Oregon State University

CS 461

Fall 2017

# Tech Review AgBizClimate©

*Author:*
Thomas Noelcke

*Instructor:*
D. Kevin McGrath
Kirsten Winters

**Abstract**

The purpose of this document is to research and consider different technical options for our application. In this document we research different options for data storage, HTTP request frame works, and testing frameworks. I will consider three possible choices for each section of the application. For each of these options I will weight the pros and cons of each. After comparing the different options I will select the option I would like to use for the *AgBizClimate* application.

# Contents

# 1 Data Storage

## 1.1 Overview

For the *AgBizClimate* application we will need a way to store data so it can be easily retrieved later. For this application we will store a variety of data including budget data, weather data, and user information. This information will need to be quickly recalled so it can be used in our application. Generally, we will want to select a data storage option that will be easy to set up and allow us a lot of flexibility with what kind of data that can be stored. We will also want a data storage option that will quickly recall stored data so it can be used by the application.

## 1.2 Criteria

To determine the best choice for our application i will analyze the performance of the data storage based on, Ease of Development, and Ease of Set Up and Flexibility. I will analyze run time speed however, this will not be one of the criteria as it is not critical that our data is retrieved as quickly as possible. Generally, its much more important to consider Ease of development, Set up and flexibility because the primary concern is being able to get the application up and running quickly. Ease of development and ease of set up will be a subjective measures for each option. In those sections I will use the opinion of other software developers along with my own experience to compare each option.For these criteria I will rate each option on a scale from very easy to very hard. Flexibility will be a measure of how easy it is to store different kinds of data using each option. For Flexibility I will measure each option from flexible to rigid.

## 1.3 Potential Choices

### 1.3.1 PostGreSQL

PostGreSQL is an open-source relational database management system. PostGreSQL uses the Sever Querying Language (SQL). PostGreSQL uses the relational database model. This model sets up tables that represent a certain type of data. We can then run SQL quires on this data base to get the data we need for our application [**?**]
Though SQL is quick, In my experience it is not as developer friendly as the other options in this analysis. Generally, writing raw SQL queries is difficult and time consuming. This is especially true when your data models are very complicated. Using Raw SQL also requires the developer to manually figure out how to make the mapping between the database and the models used at the application level. With raw SQL it is also more difficult to change the structure of the database once you have created the database. These changes will require potentially complicated scripts along with scripts that convert the data to fit into the new structure [**?**].
Another problem with SQL in my experience is that it is more difficult to set up. The configuration process can be complicated. Additionally, if you choose to use raw SQL you must also set up your data base as third party application separately from your actual application.
Another problem with SQL is that it is rather rigid in the way that you must store data. For example if you want to store a list object in an SQL data base you must create a new table where one row represents one item in the list. This item must have its own unique id. Additionally, if you want to have a list of lists it gets even more complicated. Now you need to create another table to represent a name and ID for each list you want to store and you must relate every item you want to put in that list back to the parent item in another table. This can get very complicated in a hurry.
Another problem is if you don't know what the structure of the data is going to look like before run time it is impossible to store this data in an SQL database. This makes PostGreSQL rather rigid in

terms of flexibility.

### 1.3.2 Python ORM

Python ORM or Object Relational Mapper does not substitute for an SQL database. There will still need to be an SQL database running on the back end. However, the ORM framework allows developers to create objects in python that then map to the data base. Often times this type of frame work allows the developer to create the objects first and let the framework deal with creating the SQL database on the backend. Given that this is not a replacement for an SQL it does make working with an SQL database much easier.
This type of frame work has several advantages, it allows for easy development and set up. The ORM frame work allows the developer to be completely insulated from the SQL data base on the back end. This means that instead of writing SQL queries to get data from the database the developer is able to use objects in python to access data that is stored in a database. This makes development and set up Easier on the developer. This is because the developer doesn't have to worry about writing complicated SQL statements or setting up complicated relationships between tables. This frame work also allows the developer to develop the code first and let the framework worry about creating the database the data will ultimately be stored in [**?**].
The ORM framework also allows for great flexibility in what you can store in a database. This type of frame work allows for mappings between python objects and the data base. So nearly anything you can store in an object in python, you can also store in a database. However, it should be noted that you must know the structure the object you are trying to store before run time.
Though this approach is very easy for the developer it isn't with out cost. The ORM approach does take a hit in terms of run time performance. The ORM framework will be slower than raw SQL. Another problem with this type of framework is that it doesnt leave the developer very much control over the database. This means that you are stuck with what you get. If the frame work structures something in the database in a way you don't like you don't have a ton of choice about that. Additionally, making database chances can cause you to loose data in the database if you are not careful about how you handle the migrations [**?**].

### 1.3.3 MongoDB

MongoDB is an open source database NoSQL database. This means that instead of using tables and rows like an SQL relational database, MongoDB uses collections and documents. Documents are individual data members where there is a key value associated with each document. Collections contain multiple documents. Collections allow the developer to store many items in a database. This structure allows for the structure of the data being stored to be determined after run time. This allows for greater flexibility because you can store dynamically generated objects [**?**].
MongoDB is also fairly simple to set up. This is because of the way that objects are stored we don't need to set up and complicated tables. We can simply set up our data base and insert the data. This also means that if we want to change the structure of the data down the road we don't need to make massive changes to the data base. This makes MongoDB much more developer friendly and also very flexible [**?**].
However, It should be noted that the usability and flexibility of MongoDB is not with out cost. The biggest cost of the flexibility of MongoDB is that it is harder to relate two items in a database. For instance if want to have one entity that represents a user and another that represents a user role, this becomes a challenge in MongoDB. There are work around and ways to solve this problem however it should be mentioned that this is a problem [**?**]. Another trade off in using MongoDB read and write operations are generally asynchronous. This means that you can tell the database to do something and then move on other tasks and the data base will return the result of your query

later. On the surface this sounds nice but if you are doing a lot of reading and writing operations in the same block of code this can cause problems. For instance if you write several items to the data base and then need to read items out that depend on those items you just added you may get errors because the first write hasn't finished yet [**?**]. Finally, MongoDB may also be much slower for some operations than traditional SQL. One example where this becomes apparent are aggregate functions where you want to preform some sort of manipulation of the order of the data [**?**].

### 1.4 Discussion

As mentioned earlier we can divide the types of data we need to store into two distinct groups, weather data and user data. The user data will generally have relationships between different parts of the data. For instance a user will have an address and a phone number. The user will also have various different budgets. These relationships make the data easier to store in an SQL database such as PostGreSQL. To farther simplify storing the data we could also use Python ORM. This would allow us to create relationships in the data, easily develop the database and store the data. The weather data on the other hand is much more like a large list. This sort of data can be sotred in an SQL data base such as PostGreSQL however, this would make the development of the project more difficult. A more suitable choice for storing the weather data would be MongoDB. This is because MongoDB allows for easy storage of lists with little complexity than PostGreSQL. MongoDB provides the easiest and most flexible way to store the weather data.

### 1.5 Conclusion

For this project we will use PostGreSQL and MongoDB. We decided to use these frame works because our client requested it. This is because we are adding to our clients system that has already been created. However, the client did put in some good though into these choices. They chose PostGreSQL because its a powerful relation database management system that will work well for the user data. They also chose MongoDB for the weather data because it provides a simple to set up and easy to use storage tool for storing large lists of data.

## 2 HTTP Request FrameWork

### 2.1 Overview

The *AgBizClimate* project will need a way for the client on the front end of the application to communicate with the API at the back end of the application. We have already determined that we will use HTTP request to facilitate this communication. However, we have not decided which framework we should use to make the HTTP requests with. Generally, the frame work we use to make HTTP requests should be easy to use, easy to read and allow us to quickly right requests.

### 2.2 Criteria

For the purposes of this analysis we will consider ease of development, readability and compatibility with AngularJS. We are considering compatibility with AngularJS because our client has required us to use angular. This will be measured on a scale from compatible to uncompilable. Ease of development will be a subjective measure of how hard it will be to develop software with a framework. This will be measured on a scale from easy to very hard. Readability will be a subjective measure

of how easy it is to decipher the meaning of the request from the code. This shall be measured on a scale from readable to incomprehensible.

## 2.3 Potential Choices

### 2.3.1 jQuery Ajax

When talking about HTTP requests in java script Ajax is probably what immediately comes to most software developers minds. Ajax is one of the original frameworks for making HTTP requests with out updating the page. Ajax allows for asynchronous requests to the backend. Generally, Ajax is not very human readable. Ajax can be read by humans but those humans are going to need to have had experience with Ajax before. It is important to note that Ajax requests will require more work that most of the other HTTP request frameworks we are discussing. Additionally, Ajax can be use with Angular however we will be required to take extra steps to ensure that the requests are handled correctly. If you choose to use this framework with angular you will also need to do more work to be able to test your code.

### 2.3.2 axios

Axios is a promise based HTTP client for web browsers. Axios will also allow us to make asynchronous calls to the backend of our application. It is also important to note that axios HTTP requests are much more concise making them human readable [**?**]. Axios can also be integrated into Angular however, this will require some set up. If you choose to use axios with angular you will need to set up your own custom tests because this frame work will not integrate into angular testing framework. Additionally, you may run into head aches down the road when angular updates are made because axios is not maintained by angular.

### 2.3.3 AngularJS $http

$http is a module in AngularJS that enables communication with remote HTTP servers by using the browsers built in tools. This frame work provides a layer over top of an ajax request that makes the request easier to read and write. These requests are much easier to read and writing the request is much easier that raw Ajax. This HTTP frame work also integrates seamlessly with AngularJS because its a core service. This means that testing these requests with the Angular testing frame work is fairly easy and updates to Angular wont break your HTTP requests [**?**][**?**].

## 2.4 Discussion

In comparing these three frame works I noticed that $http really stood out from the other two frame works. This is because $http is a core service in the AngularJS framework. This makes it much more compatible with Angular than Ajax or axios. Comparing $http and axios out side of the context of Angular they are pretty similar. Both produce much more readable http requests than Ajax and both are are fairly easy to use.

## 2.5 Conclusion

For this project we will use $http to make http requests between the front end of our application and the back end of our application. We chose to use this framework because it integrates seamlessly with Angular while also making HTTP request readable and easy to develop.

# 3 Testing Frame Work

## 3.1 Overview

For the AgBizCliate system we will be required to write unit tests for our code. Unit tests will help us ensure that our application meats the functional requirements. Unit tests will also be helpful for feature projects to ensure that parts of the application have not been broken by a change. In general good unit tests will improve the reliability and extend the life of a project. For this project we want a testing frame work that will allow us to quickly create unit tests in Python for Django Applications. We want to be able to do this because our client has already specified that we will be using Python and Django.

## 3.2 Criteria

For the purposes of this analysis we will consider ease of development, compatibility with Django, and simplicity of setup. Ease of development is a subjective measure of how easily tests can be developed with each framework. Ease of development will be rated on a scale of easy to hard. This frame work will also need to be compatible with Django as this is the frame work our application will be written in. Finally, we will also consider ease of set up. This is a measure of how easy a framework is to set up so we can begin writing unit tests.

## 3.3 Potential Choices

### 3.3.1 Python unittest

Python unittest is a testing frame work built into Pythons standard library. The unittest framework provides the necessary scaffolding to setup, shutdown and run unit tests. This testing framework will feel familiar is it is part of the greater Junit project[**?**]. This makes writing unit tests feel familiar and fairly easy. However, for this application using this testing frame work would require a lot of set up. This is because this is a web application and unittest does not include tools that make it easier to set up and run tests for web applications. More generally, unittest wills take longer to develop because it provides less scaffolding [**?**]. This makes unittest less developer friendly. Unit test can easily be used with Django as it is part of the python library. For this reason unittest is also very simple to set up.

### 3.3.2 py.test

py.test is a popular python testing framework that helps reduce the repletion required in unittest. py.test also contains powerful utilities that make testing many kinds of applications easier. For our project py.test would make testing our web app much simpler than unittest. This would make test development quick and easy. Though this framework makes tests easier to develop that does come at a cost. This framework requires extra setup to integrate with Django. This will make setting up this framework more difficult that either Django.test or unittest. Though the set up isn't complicated it would make testing our application more difficult and more complicated [**?**].

### 3.3.3 Django.test

Django.test is a testing frame work that is baked into Django. This frame work provides powerful tools for testing Django applications. This frame work is part of the greater Django framework. Because Django.test is built into the Django framework, there is very little set up. This also means that there will be no compatibility issues with Django either now or in the future. This testing suite

also provides powerful utilities that make it easier to develop tests. This frame work also gives us specific tools designed to test Django applications [**?**].

### 3.4 Discussion

While considering the options there is one option that fits our application the best which is Django.test. Django.test will allow us to test our Django application with minimal effort as it requires no special set up. Additionally, Django.test also contains tools like py.test that will enable quick and easy development of tests. Though py.test may also be a good option, it doesn't integrate into Django as well as Django.test. Unittest is also another good testing option as it is apart of the python language. However for the purposes of this project unittest would make the development of tests more difficult.

### 3.5 Conclusion

For this project we will be using Django.test to write unit tests for our application. We chose to do this in part because our client asked us to use this frame work. However, our client asked us to use this frame work with the goals for our application in mind. For this application we want to be able to quickly and easily generate unit tests and Django.test is one of the best tools available for testing Django applications.

## 4 References