

Development Plan

Team Pina Colada | AgMonitor

[Kaiwen Li](#) (Team Lead), [Alex Mei](#) (Scribe), [Jasun Chen](#), [Jayden Yu](#), [Yuyuan Wang](#)

Features

Data Visualization [LOW PRIORITY]

- Implement pagination of data because we are exceeding memory and storage issues. [DONE.]
- Integrate the raw data for visualization with the backend.
- Visualize display of AI output.

Asset Configurations [HIGH PRIORITY]

- Connect asset frontend forms to backend. [DONE.]
- Acquire email dynamically from the login feature. [DONE.]

AI Optimization [HIGH PRIORITY]

- Algorithm Input Dataflows
 - Get/Set battery configuration [HIGH PRIORITY]
 - *lowerLimit*, *maximumLimit* for battery threshold
 - *Battery Storage Size*. (kWh)
 - Get/Set User Preferences for algorithm [HIGH PRIORITY]
 - User preference for balancing between cost and shutoff risk (0 to 100)
 - Sliders on dashboard page
 - User preference for number of hours of power during shut off (0 hr - 20 hr)
 - Get weather data [HIGH PRIORITY]
 - Can also Integrate with weather API (<https://www.weatherbit.io/api/alerts>)
 - Write function that takes lat/long and returns an array of severity ratings
 - Get solar generation data [MEDIUM PRIORITY]
 - Will have to get user's lat/long and store that. User provides address -> lat/long
 - Integrate with this api <https://forecast.solar/> or nrel-pysam to predict user solar generation for next (two?) days
 - Need to add generation asset with *plane declination*, *plane azimuth*, *modules power* fields
 - Get/Set user location and convert into lat/long [MEDIUM PRIORITY]
 - Get current Battery charge level [LOW PRIORITY] (will have to mock Tesla API)
 - Get/Set Tesla (Flexible load) energy demand [LOW PRIORITY]
 - Note: For Tesla, we can subtract the current amount of charge from the charge limit to find the energy demand, but we will most likely not integrate with the Tesla API because of security. Therefore, we will have to just mock a function that gets the Tesla's current charge level and its charge limit.
- Algorithm
 - Daily job to run algorithm code on server [HIGH PRIORITY]
 - Algorithm Implementation to pull input data and output optimum settings [HIGH PRIORITY]
- Algorithm Output Flow
 - Sending of text or email to user with ideal settings [LOW PRIORITY]

Testing [MEDIUM PRIORITY]

- Setup CI/CD pipeline for frontend. [DONE.]
- Write more unit tests for the backend.

UI Improvements [LOW PRIORITY]

- UI Changes based on user interview feedback.
- UI Test for Tablet compatibility.
- Standardize UI formatting.

Timeline

Sprint 5 (01/04/2021 - 01/18/2021)

Week 1

- Jasun: Determine concrete roadmap for AI Optimization.
- Alex: Implement pagination of data because we are exceeding memory and storage issues.
- Jayden: Connect asset frontend forms to backend.
- Kaiwen: Setup CI/CD pipeline for frontend.
- Alice: Pass logged in user information to other pages. [Acquire email dynamically from login feature.]

Week 2

- Alex: Solar Data Integration
- Jayden (Frontend) + Kaiwen (Backend): Get/Set battery configuration; Get/Set User Preferences for algorithm; User Address (Conversion to Lat/Long) and Store in DB. Unit Tests.
- Jasun: Finalize Cost Function Research and Start Algorithm Implementation to pull input data and output optimum settings.
- Alice: Weather Data Integration

Sprint 6 (01/18/2021 - 02/01/2021)

- Algorithm Implementation
- Data Visualization
- Implement Job Scheduler for AI Algorithm Computation
- Implement AI Settings Notifications
- "Integrate" with Tesla API

Sprint 7 (02/01/2021 - 02/15/2021)

- User Testing
- UI Improvements
- Last Minute Patching
- Presentation Prep

Sprint 8 (02/15/2021 - 03/01/2021)

- User Testing
- UI Improvements
- Last Minute Patching
- Presentation Prep