Oregon State University

CS 461

Fall 2017

Problem Statement

Author:
Thomas Noelcke

Instructor:
D. Kevin McGrath
Kirsten Winters

Abstract

The Project we are working on requires us to connect the Northwest Climate Toolbox database to an already existing AgBizClimate System. This System Will take the data from the weather database and produce a crop yield forecast. This forecast will then be displayed though the AgBizClimate system. The goal for this project is to provide guidance for farmers, researchers and government agencies on how climate change will affect crop yield.

We will solve this problem using a python based RESTFUL API as a Django application. Essentially our application will go out to the database get the necessary data and package it up as a JSON object for the AgBizClimate system.

1 The Problem

Climate Change, a big and difficult problem for every one, but especially so for the worlds food producers. As the climate continues to warm the current agricultural system will be pushed to innovate, become more efficient and ultimately feed the world. To meet this goal farmers will need cutting edge tools that provide farm-level support. Governments and researchers will need tools to help provide guidance on the effects of global warming and global warming related policies. The AgBizClimate program is one solution to this need. Currently the AgBizClimate system is just the business logic along with the view to solve this problem.

This is only half of the solution to providing accurate climate predictions for farmers, researchers and Governments. To complete the solution we will need the necessary climate data for the AgBizClimate system. Luckily one such database already exists in the Northwest Climate Tool Box. Our problem is to connect the current AgBizClimate system to the Northwest Climate Toolbox database.

ADD ADDITION INFORMATION AFTER MEETING WITH CLIENT

2 The Solution

To solve this problem we will create a python based RESTFUL API in Django. We will use the Django REST framework to create the API. We will use the REST frame work to connect to the Northwest Climate Toolbox. Once connected to the database we will need to determine what data we need from the database and what form the AgBizClimate system will need the data in. Once we know what the AgBizClimate system needs it will be fairly easy to process the data and pass it to the AgBizClimate system.

Another problem that will need to be solved is how to get meaningful data out of the data base. I anticipate that this database will be rather large and contain lots of tables. To make accessing the data easier I would like to build a transpiler. The transpiler is another layer of abstraction that will write the necessary query statements need to get the data out of the database. This will make writing the API much more simple as we will not have our Query Statements spread out so much in the file. This will also make building complex Query statements easier and will lead to clearer faster code.

ADD ADDITION INFORMATION AFTER MEETING WITH CLIENT

3 Performance Metrics

We will know when we have completed this project when our system successfully and accurately responds to every request made by the AgBizClimate system. Successfully meaning it delivers the data and accurately meaning the data is correct. Our system will need to respond in under 2 seconds to each request.

We will also provide quality assurance for our project as 90 percent of the code will be covered by unit tests with 100 percent of the unit tests passing. These tests will be non-trivial and will attempt to emulate how the system is intended to be used. These test will use mock data instead of pinging the database.

ADD ADDITION ITEMS AFTER MEETING WITH CLIENT