CS Capstone Software Design Document

November 29, 2017

Linking Seasonal Weather Data to AgBizClimate^{TM}

Prepared for

Oregon State University

Clark Seavert

Sean Hammond

Prepared by

Group26

Thomas Noelcke

Shane Barrantes

Shengpei Yuan

Abstract

This design document will cover the proposed design of the AgBizClimate^{TM}project. We will first give a general introduction to the project. This section will provide some context for why we are doing this project and what this project hopes to accomplish. Next we will talk about Architecture design. This section will describe a high level structure for the project. After that we will discuss the data for the project and how it will be structured. Then we will discuss in detail the design of each component. Then we will discus the design of the user interface. Finally, we will provide a requirements matrix which will show how each component fulfills the functional requirements.

Contents

List of Figures

1 Introduction

1.1 Purpose

The purpose of this Software Design Description (SDD) is to describe the architecture and system design of the *AgBizClimate* project. This document will provide a high level design for the *AgBizClimate* short term climate tool. This document will also provide a detailed description of the design of the data this project will need to use. We will also break down each component and discuss the design of each component in detail. After that we will discuss the design of the user interface. Finally, we will have a requirements matrix. The requirements matrix will show how each component fulfills the functional requirements for this project. This document is intended for the project owners and software developers of the *AgBizClimate* system. This document is intended to be a guide for the implimentatino of the *AgBizClimate* short term climate tool.

1.2 Overview

Seasonal climate is one of the essential factors that affects agricultural production. As a module of *AgBiz Logic*, *AgBizClimate* delivers essential information about climate change to farmers, and help professionals to develop management pathways that best fit their operations under a changing climate. This project aims to link the crucial seasonal climate data from the Northwest Climate Toolbox database to *AgBiz Logic* so that it can provide changes in net returns of crop and livestock enterprises through powerful graphics and tables.

1.3 Scope

This project is a part of a much larger AgBiz Logic^{TM}program. However, the purpose of this project is to add a short term climate tool to the *AgBizClimate* module. This limits the scope of the project to the *AgBizClimate* Module. Additionally, we will only be adding the short term climate data tool as the long term climate data tool already exists.

Currently *AgBizClimate* has a long-term climate tool but no such tool exists for short term climate data. We will implement a tool to extract short-term climate data from the Northwest Climate Toolbox database, display it to the user and allow the user to adjust crop and livestock yields or quality of products sold and, production inputs. Moreover, a landing tool will be developed to allow users to switch between short-term seasonal tool and long-term climate data tool.

1.4 Definitions, Acronyms and Abbreviations

REST - Representational State Transfer, This is a type of architecture that manages the state of the program. This is especially popular in web development.
API- Application Programming Interface. This is a piece of software that allows a connection to another piece of software providing some sort of service.
NWCTB - Northwest Climate Toolbox. This is the database we will be connecting to that will provide the short term climate data we plan to use.
Climate Scenario - This is a theoretical calculation of yields, inputs and of the overall budget for one situation based on the climate data.
SQL Database - This is a relational database that allows for storing and accessing data.
NOSQL Database - This is a non-relational database that allows for data storage and data access.
UI - User Interface, This is a piece of software that allows a human to interact with the software. Often this is what the user sees while using software.

1.5 References

[1] C. F. Seavert, "Negotiating new lease arrangements with the transition to direct seed intensive cropping systems," 2017.

[2] S. Y. Thomas Noelcke, Shane Barrantes, "Problem statement," 2017.

2 System Overview

2.1 Product Functions

AgBizClimate is a web based decision tool that will allow users to gain specific insight on how localized climate data for the next seven months will affect their crop and livestock yields or quality of products sold and production inputs. The *AgBizClimate* tool will allow users to input their location (state, county) and a budget for the specific crop or livestock enterprise. *AgBizClimate* will select climate data for the next seven months for that location and provide graphical data showing temperature and precipitation. Users will then be able to change yields or quality of product sold by a percentage they think these factors will affect and modify production inputs. Finally the tool will calculate the net returns.

2.2 User Characteristics

AgBizClimate users can be split into two subgroups, agricultural producers and climate researchers. The first subgroup, agricultural users who use this product tend to be between fifty and sixty years old of mixed gender. Their educational background ranges from high school to the completion of college. The primary language this group uses is English, but there are some Spanish users as well. Most of the users in this group tend to have novice computational skills. The primary domain for these users is agricultural and business management. Most agricultural producers who use this product are motivated by the potential profit that the decision tool *AgBizClimate* could potentially offer. The second subgroup, climate researchers range from ages twenty to forty and are of mixed gender. The educational background for most climate researchers exceed the postgraduate level with their primary language being English. These users generally have advanced computational skills and are motivated by the easily accessible climate and weather data.

2.3 Constraints

There are several key constraints that this product has to work within. The first constraint is that we only have access to two data parameters from the North West Climate Tool box, precipitation and temperature. Secondly, we only have access to their data via the NWCTB API which could have additional restrictions such as limited usage per day, mislabeled data, or poor documentation. Thirdly, we dont have access to any of the hardware that *AgBizClimate* is exists on as it is being managed by a third party. This will prevent us from improving the hardware or cause roadblocks if their servers are having issues. Lastly, we are limited to using the languages Python and JavaScript since we are integrating our product into an already existing project.

2.4 Assumptions and Dependencies

We are assuming that the Northwest Climate Toolbox is a functional API that will allow us to pull location based temperature and precipitation data. This data will most likely come in the form of a text body of which we will then format into a JSON object and store in a MongoDB database for future use. Due to the fact that we are writing an addition to an existing project we do not need to interact with the user budgets as these have already been defined. This fact extends to the calculations portion of the *AgBizClimate* product. Our team will simply be accessing data via the NWCTB API, then format the data, store the data, and hand the data over to the tool while will provide

some additional front end support.

3 System Architecture

3.1 Architectural Design

Shown Below is the architectural design for the *AgBizClimate* project. This UML diagram shows the high level components of this application. This Diagram also shows how these components will interact.

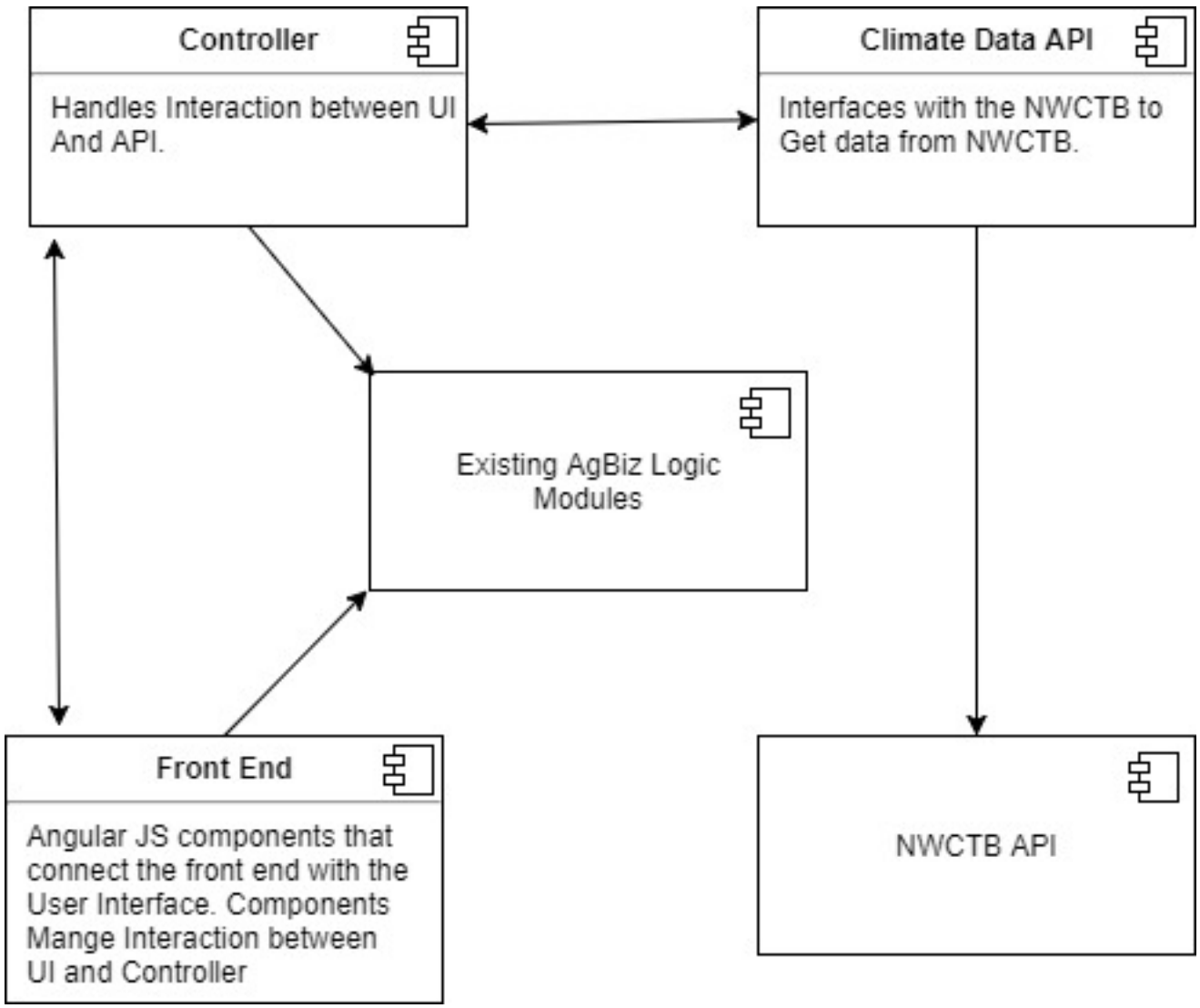

Figure 1: System Architecture Design for *AgBizClimate* project

3.2 Decomposition Description

The *AgBizClimate* project can broken down into six components. The six components are the Backend Controller, The Front End Controller, UI, Existing *AgBiz Logic* Modules, Climate Data API, and the NWCTB API. In this section we will describe each models function and how it interacts with the other modules.

3.2.1 Backend Controller

This component is responsible for connecting the dots between the rest of the components. Generally, the backend controller will handle incoming requests from the front end controller and return the requested content. This component will also interface with the existing *AgBiz Logic* modules and Climate Data API so it can provide all the requested information.

3.2.2 Front End Controller

The Front End Controller will act as an interface between the UI, the existing *AgBiz Locic* modules and the Backend Controller. This component will handle UI action made by the user and will use those to make requests to the Backend and existing *AgBiz Logic* Modules. This component will then take the result of these requests and display the relevant information to the user. This component will also handle user inputs such as a button push or clicking on a drop down menu. This component will take these actions and modify the UI to reflect the actions the user preformed.

3.2.3 UI

This is the portion of the application that the user will see and interact with. The primary responsibility of this component is to display information to the user. This component will also be responsible for interacting with the front end Controller to ensure that user actions so the program responds correctly to user action.

3.2.4 Existing *AgBiz Logic* modules

This is not a component but rather a collection of components that already exists as part of the *AgBiz logic* system. We will use these components to preform a variate of actions including, retrieving budget data, managing user information, making modifications to budget data and saving budget data back to the database. We will interface with these component from the Front End Controller to handle budget data. We will also interface with these components from the back end to handle user data.

3.2.5 Climate Data API

The Climate Data API component will interface withe the NWCTB to provide long term forecast data. This component will take requests, with location data, from the Backend controller and will respond with the formatted data from the NWCTB. To do we will interface with the NWCTB API to retrieve the data. Then we will take the data from the NWCTB, parse it into JSON, apply some formatting and pass it back to the Backend. For the purposes of this project this component is only going to interface with the Backend Controller. However in the future this API maybe used by other sections of the application as well.

3.2.6 NWCTB API

The NWCTB API will be our data source for this project. This component will provide the climate data by interfacing with the back end controller. Currently we are not sure how this will happen as the NWCTB has not responded to our requests for API access.

3.3 Design Rationale

We've chosen to design this system this way in part because of the nature of our system. We need a front end controller to handle the clients interaction with the server because this is a web development project and the front end will be separate from the server. The front end controller will facilitate the communication between the client and the server.
We also chose to use a Backend Controller so we can facilitate the communication between the front end controller and the various components on the backend. This makes the application easier to build, test and maintain. This also allows for one line of communication between the backend and the front end. This is necessary to keep the interactions between the backend code and front end code simple. This allows for large changes to be made to both the front end and the backend with out causing them to impact each other.
We also chose to create the Climate Data API as its own service. We chose to do this because it's easier to test and then the Climate Data API can be reused in future projects and with other components. If we had built the Climate Data API into the backend controller this would not have been possible.
The NWCTB and Existing *AgBiz Logic* components have already been implemented out side the scope of our project. However, we are still planning to use them in our project. This is why we have create these modules because we will be using them as part of the design of our system but do not want to tightly couple our project with these existing components.
More generally the components in our system use the REST API architecture type. We chose to do this because it allows for flexible reusable modules. The REST API Architecture also allows us to break our application into independent modules that are easier to develop, test and maintain. This division of our application also makes it more scalable allowing it to keep up with future demand.

4 Data Design

4.1 Data Description

In this section we will discuss the design of the data required for this system. The data needed to implement this system includes the user data, climate scenarios, Budget Data and the climate data. It should be noted that this project adds the climate data to the system. User data, climate scenarios and budget data have already been implemented as part of the existing *AgBiz Logic* system. However, since we will be using this data as part of our project I've included their design in this section.
Shown below is the design for the data we will use in this program. In this UML diagram are all the various entities required by this system. Additionally we also show the relationship between different entities. Shown below are the UML diagrams for User Data, Climate Scenarios, Budget Data, Climate Data and Related Entities.

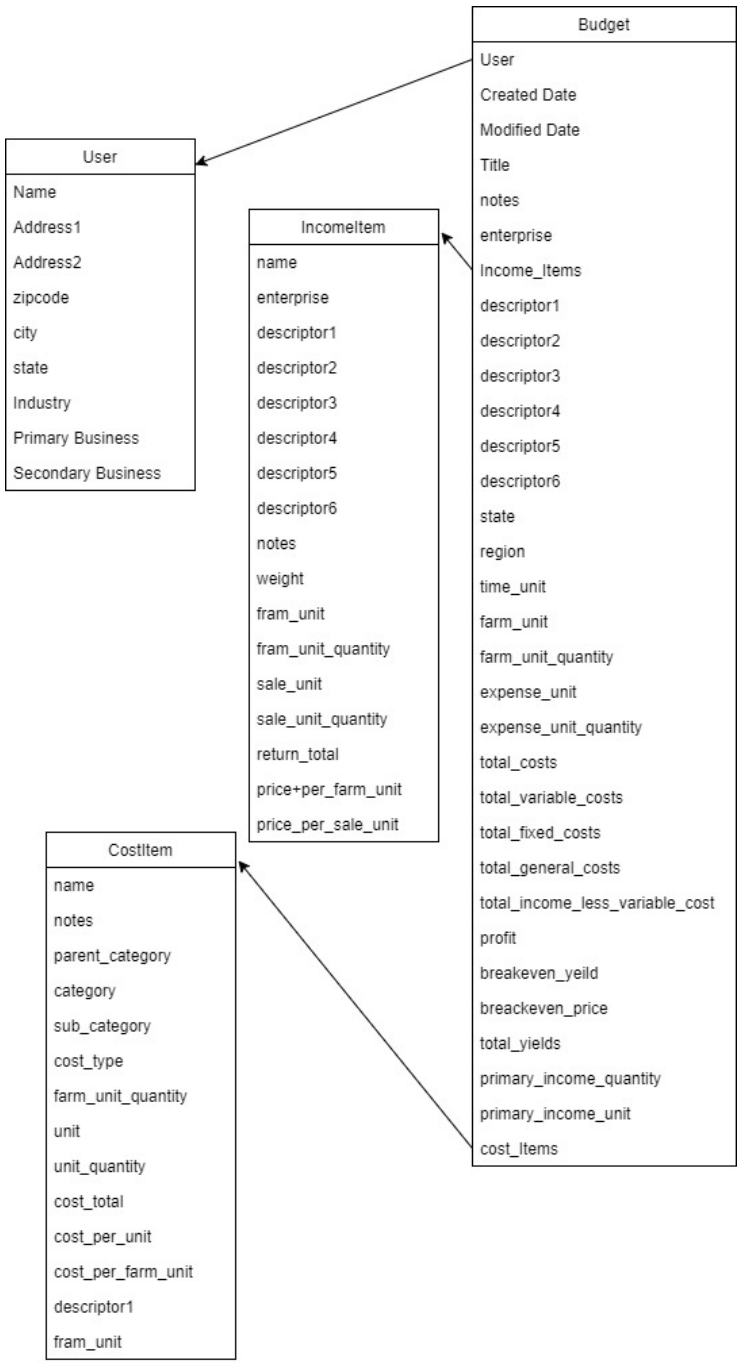

Figure 2: User Data Currently Implemented in the *AgBiz Logic* project

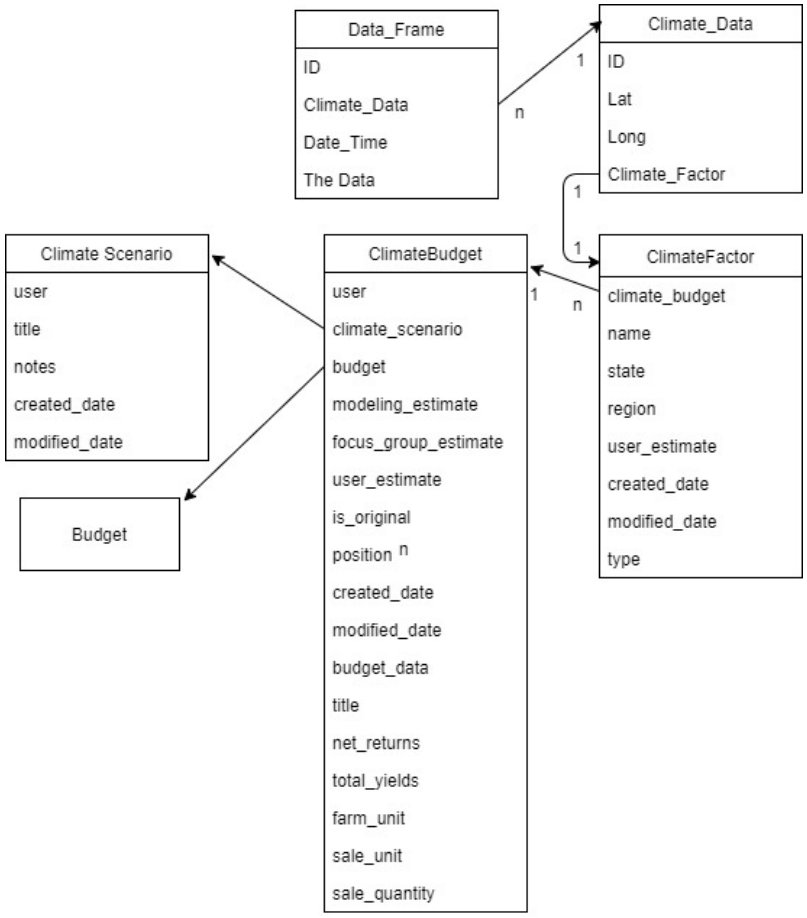

Figure 3: Design For the Climate Data for the *AgBizClimate* project

4.2 Data Dictionary

In this section we will describe each piece of data and what it represents. We will also discuss possible values for each piece of data.

4.2.1 User Data

4.2.2 Climate Scenarios

5 Component Design

5.1 Front End Controller

5.1.1 Angular Components Design

5.2 Controller Design

5.3 API Design

5.3.1 Overview

In this section we will discuss the design of the API that will interface with the NWCTB. This API needs to get the data from the NWCTB, format the data, and send it to the client. Currently, there is a lot of uncertainty around the design of this API because we do not know what sort of API access that we will be given from the NWCTB. We are trying to contact the NWCTB development team regarding our API access but the NWCTB hasn't been very responsive. Because we still don't have NWCTB API access yet and have no date when this might be accomplished, we will discuss several possible options that do not require NWCTB API access along with one design option that includes NWCTB API access.

5.4 Cached Climate Data Data Base

5.5 Testing Design

5.5.1 Front End Testing

5.5.2 Controller testing

5.5.3 API Testing

6 User Interface Design

6.1 Overview of User Interface

6.2 Screen Images

6.3 Screen Objects and Actions

7 Requirements Matrix