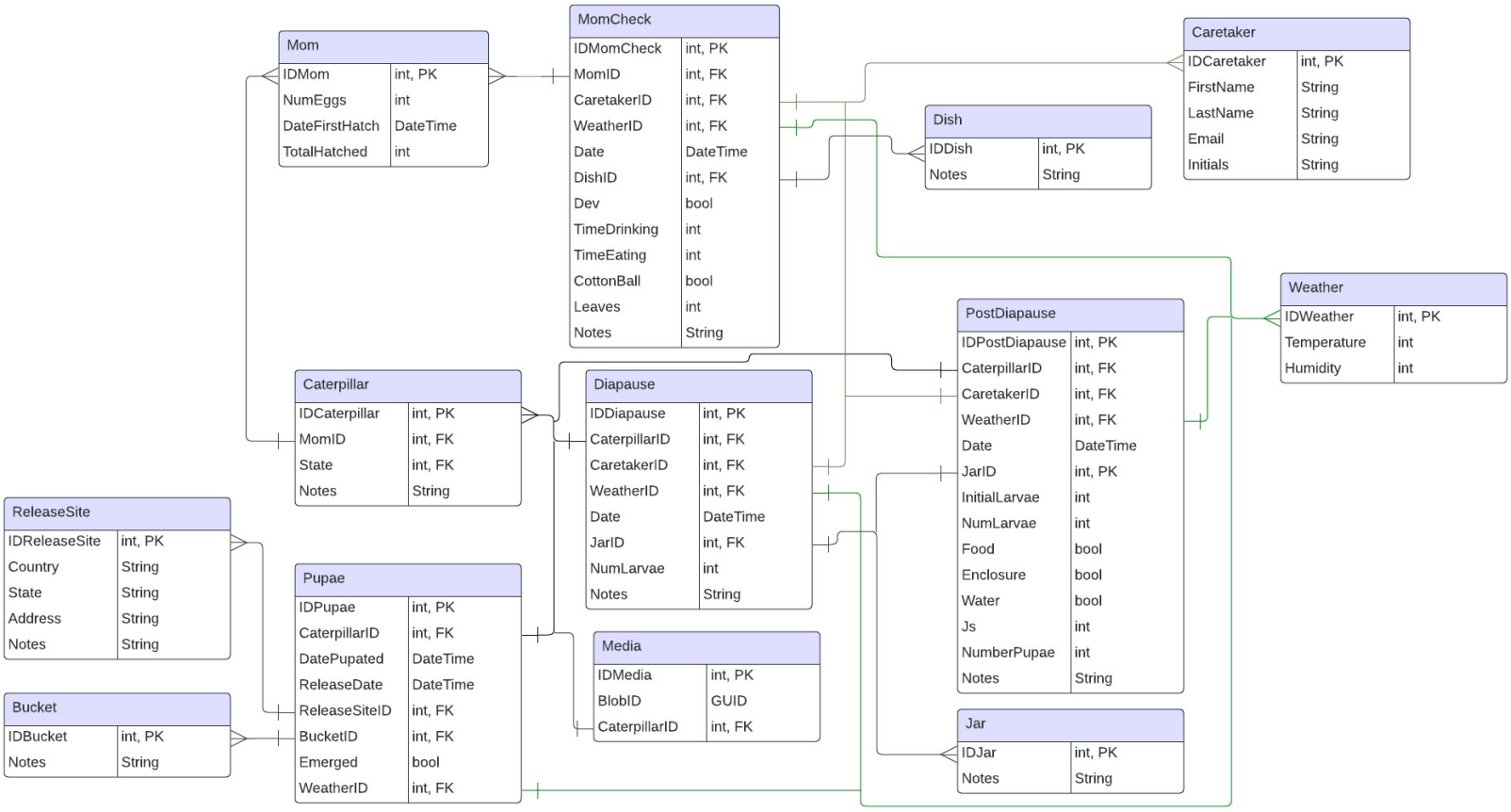
View One Series, **Record Field**, and **Record Overview** are screens that have been scrapped and replaced by the screens on the visual below. The **View Records** and **Input Data** screens cover a few



requirements - they are designed to be usable on both mobile and desktop, with minimal text, let the user decide what they want on their screen, should be faster than hand tracked data, will gather

data automatically when possible, and scale well with large amounts of inputs. On top of that, the colored statuses are meant to make it feel gamified.

Below is the ER diagram for the design's initial database setup, which is subject to change.



"Caterpillars" and "Moms" are entities that have events connected to them in the form of logs that are meant to happen on a recurring basis. There is one caterpillar/butterfly per **Caterpillar**/**Mom** entry, but there can be many entries for the **MomCheck**, **Diapause**, and **PostDiapause** tables. For reporting purposes, it's unlikely that this database setup will create any substantial reports for analysis, as well as the aspect of users/organizations that we plan on resolving with feedback from the client during the testing phase.

The goal of these vague foundations is to promote close collaboration with the client. It is also meant to contribute to giving our client the final product that they would desire more so than anything - it is impossible for anyone to perfectly convey exactly what a final product will look like, just as it is impossible to create a final product with a single design.

Functionality

Researchers will be using our application in multiple settings. They can open it on their phone to track sightings, or they can enter information on their computer in the lab. With that information, they will be able to print/download reports that they can use for data analysis, among other things. Based on the user's organization/standing within their organization, they may be able to utilize administrative privileges such as creating new accounts. Non-admin users will be able to perform actions that align with either default settings or settings set by an administrator.

Our design satisfies the functional requirements as well as all of our client's needs. It's set up for tracking and recording a variety of different things for different life stages. Additionally, the non-functional requirements of ease of use and efficiency are met.

3.8 Technology Considerations

The technology available to our group, be it by the client or by Iowa State, includes: a server, git, current recording devices used by the client to possibly link up with our application, and setups capable of coding the webapp.

With the server technology available to us, we can easily make a database to store the information needed to track. It would be a lot better than creating a database from scratch and buying the space needed to host the information. However, getting a database from Iowa State will make it a process to transition to an alternative once the work is mostly done, which we will need to account for. An alternative would be to set up a database that is not on Iowa State servers.

Git is something that we have all worked with and will allow us to set up a repository and CI/CD for easy access and storage of our work. We see only pros to working with git as opposed to alternatives we can think up.

Utilization of recording devices for certain data points has advantages in fulfilling requirements of our process being faster than paper, but may end up costing more than we can handle.

Technology for actually coding/implementing our project such as PCs or Laptops have some pros and cons depending on what from ISU is being used, but will not have a lasting outcome of our final product.

3.9 Design Analysis

There are a few things to consider with our current design - some have already been mentioned. Intrinsically with our current design we're leaning towards creation of an iterative week-by-week product that will get better based on what we put out and the feedback we receive. It would be beneficial for us to put out a live testing site for our client and receive feedback as much and as quickly as possible for this reason.

Another thing is that while we have a good foundation for the UI and Backend, we lack any sort of baseline for creating an app in React and Node.js. Our design is intentionally lacking in this area to give us an incentive to learn more about the software we'll be using. This may help with API integrations as we'll be able to better understand and select APIs that work well with our code, but it also creates a lot of room for error.

Overall, our design is a solid foundation to create a product with.

4 Testing

4.1 Unit Testing

We plan on testing the login process, a few random instances of every modular functionality we create, and each eventual complicated functionality that we create for different input fields.

Modular functionality includes:

- Input fields
- Tables
- Buttons
- Reports
- Requests

How:

- Create test scripts that test different parts of the app
- For backend, test routes using scripts
- If we find it not too difficult, use Selenium and ChromeDriver to test website

Tools:

- Frontend: Jest, Selenium
- Backend: Jest

We'll be using Jest for our Unit testing.

4.2 Interface Testing

User Interface:
Combination of the login process, buttons, input fields, tables, and complicated input functionality.

Backend Interface:

The set of requests (API or to our own db) that are used to access a database

Reporting Interface (Subset of UI + Backend):

Combination of UI for getting reports and requests to database that provide data for reports.

We'll be using Jest for our Unit testing.

4.3 Integration Testing

There are many critical integration paths. One is inputting data, creating a new record, and storing the record in the database. This path flows from the UI to the database. Another is calling to view a record, retrieving the record from the database, and displaying it on the screen. Logging in is also an integration path, the user inputs their information, the server is contacted to find their account, and their data is retrieved from the database.

We plan to test this by using the system as a test user and checking each stage of each path to ensure the correct responses are being received and sent. Our test users will be both us and other people not familiar with the app, such as employees from Reiman Gardens.

4.4 System Testing

When pushing updates to live, we'll run a smoke test after it restarts in the pipeline to make sure it's running. After the smoke test, we'll run through all of the critical integration paths and a random set of combined UI/Backend unit and interface tests. This will be done through a combination of a CI/CD pipeline and Jest.

4.5 Regression Testing

As we add new features, we will test them to make sure they do not break old functionality. We will test new features as we are adding them, as well as running more comprehensive tests of new and old features when they are implemented. The most important feature is inputting data about the caterpillars. This needs to be functional at all times as the users will be relying on our application while doing their daily data recording. Our first priority when adding new features is to test the data input feature. This involves testing the UI as well as whether the correct data is received in the database. From there, we will test other previously implemented features.

4.6 Acceptance Testing

Once our tests are completed, we will show the client we have met the requirements through a practical demonstration. They will be able to see that we have... Then, with the demonstration done, we will hand the alpha version of our application to our client so they can test it out.

Through their rigorous use, they will be able to find corner cases and bugs that we can iron out.

4.7 Results

At the conclusion of our testing, our system should work as intended. Data inputted on the frontend should be stored in the backend and data requested from the database should be displayed on the frontend without any changes or problems. New features added over time should not cause problems in previously implemented features.

Test	Correct	Incorrect
React - Frontend	Easily Navigable Simple/uncluttered Easy to See	Cluttered Too many components non-uniformity Slow Difficult to understand
NodeJS - Backend	Quick Provides Correct Data Inputs Correct Data	Slow Incorrect data returned Inputs don't match save in DB Inputs don't save in DB
Smoke Test - Reporting interface	System is stable, little to no loss of uptime, and can be navigated through multiple steps with no “leaks”, errors.	System is not stable crashes frequently errors when navigating thru-multiple system steps

5 Implementation

1. Set up project environment
2. Populate Git board and assign issues
3. Create UI
    a. Develop frontend modules
    b. Create fields and space for lists
    c. Develop data collection process
4. Create Backend
    a. Create database and tables
    b. Link database to frontend logic
5. Develop round trip communication between frontend and backend

6. Integrate 3rd party API
    a. Weather data
    b. Location data
7. Create data reporting
    a. Determine criteria to report on
    b. Develop method to export data
    c. Create method to export to CSV
8. Testing
    a. Unit testing on modules
    b. System testing on communication and interaction between parts
    c. Ensure format works on computer browsers

6 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

6.1 Areas of Responsibility

Area of Responsibility	Definition	NSPE Canon	SE Code of Ethics
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence	Perform services only in areas of their competence; Avoid deceptive acts.	Ensure that you are suitably qualified, which is similar to NSPE canon. Promote maximum quality, which is similar to the definition.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	Promote minimum cost and ensure all estimates are realistic relate to the definition. They are similar to the NSPE canon but NPSE isn't as specific.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	Be fair and truthful in all manners applies to the NPSE canon and the definition. Also, be accurate and honest regarding software.
Health, Safe, Well-Being	Minimize risks to safety,	Hold paramount the	Always put the public's

	health, and well-being of stakeholders.	safety, health, and welfare of the public.	interests first overlaps with both the NPSE canon and the definition.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	Only use property as authorized by the owner is part of the definition under respecting property of others. The NPSE canon is more general but still relates.
Sustainability	Protect environment and natural resources locally and globally.		There aren't any specific codes for sustainability in the SE code of ethics, just like there is no NSPE canon. Always put the public's interest first loosely relates.
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	Approve only safe well tested software and donate professional skills to good causes both relate closely to the definition. NSPE canon is more general, but one code that relates is place professional interests before personal.

6.2 Project Specific Professional Responsibility Areas.

Work Competence: This applies heavily to our project. We are working directly with our client because we want to produce work of high quality. Not only to do well in our class, but also to aid Reiman Gardens in their conservation efforts. We also strive for timeliness as we will only have a few months to develop the application. We have yet to produce any software so we can't rate it's competence.

Financial Responsibility: This applies to our project. It's our responsibility to find a platform to host the web server at a reasonable cost. We have yet to finalize or start paying for any service so we cannot rate our financial responsibility.

Communication Honesty: This applies heavily to our project. We are in close communication with our client through messages and in person meetings. They have a lot of ideas for features to enhance the product. It's our responsibility to be frank about what we can and cannot do in the

given time frame. We are performing high in this area. So far it has just been through being realistic with our abilities, but will be more applicable once we start development.

Health, Safety, Well-Being: This does not apply to our project. There are no concerns for health or safety of the stakeholders. Technically we will improve their well-being by making their work easier, but this area really doesn't apply.

Property Ownership: This applies to our project. We plan to utilize 3rd party APIs, and it is important we do so legally and with respect to their owners. We haven't done this yet so we can't rate our performance.

Sustainability: Our project relates to the sustainability of butterflies. Still, this area doesn't apply to our development of our product. There are no sustainability concerns.

Social Responsibility: This applies to our project. We are aiming to produce a product that improves society and specifically the butterfly conservation community. It's our responsibility to produce an application that is better than their current research process. We have yet to produce any software so we can't rate our performance.

6.3 Most Applicable Professional Responsibility Area

Communication Honesty: Other areas will become more relevant as time goes on, but communication honesty has been important from the start.

7 Closing Material

7.1 Discussion

So far, we have gathered all the requirements from the client and have begun to address which parts of our product will incorporate those requirements. Currently, we have not implemented any, but have made diagrams and prototypes of some data models to start.

7.2 Conclusion

We have made design sketches and various data models for the butterflies. Using Figma, we have a basic flow diagram of the app as well. Our goals for this semester were to get started on the design of the screens as well as get an idea of how the app will flow. For the most part, the team believes that we have met these goals for this semester. Going into the second semester, we will want to start working on implementing the screens in React as soon as possible so we can begin adding in databases and the various routes.

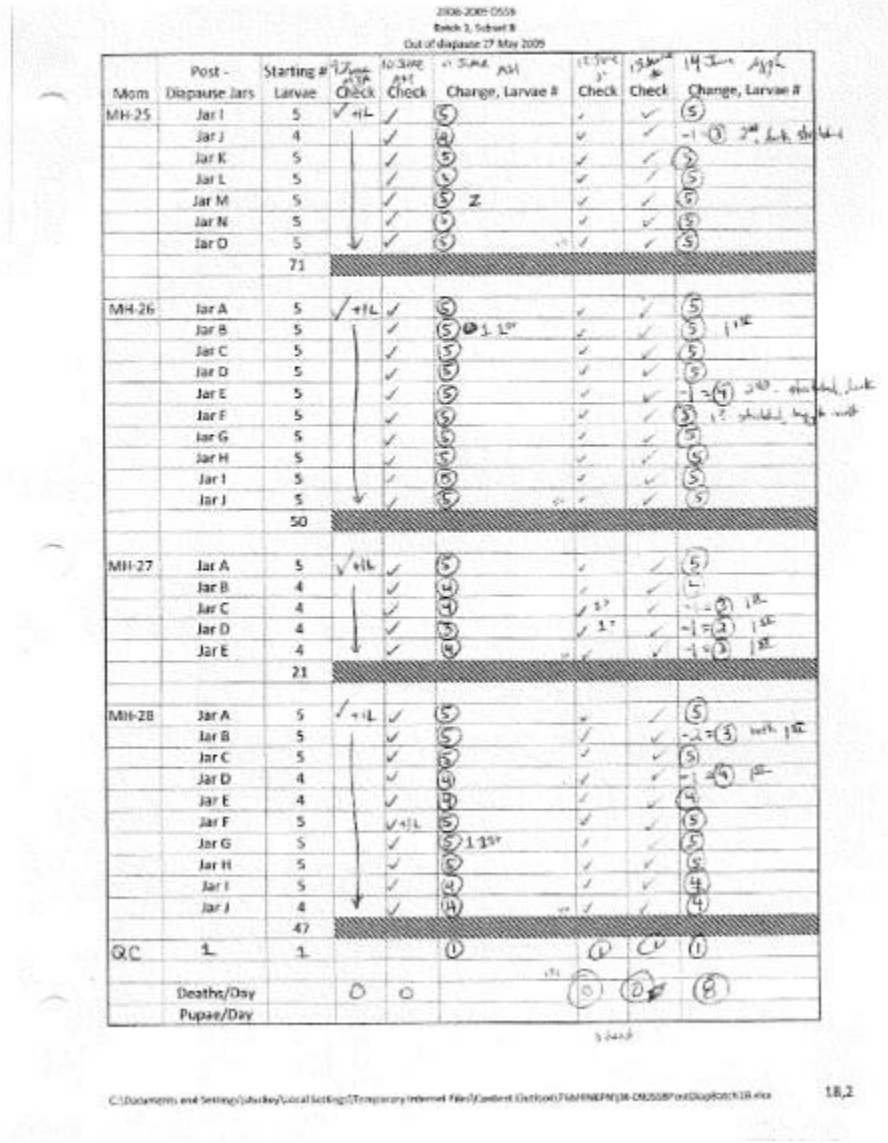
7.3 References

Outside research not needed at this point, will update when we start developing the app

7.4 Appendices

One data sheet that was/is being used by the Oregon Zoo: courtesy of the Oregon Zoo

Post-diapause: Oregon Zoo



2008-2009 OSU
Batch 3, Subset B
Out of diapause 27 May 2009

Mom	Post-Diapause Jars	Starting # Larvae	Check	Check	Change, Larvae #	Check	Check	Change, Larvae #
MH-25	Jar I	5	✓ +L	✓	5	✓	✓	5
	Jar J	4		✓	4	✓	✓	-1 ③ 2nd, dark, thick
	Jar K	5		✓	5	✓	✓	2
	Jar L	5		✓	5	✓	✓	5
	Jar M	5		✓	5 Z	✓	✓	5
	Jar N	5		✓	5	✓	✓	5
	Jar O	5		✓	5	✓	✓	5
		71						
MH-26	Jar A	5	✓ +L	✓	5	✓	✓	5
	Jar B	5		✓	5 1 1st	✓	✓	5 1st
	Jar C	5		✓	5	✓	✓	5
	Jar D	5		✓	5	✓	✓	5
	Jar E	5		✓	5	✓	✓	-1=4 2nd, dark, thick
	Jar F	5		✓	5	✓	✓	5 1? shiny, big, with
	Jar G	5		✓	5	✓	✓	5
	Jar H	5		✓	5	✓	✓	5
	Jar I	5		✓	5	✓	✓	5
	Jar J	5		✓	5	✓	✓	5
		50						
MH-27	Jar A	5	✓ +L	✓	5	✓	✓	5
	Jar B	4		✓	4	✓	✓	4
	Jar C	4		✓	4	✓ 1?	✓	-1=3 1st
	Jar D	4		✓	3	✓ 1?	✓	-1=2 1st
	Jar E	4		✓	4	✓	✓	-1=3 1st
		21						
MH-28	Jar A	5	✓ +L	✓	5	✓	✓	5
	Jar B	5		✓	5	✓	✓	-2=3 with 1st
	Jar C	5		✓	5	✓	✓	5
	Jar D	4		✓	4	✓	✓	-1=3 1st
	Jar E	4		✓	4	✓	✓	4
	Jar F	5		✓ +L	5	✓	✓	5
	Jar G	5		✓	5 1 1st	✓	✓	5
	Jar H	5		✓	5	✓	✓	5
	Jar I	5		✓	4	✓	✓	4
	Jar J	4		✓	4	✓	✓	4
		47						
QC	1	1			1	1	1	1
	Deaths/Day		0	0		0	0	0
	Pupae/Day							

C:\Documents and Settings\Jdunlap\Local Settings\Temporary Internet Files\Content.Outlook\7IAHYNEP\08-09SSBPostDiapBatch3B.xls

18,2

7.4.1 Team Contract

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

We will meet with our team on Mondays at 11am ideally at the library face-to-face. Additional meetings with Client and Advisor are set for Friday at 2:30pm in Coover on a weekly/semi-weekly basis If more meetings are needed, we will schedule those as a team.

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

Discord will be used to communicate among team members. Client, faculty, and TAs will be contacted via email.

3. Decision-making policy (e.g., consensus, majority vote):

Most decisions will be made via a group consensus.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

We will create a separate document that will list the date of a meeting as well as how many minutes we met for, and also a summary of the main points we discussed during the meeting for future reference or team members who were unable to attend.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

   Everyone is expected to attend all team meetings. If conflicts arise, the member must notify the group as soon as possible and find out what they missed.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

   Everyone is expected to work on their assigned tasks and adhere to deadlines. If they are falling behind they should let the team know so they can help.

3. Expected level of communication with other team members:

   Everyone is expected to check their Discord messages once a day M-F

4. Expected level of commitment to team decisions and tasks:

   Everyone is expected to give input into major decisions. Smaller decisions can be made by 4-5 members.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

Mike Gradle: Client Interaction

Kristen Hawken: Database Admin, Backend development

Ricky Smith: Testing, Backend development

Joanna Besselievre : Individual component design

Rose Druce Hoffman: Scribe , Backend development

Gabriel Owen : Testing, Frontend development

2. Strategies for supporting and guiding the work of all team members:

The team will set goals throughout the semester for both individual members and the group as a whole. The team will meet at least once a week to discuss these deadlines and set more as needed.

3. Strategies for recognizing the contributions of all team members:

During weekly meetings, the team will assess the contributions of each member to both adjust and recognize contributions. The goal is to have everyone contribute - a large amount of contribution isn't necessarily a good thing.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

Mike: Experience using Java, C#, C, Python, and some other languages. Mainly have done frontend development. Very familiar with both Git and Azure DevOps. Working at an internship that uses C# with Visual Studio integrated with DevOps.

Ricky: Experience working with Java Spring Boot, React JS in web app development from both classes and prior internships. Focus on Backend development, experience with SQL queries and 3rd party API integration. Familiar with Agile development approaches and comfortable with Azure DevOps and Git based source control/task management.

Rose: Experience with Frontend development with Android Studio from prior classes. Most familiar with Java and C but willing to learn as needed. Familiar with the Agile development process.

Jonah: Experienced with Java, C, and MySQL from classes. Also experienced with app and web development from classes and personal projects. Familiar with agile development process.

Kristen: Experience with Java, C, Javascript, and MySQL from classes. I used Scrum methodology in my previous internships. I feel most comfortable doing backend but have experience with web development.

Gabe: Production mobile web-app design experience, although mostly with software that has very high licensing costs. Most confident in C#, C, then Java, respectively. Would be very familiar with Azure Devops, VS, and SSMS - but am okay with using anything.

2. Strategies for encouraging and support contributions and ideas from all team members:

All members are encouraged to share their ideas at any time for group consideration. The group will not disparage any ideas from anyone and will make an effort to ensure that all members' ideas are incorporated in the work.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

Team members should speak up when they feel they are not being included in decisions or work. The team should then make an effort to include that member and ensure that their contributions are recognized.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

Create all necessary documentation for easy development of our solution.
Start working on a solution ready for implementation by the end of the semester.
Demonstrate to the faculty panel our ideas on the solution.
Understand all requirements the client is expecting for the solution.

2. Strategies for planning and assigning individual and team work:

Depending on the requirements, since we have a team of 6, we can split into two teams then divide the work among those two teams equally. The teams will most likely be frontend and backend.

3. Strategies for keeping on task:

Create a detailed list of requirements.
Mainly, work and tasks will be kept on the team's GitLab.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

A friendly reminder should be enough for the first infraction.

2. What will your team do if the infractions continue?

If absolutely necessary, we will contact either the TAs or the professor.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

1) Mike Gradle DATE 09/07/23
2) Ricky Smith DATE 09/08/23
3) Kristen Hawken DATE 09/08/23
4) Rose Druce-Hoffman DATE 9/9/23
5) Gabe Owen DATE 9/10/2023
6) Jonah Besselievre DATE 9/10/23