**Gymnastics Scoring Software**

The purpose of this paper is to review different technical options for the proposed gymnastics scoring software.

**Mary Jacobsen**

Tech Review

Group 18

10.0 Software

# Introduction

## Accomplish

The gymnastics meets held in Gill Coliseum are fast paced, usually televised competitions attended by thousands of fans. In order to run the competitions smoothly and provide a positive experience for fans, athletes, and staff, there must be a way to record, organize, and display the scores that the judges give to each routine using the hardware already in Gill Coliseum. The Oregon State Gymnastics Team would like new scoring software. With the addition of a new jumbo-tron inside of Gill coliseum, the pre-existing software used to display and keep scores has become outdated and has issues displaying on the current hardware. Because of this, it is important for new, custom software to be built that can accurately and elegantly display scores on Gill Coliseum's hardware in a readable manner that is intuitive for fans to understand while providing all the necessary functionality such as exporting to specific formats.

## Role

My role in the gymnastics scoring software project is to work on the web portal and possibly the API as well. There will be other members working with me on the web portal and API. These roles require research to be conducted on containers, frameworks, and languages.

## Tech Options

This tech review compares different options for the design of the gymnastics scoring software, 10.0 Software. It looks at three options for containers, three options for frameworks, and three options for languages. Either Docker, Open Shift, or Kubernetes will be used if it is decided upon to use a container. Either Django, Node, or Beego will be used depending on the selection criteria for each framework described. And finally, one of Javascript, Python, or Go will be chosen as the language to develop in.

# 1. Containers

## 1.1 Docker

Docker was developed from LXC, which allowed multiple Linux systems to run on one computer isolated from the other systems. Docker is an open source project that was made to make an LXC container for a single application. Docker is now a Linux container runtime environment that can make, deploy, and run a container. [11] As of recently, Docker is getting a lot of attention. There are many reasons to use Docker. Docker is the current industry standard. It is the main reason containers are currently so popular in the software engineering industry. Docker is lightweight. Instead of requiring an operating system for an application, it shares the operating system kernel with the computer. Docker is secure and trusted. The application will be safe inside the Docker container. [5] Docker provides all the benefits of an extra layer of abstraction without very much overhead at all. Docker also allows for portable shipment of applications. Docker is focused primarily on the application as opposed to the machine or other things. Another convenient part of Docker is that it provides source control which doesn't affect the decision of using it for this project because using Github is required but it is a very good feature anyway. [1] Because of all these features Docker provides and because it is trusted by so many, Docker seems like the safe choice.

### 1.2 Open Shift

Open Shift is an application container platform. It is a different option for the use of a container but it still can utilize Docker containers. It is not as well known or as commonly used as Docker but it does have some appealing features. One of the more exciting parts of Open Shift is that it is built to work along side with Github which is ideal for the project since Github must be used. Not only that, most software developers are very familiar and comfortable with using Github so it makes the learning curve of Open Shift not as steep. Open Shift makes it easier to deploy applications because applications can be deployed using git push or pushing a button. Another helpful feature of Open Shift is that it has templates for a few platforms and languages including Node and Python. [9] Open Shift is free to use for just one project so that would suffice. After some research, it does not seem like the features added by using Open Shift are worth the extra time it would take to set it up. The Open Shift features are meant to save time but they may not save enough time to choose this option.

### 1.3 Kubernetes

Kubernetes is a Google open source project that manages containers. Using a container is not one of the requirements for this project but a container may be used if the benefits outweigh the work of setting it up. Containers at Google, like other containers, provide an application environment that does not change, as well as the ability to run applications anywhere, and a layer of abstraction from other parts of the machine and other applications. Containers also have a very small over head and do not take up much memory. Kubernetes automates the process of managing a container while still providing all the benefits of using a container. Kubernetes runs best on Google Cloud Platform because it is integrated in but can be run elsewhere. Kubernetes' "goals are to build on the capabilities of containers to provide significant gains in programmer productivity and ease of both manual and automated system management." [6]

## 2. Frameworks

### 2.1 Node.js

This project will definitely require using a framework due to time constraints and because it is more efficient. Node.js is an open source, server side run-time environment built on the Google Chrome's JavaScript V8 engine. Node.js works on the main operating systems (OS X, Windows, and Linux). Node.js is event driven and asynchronous. That will help with waiting for data to be returned. Node.js is also very fast and won't slow down the code execution. Node.js is single threaded but because it is event driven, it is still scalable and will suffice for this project. The Node.js applications return the data in pieces so it never has to buffer any data. Node.js also partners well with JSON APIs which might be an option for this project. Node.js is also very trusted. Some of the companies that use Node.js are Uber, eBay, Microsoft, and Paypal. [12]

### 2.2 Django

Django is a web platform based on Python that is open source and free to use. The three main reasons developers usually choose to use Django and why we might choose it is its speed, security, and scalability. Django supports very fast development. Django also helps with having sufficient security. And because of its superior scalability, "Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale." [3] Django has some added features that would help specifically

with this project. One of the more helpful extra features that Django includes is that it takes care of authentication. That is important for this project since it requires different levels of authentication to allow only authorized staff to input scores. Django is also a well trusted, popular, and versatile framework. ”Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.” [4] Django would definitely work for this project so it really depends on if Python is chosen as a language since the framework and language decisions really depend on each other.

### 2.3 Beego

Beego is an open sourced application framework built on the Go language. Beego is one of the more commonly used frameworks for the Go language which is growing rapidly in popularity. There are four main reasons why developers would choose Beego. Beego is easy to use, intelligent, modular, and has high performance. Beego has automated testing, packing, and deploying. Beego has some interesting monitoring features as well. It can monitor memory and CPU usages as well as other things. Beego works for most types of applications. ”With powerful built-in modules including session control, caching, logging, configuration parsing, performance supervising, context handling, ORM supporting, and requests simulating. You get the powerful foundation for any type of applications.” [2] Applications built with Beego can also handle lots of traffic which does not specifically apply to this project unless the IOS and Android applications stretch goal is completed. If the language Go is decided upon, the Beego framework would be a good choice for this project.

## 3. Languages

### 3.1 Javascript

This project will obviously require the use of a language and whatever language is chosen will affect which framework will be chosen. Javascript is used for so many web applications and would be an easy choice with the use of Node.js. There are many aspects of Javascript which make it an appealing option but also some that do not. It is not as easy to use as other languages such as Python and Go. It is built into the browser which eliminates set up time which is probably negligible for this project but it is still convenient. Javascript is very widely used. It is used in web browsers, server side, mobile, desktops, games, and more. The reason for this is because it is ”the native language of the web browser” and not necessarily because its the best choice. [10] Javascript does work really well for forms. If Javascript is used correctly, it can enhance a web page and make it friendlier for the user. [13] Overall Javascript makes sense to choose because it is so well known and used and because of the quality of Node.js but there are definitely possible downsides to using Javascript as well.

### 3.2 Python

Python is considered an easy to use language by most developers. ”Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax.” [8] Python also has many libraries that can be included. Python is extremely versatile and is used in many different types of applications. Due to the large standard library, Python is very good for working with strings unlike many other languages which makes code much simpler. Also included in the standard library are ways to easily use internet protocols such as HTTP and XML which will be needed for this project. [8]

Python also has very good documentation which is so important for finding the best ways to use the language and solving problems. Overall, Python seems like the easiest language to use.

### 3.3 Go

Go was created at Google and is an open source language. ”Go was actually developed as an alternative to C++.” [7] Go was designed to better serve software engineers and not to research programming languages. Go is somewhat similar in syntax to C and is therefore easy to learn for developers who already know C. Go has many strengths since it was built to solve current software engineering problems. Go is statically typed which C/C++ developers can really get behind. Go has a very fast compilation speed and execution speed which was one of the big problems that the language was created to solve. The code compiles directly to machine code which not only makes the execution speed very fast, but it also makes it portable. Go has automatic garbage collection so pointers don’t need to be freed unlike C. Go has many good libraries that can be included that make things easier for the developer such as HTTP client and server implementations, SQL databases, and JSON. ”JSON is treated as a first class member of the standard language.” [7] Go would be a good choice because of all the listed selection criteria and because it is an exciting language to learn.

## Conclusion

After researching the three pieces that relate to my role in the 10.0 software project and each of the options for each piece, the reasons for choosing each of the options are far more clear. For containers, Docker seems to be the clear choice because it is lightweight, secure, and trusted. Kubernetes is a good choice to manage the Docker container because it is easy to use and automates much of the process. For the frameworks, it really just depends on which language is chosen because Node.js, Django, and Beego all seemed usable. Which language to use is not as clear of a choice. The clear choices are either Python or Go because Javascript is not as simple and easy to program in. Both Python and Go have great libraries and are easy to learn and use. This research will give the group the necessary information to make these decisions.

## REFERENCES


[1] Aater Suleman. *https://www.airpair.com/docker/posts/8-proven-real-world-ways-to-use-docker*.

[2] Beego Framework. *https://beego.me/*.

[3] Django Project. *https://www.djangoproject.com/*.

[4] Django Project. *https://www.djangoproject.com/start/overview/*.

[5] Docker. *https://www.docker.com/resources/what-container*.

[6] Google. *https://cloud.google.com/containers/*.

[7] Kanishk Dudeja. *https://hackernoon.com/the-beauty-of-go-98057e3f0a7d*. Hackernoon, 2017.

[8] Python. *https://docs.python.org/3/faq/general.html*.

[9] Red Hat. *https://www.openshift.com/products/features/*.

[10] Richard Kenneth Eng *https://medium.com/javascript-non-grata/the-five-top-reasons-to-use-javascript-bd0c0917cf49*. 2016.

[11] Roderick Bauer. *https://www.backblaze.com/blog/vm-vs-containers/*. Black Blaze, 2018.

[12] Tutorials Point. *https://www.tutorialspoint.com/nodejs/nodejs*$_i ntroduction.htm$.

[13] Stephen Chapman. *https://www.thoughtco.com/why-javascript-2037560*. ThoughtCo, 2018.