
AgBizClimate Final Progress Report

— By: Shane Barrantes, Thomas
Noelcke, Shengpei Yuan —

Project Overview

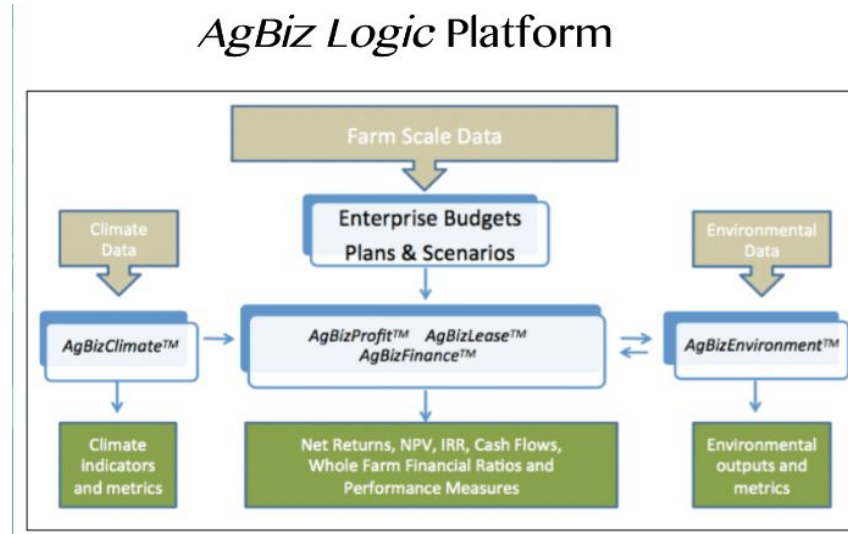
- 1) Project Introduction
- 2) Summary of Progress
- 3) What's Left
- 4) Major Blockers
- 5) Demo
- 6) Backend API
- 7) Retrospective

Project Introduction

- What is AgBizLogic?
 - Suite of decision web tools
 - Current tools Include: AgBizProfit, AgBizLease, AgBizFinance, AgBizEnvironment
- Primary Goal: Create AgBizClimate
- Functionality: AgBizClimate aims to give farmers and ranchers tools to help mitigate the effects of Climate on crops and livestock.

Scope

- Our Project Is part of the Greater AgBiz logic Platform.
- We are specifically adding functionality to the AgBizClimate Module.



Functionality Pipeline

- Users log in
- Users selects budget
- Users selects region
- User Selects scenario type
- Users get visualized climate data for region
- Users adjust yields based on results
- Users review their budgets
- Note: Process is iterative

Summary of Progress

- Resolved work items.
- Items in progress.
- ToDo's

Resolved Work Items

- Updated Requirements Document to reflect changes to API access.
- Created Concept script that connects to the threads database and gets data using NETCDF4.
- Create Install script for NETCDF4 and dependencies.
- Added UI elements needed to create a short term climate scenario.
- Developed charts page that creates visualization of short term climate data and allows users adjust yields.
- Implemented and refined work around for NetCDF issues.
- Set Up Endpoint to serve data from the API

Resolved Work Items - Cont.

- Bug Fixes
 - Fixed bug that prevented using multiple budgets.
 - Fixed issues that caused error when going from the chart page to the budget page.
- Updated the Frontend API to Use the dynamically generated data.
- Created Unit Test for Front End Code.
- Updated Unit tests for the rest of the front end to reflect changes to the application.

Items in Progress

- Creating a fail safe for the application if the climate data source is under maintenance.
- Document Updates.
 - Requirements Document
 - Poster

What's Left

- Document Updates.
 - Design Document.
 - Technical Review.
- Clean Up Back end Code.
 - Pull out API class and set it up so we can import it into the end point.
- QA and Testing.
 - Backend Testing Unit Tests.
 - Updating Front End Unit Tests.
 - Manual testing/Ad Hoc testing.

Stretch Goals

- Create PinPoint tool for selecting location.
- Add more models to the plot.
- Create plot so we can show 90% confidence interval.

Overview of Major Blockers

- Since our midterm progress report we have had no major blockers
- All previous major blockers have been resolved
- Overcame major blocker with HTTP method of getting climate Data.

Demo

- Demonstration of The Application.
 - Overview of the app
 - Demo with dynamic data and multiple budgets.
- Discussion of the API and endpoint.
 - Discussion of the API.
 - Discussion of the end point.

Retrospective - Positives

- We came up with a URL scrape via python to access the NorthWest Climate Toolbox
- We improved greatly on having purpose and teamwork
- We have fully functional beta
- Despite being behind for the first several weeks of the term we are now on schedule

Retrospective - Deltas

- As of this point in development we have no major deltas
- All of our deltas from the midterm have been resolved via action items
- We now have to go back and update our documentation to reflect our implementation
- We need to switch gears to start working on testing and QA

Retrospective - Actions

- We will now set weekly tasks for specific documents to update
- We will now set bi-weekly tasks for testing goals