Oregon State University

CS 461

Fall 2017

Tech Review

Author:
Thomas Noelcke

Instructor:
D. Kevin McGrath
Kirsten Winters

Abstract

The pourpose of this document is to research and consider different technical options for our applcaiton. 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.

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 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.
To analyze our options for our application we will define Criteria by which to compare the different options. We will then use the criteria to compare the three options for our application. Once we have compared our options we will select which option or options we plan to use for our application. We will then justify why we choose that option. If we chose multiple options we will describe which situation we will use one option over the other.

1.2 Criteria

I would like to analyze the performance of the data storage based on, Run Time Speed, Ease of Development, and Ease of Set Up and Flexibility. Run Time Speed will be a concrete measure of run time performance. Ease of development and ease of set up will be a much more subjective measure of 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.

1.3 Potential Choices

In this section we will discuss the pros and cons of each potential choice. I will also discuss each option as it relates to the criteria defined above.

1.3.1 PostGreSQL

PostGreSQL is an open-source relational database management system. PostGreSQL uses the Sever Querying Language. PostGreSQL uses the relational database model. This model sets up tables that represent a certain type of data. This is called an entity. For instance if I wanted to create a database that manged students I would create a students table aka a students entity.
SQL databases are great for run time performance. PostGreSQL will have superior run time performance in comparison to all of the other option listed in this analysis. Generally, If you are looking for a way to store and retrieve data quickly SQL is what you are looking for.
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.
Another problem with SQL in my experience is that it is more difficult to set up and maintain. 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

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.
This type of frame work has several advantages, it allows for easy development, set up and maintenance. 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 Easy. This is because the developer doesn't have to worry about writing complicated SQL statements. This frame work also allows the developer to develop the code first and let the frame work 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 a python database you can store in an object in python you can store in a database. The ORM framework also makes maintaining a data base easier that using raw SQL. This is because the ORM framework has tools for migrating changes to the objects in the python code. This allows the developer to make changes to the models in only on spot. This also means that the Developer doesn't have to worry about how to make the changes on the models work in the database.
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

1.4 Discussion

1.5 Conclusion

2 HTTP Request FrameWork

2.1 Overview

2.2 Criteria

2.3 Potential Choices

2.3.1 Ajax

2.3.2 XML Http Request

2.3.3 choice 3

2.4 Discussion

2.5 Conclusion

3 Testing Frame Work

3.1 Overview

For the AgBizCliate system we will be required to write unit tests for our code. In this section I will discuss the various testing frame works we might use for our project. I will restrict the conversation to frameworks that we can use in Python as parts of our clients application are already in python. It is important for use to write tests for our code for several reasons. Firstly, We will want to ensure that our code actually does what we intended. Secondly, We will want to make sure that when we make changes we do not break our code. Thirdly, Our unit tests will become useful when we want to do integrated builds for our project.

3.2 Criteria

3.3 Potential Choices

3.3.1

3.3.2 Choice 2

3.3.3 choice 3

3.4 Discussion

3.5 Conclusion

4 References