

# CS CAPSTONE SOFTWARE DESIGN DOCUMENT

NOVEMBER 26, 2017

## Linking Seasonal Weather Data to AgBizClimate™

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™ 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 THE FRONT END DESIGN. IN THIS SECTION WE WILL TALK ABOUT THE USER INTERFACE AND HOW WE PLAN TO IMPLEMENT THE UI. AFTER THAT WE WILL TALK ABOUT THE DESIGN OF THE BACK END CONTROLLER. IN THIS SECTION WE WILL DISCUSS HOW THE PAGES WILL BE SERVED. FINALLY, WE WILL TALK ABOUT THE DESIGN OF THE API. IN THIS SECTION WE WILL TALK ABOUT HOW WE PLAN TO INTERFACE WITH THE NORTHWEST CLIMATE TOOL BOX TO GET THE DATA AND SERVE IT TO THE CLIENT.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Overview . . . . .	2
1.3	Scope . . . . .	2
1.4	Definitions, Acronyms and Abbreviations . . . . .	2
1.5	References . . . . .	2
<b>2</b>	<b>System Overview</b>	<b>3</b>
2.1	Product Functions . . . . .	3
2.2	User Characteristics . . . . .	3
2.3	Constraints . . . . .	3
2.4	Assumptions and Dependencies . . . . .	3
<b>3</b>	<b>System Architecture</b>	<b>4</b>
3.1	Architectural Design . . . . .	4
3.2	Decomposition Description . . . . .	4
3.2.1	Backend Controller . . . . .	4
3.2.2	Front End Controller . . . . .	4
3.2.3	Existing <i>AgBiz Logic</i> modules . . . . .	4
3.2.4	Climate Data API . . . . .	4
3.2.5	NWCTB API . . . . .	4
3.3	Design Rationale . . . . .	4
<b>4</b>	<b>Data Design</b>	<b>4</b>
4.1	Data Description . . . . .	4
4.2	Data Dictionary . . . . .	4
<b>5</b>	<b>Component Design</b>	<b>4</b>
5.1	Front End Controller . . . . .	4
5.1.1	Angular Components Design . . . . .	4
5.2	Controller Design . . . . .	4
5.3	API Design . . . . .	4
5.3.1	Overview . . . . .	4
5.4	Cached Climate Data Data Base . . . . .	5
5.5	Testing Design . . . . .	5
5.5.1	Front End Testing . . . . .	5
5.5.2	Controller testing . . . . .	5
5.5.3	API Testing . . . . .	5
<b>6</b>	<b>User Interface Design</b>	<b>5</b>
6.1	Overview of User Interface . . . . .	5
6.2	Screen Images . . . . .	5
6.3	Screen Objects and Actions . . . . .	5
<b>7</b>	<b>Requirements Matrix</b>	<b>5</b>

## 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. We will also break down each component and discuss the design of each component in detail. This document is intended for the project owners and software developers of the *AgBizClimate* system.

## 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*<sup>TM</sup> 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.

## 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 don't have access to any of the hardware that *AgBizClimate* 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 the application. This Diagram also shows how these components are connected.

### 3.2 Decomposition Description

The *AgBizClimate* project can be broken down into five components. The five components are the Backend Controller, The Front End Controller, Existing *AgBiz Logic* Modules, Climate Data API, and the NWCTB API. In this section we will describe each model's 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. Through this process this component will also interface with the existing *AgBiz Logic* modules along with the Climate Data API.

#### 3.2.2 Front End Controller

The Front end Controller will be responsible for

### 3.2.3 UI

### 3.2.4 Existing *AgBiz Logic* modules

### 3.2.5 Climate Data API

### 3.2.6 NWCTB API

## 3.3 Design Rationale

# 4 Data Design

## 4.1 Data Description

## 4.2 Data Dictionary

# 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**