DOCUMENTATION – FLUTR
PROJECT PLANS

CLIENT: MR. NATHAN BROCKMAN
ADVISOR: DR. DIANE ROVER

Group Members

Amanda Friis - Full Stack Developer, Documentation

Tayler Barnhart - Team Leader, Cloud Integration

Nathan Geater - Full Stack Developer (Back End)

Alex Brown - Front End Development Leader

Muralikrishna Patibandla - Integration & Back End Developer

Project Management/Tracking Procedures

As features and work begin to be implemented, we will be adopting an agile sprint strategy, where the team will run 2 week sprints in order to assign work and implement features. The main idea behind this is that it will allow our team to work in smaller groups on different things, all while staying on the same track with one another. With the Flutr project, the project has a wide array of different features and problems we need to solve (which can be tied to the requirements needed). Allowing members to take smaller pieces and work on them over a couple of weeks helps with time management, as school, clubs, sports, and other things can make it difficult for the group to meet together often.

For tracking progress, we will use a couple of different tools. The first being the weekly reports that the team fills out - this makes for an easy tracking method, as we can see a short description of the project each member is working on and the hours that each member has put in each week. Additionally, we will also be using Gitlab to track individual features with milestones. This allows us to track more detailed information about the problems a member might be facing. Additionally, we will communicate over Discord multiple times a week for general check-ins and questions. Finally, the team leader (Tayler) will periodically check in with each member to make sure all problems are solved and that the team is moving forward together.

Task Decomposition

The tasks in this project mainly pertain to the web app, though a small bit deals with the kiosk. As such, they will be separated into different sections.

Web Application

- Finalize frontend and backend tools
- Implement database and app onto cloud hosting service
    - Design database to hold long periods of tracked data, as well as butterfly facts, details, and images
- Create communication interface between frontend and backend
- Create front page framing & general website design
- Build the butterfly tracking system & connect with backend database
- Allow previous database access & editing
- Export previous data to external source (PDF, Spreadsheet)
- Implement security admin system with multiple levels of admin & permissions
- Design app for multiple devices, including computer, tablet, and phone touchscreen
- Create automated butterfly counter with death counting
- Use data for intuitive maps, charts, and facts about the current butterflies in the database
- Allow for generation of new houses
    - Incorporate a simple tutorial system for other houses
    - Allow for house customization from the head admin
- Create documentation of our process and final design for bug testing and future updating

Kiosk

- Research correct computer parts needed for the kiosk
- Design & implement "parental controls" software to prevent the kiosk from accessing the internet freely
    - Implement QR code system to allow users to find the social media on mobile devices without leaving the Flutr mainpage.

- Test hardware parts in hot and humid environment (80℉ and 80% humidity)
- Install computer parts into kiosk
- Test software and internet prevention in kiosk

Project proposed Milestones, Metrics, and Evaluation Criteria

- A New Computer System Will Be Built for the Reiman Gardens Kiosk (End of Spring Semester)
    - Cooling System Will be sufficient to get through the summer months.
- Login System Created On The Backend with all User classes accounted for
- Basis for landing screens created on frontend. Functional with simulated postman server.
- Login connected on frontend.
- Location Creation and Head Admin Creation process implemented full stack.
- Entry of shipment information at individual locations created.
    - Allows for editing of submitted forms. IMPORTANT.
- Locations can perform "Release" action on their database of received butterflies.
- Head Admins can customize their location's database specifics such as Common Names, Estimated lifespans
- Head Admins can customize the look of their customer facing website (logo, color scheme, social links, contact information, 'The Note")
- Frontend displays the customer facing site for unauthenticated users.
- Statistics and Butterfly of the day created for customer facing site.
- Quality of life improvements (Switching Common and Scientific Names on the fly, picture toggling.)
- Beta Test by November '24
- Final Touches/ Bug fixes found in Beta by Mid December '24

Project Timeline/Schedule

Target 1: Kiosk

Planning

- Decide on what hardware we want to use

Design

- Create a plan for how to fit everything into the kiosk

Development

- Tear apart old kiosk to make room, find what we need to replace
- Replace all old tech inside kiosk with new tech

Testing

- Set up in Reiman Garden Pavilion
- Fix any concerns that arise during testing period

Deployment

- Permanently install

Phase	Start Date	End Date
Planning	March 1, 2024	March 16, 2024
Design	March 17, 2024	March 20, 2024
Development	March 21, 2024	April 19, 2024
Testing	April 20, 2024	May 4, 2024
Deployment	May 5, 2024	May 10, 2024

Target 2: Website Creation

Planning

- Requirements gathering
- Research on platforms
- Study previous projects for their failures and successes

- Develop general time frame

Design

- Choosing platforms for frontend and backend
- Wireframes for potential web pages
- Finalize requirements
- Plan software architecture

Development

- Create database
- Develop webpages
- Implement features

Testing

- Website testing
- Database testing
- Communication testing

Deployment

- Set up in Reiman Gardens for initial user-testing period

Phase	Start Date	End Date
Planning	February 1, 2024	March 10, 2024
Design	March 11, 2024	March 31, 2024
Development	March 18, 2024	May 15, 2024
Testing	April 15, 2024	September 1, 2024
Initial Deployment	May 10, 2024	May 15, 2024

Target 3: Website Customization

Planning

- Decide on database framework to use (Mongo DB)

Design

- Prioritize intuitive and functional design and aesthetics for visitor and staff experience
- Add fun facts for users to see

Development

- Build login system for different flight houses with their own permissions

Testing

- Website testing
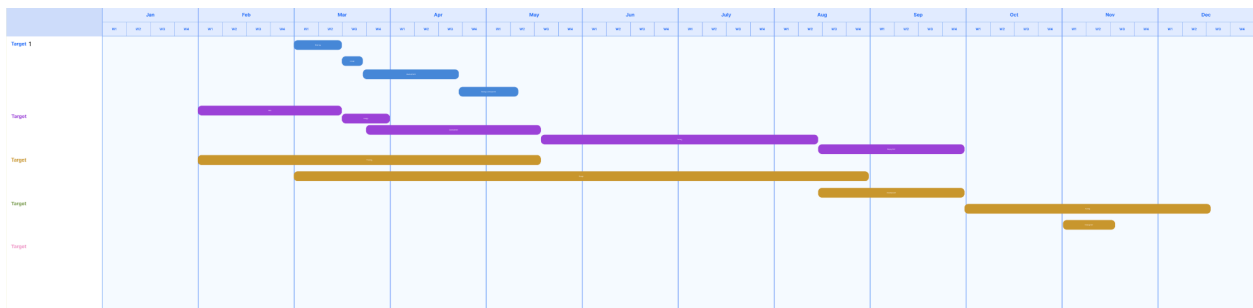- Database testing
- Communication testing

Deployment

- Deploy to other pavilions

Phase	Start Date	End Date
Planning	February 1, 2024	May 10, 2024
Design	March 1, 2024	September 1, 2024
Development	August 15, 2024	September 30, 2024
Testing	October 1, 2024	December 10, 2024
Deployment	November 1, 2024	November 15, 2024

Gantt Chart

Beginning February and ending in December.

Risks and Risk Management/Mitigation

Web Application

- Implement Database and App onto Cloud Hosting Service
    - Risk: Overspending or underestimating required resources.
    - Probability: 0.5
    - Mitigation: Use cloud services with extensive documentation, accessibility features, and scalable payment models. Conduct in-depth analysis to compare services.
- Design Database for Long Periods of Data Storage
    - Risk: Inefficient data retrieval and storage costs.
    - Probability: 0.2
- Create Communication Interface between Frontend and Backend
    - Risk: Incompatibility issues leading to poor performance.
    - Probability: 0.1
- General Website Design and Front Page Framing
    - Risk: Design not being responsive or user-friendly on all devices.
    - Probability: 0.2
- Build Butterfly Tracking System
    - Risk: Tracking system inaccuracies or inefficiencies.
    - Probability: 0.6
    - Mitigation: Prototype early and test with real data. Act on and incorporate feedback from client, staff, and volunteers based on testing.
- Implement Security Admin System
    - Risk: Inadequate security leading to unauthorized access.
    - Probability: 0.5
    - Mitigation: Use a tried and tested authentication framework. Regular security maintenance and updates.
- Design App for Multiple Devices
    - Risk: Poor user experience on certain devices.
    - Probability: 0.1

- Create Documentation for Maintenance, Updates, and Customization
    - Risk: Documentation being unclear, insufficient, or complex.
    - Probability: 0.5
    - Mitigation: Follow best practices in technical writing. Regular updates and peer reviews of documentation. Conduct user testing for feedback on features and simplify the UI.

Kiosk

- Research and Install Computer Parts
    - Risk: Parts not functioning as expected in environment.
    - Probability: 0.7
    - Mitigation: Select components rated for extreme conditions. Test and compare different options in various possible conditions.
- Design and Implement Administrator Control Software
    - Risk: Software being bypassed or failing to function.
    - Probability: 0.2

Personnel Effort Requirements

Task	Hours	Explanation	Source
Authenticate admin level users at individual location	30	Need to ensure that the super user can create other admins and admins can create other general users at their given location. Need to ensure security at each authentication level.	These estimates are gathered from our own personal experiences from classes and internships.
Take in information about shipment of butterfly from the user and save (Species, count, supplier)	40	Webframe design and development of webpage: 10 hours Database design and development: 10 hours	
Take in information about which butterflies are being release into habitat and which shipment they came from	40	Webframe design and development of webpage: 10 hours Database design and development: 10 hours	
Edit general information about the location and the site	30	Ensure user can edit a webpage based on their individual location	
Update shipment database	20	Edit/add/delete information from the shipment database	
Update list of which butterflies are in habitat, and start their respective timers	30	Develop life timer based on release and estimated life span Edit/add/delete information from the release database	
Serve current information to the general public	10	Home-page website that users can view to see butterflies currently flying	

TOTAL HOURS: 200 hours

HOURS PER PERSON: 42 hours

Given a time period of February - December, and excluding summer (about 33 weeks)

HOURS PER WEEK PER PERSON: 1.2 hours

Other Resource Requirements

This project requires the use of a hosting service, which DigitalOcean is currently the chosen option, to deploy the site on. It also requires a domain, which will be "flutr.org" and is currently owned by Reiman Gardens. While both of these are closely related to financial resources, they can sometimes be forgotten as a resource needed.