At first I thought the advantages of the Django Rest Framework were substantial enough to support reasoning for using it to build out the APIs. However, as I continued to research I began to more fully understand what factors MongoDB brought to the table. Django is friendlier with relational databases and so I began to realize for this assignment it may not be the best fit. Along with that came the realization that performance would also likely take a hit. After reading more about the dynamics of the technologies I decided that using a more lightweight python framework like Falcon or Flask alongside MongoDB made more sense.

Per the assignment document, because the ideal technology to power Yewno would be high performance and be highly available, I in turn selected Falcon. Falcon is a high-performance Python framework for building APIs. It encourages the REST architectural style, and tries to do as little as possible while remaining highly effective. Falcon offers reliability and freedom while enabling more requests on the same hardware. It is designed to be light, fast, and flexible. Given what needs to be achieved in the assignment along with no mention of functionality like user authentication, an ORM, and so on I think it is the perfect fit. Falcon is highly optimized for the exact tasks listed in the API of the assignment.

When Falcon runs on top of PyPy which is a more optimized version of Python it beats out the older deployment of running on top of CPython.

PyPy 2.5.1

Framework	req/sec	µs/req	Performance
Falcon (0.3.0)	256,417	4	27x
Bottle (0.12.8)	145,411	7	15x
Werkzeug (0.10.4)	44,276	23	5x

Pecan (0.8.3)	16,031	62	2x
Flask (0.10.1)	9,471	106	1x

I used PyMongo to interface between Python and MongoDB. I used a .js script to load my data into the MongoDB as well. I created the data structure containing the data using a python script that ingested a .txt file.

To create the MongoDB from the client I simply did:

```
mongo 192.168.99.100:32769/db journals.js
```

From there I began to learn about Docker and getting that VM set up with Falcon, Mongo, Redis, etc. Once I had the environment established I began to write my Falcon code and thus the API. Following writing the API I wrote some tests in an effort to test the API. Not all API endpoints can be ensured to work mainly because I was not able to writ sufficient test scripts. I also was not able to break the 2 APIs into 2 projects due to limitations in my abilities with Docker and sharing Mongo for example. I thus was not able to attempt building HAProxy or Ngnix to map the Ports.