CS Capstone Technology Review and Implementation Plan

November 13, 2017

NASA University Student Launch Initiative

Prepared for

Mechanical Engineering, Oregon State University NASA

Dr. Nancy Squires _____ _____
Signature *Date*

Prepared by

Group 33
USLI CS Payload Subteam

Mark Bereza _____ _____
Signature *Date*

Joseph Struth _____ _____
Signature *Date*

Kevin Turkington _____ _____
Signature *Date*

Abstract

This document is written using one sentence per line. This allows you to have sensible diffs when you use LaTeXwith version control, as well as giving a quick visual test to see if sentences are too short/long. If you have questions, "The Not So Short Guide to LaTeX" is a great resource (https://tobi.oetiker.ch/lshort/lshort.pdf)

CONTENTS

1 ROVER

1.1 Collision avoidance

The rover itself is a key component to the University Student Launch initiative. As it provides Oregon State University to produce a given experiment that will be contained within the launch vehicle. The sole purpose of this experiment is to move autonomously away from any piece of the launch vehicle at a minimum of five feet while deploying solar panels upon reaching its final destination.

1.1.1 Option 1

The first and most simple option for the rovers collision avoidance, is to rig the rover solely with bumper style sensors. Similar to the Tekbots used in Oregon State Universities ECE 272 and ECE 375 labs. After deployment the rover would simply move forward in a single direction away from the rover. The bumper sensors would assist when the rover simply runs into an object. Upon impact a set of instructions would be run turning the rover slightly to either the left of the right depending on the bumper that was activated. Additionally sub routines could be introduced making the rover move completely in a different direction upon three consecutive bumper activations. However this method would not guarantee the rover has moved at least five feet from any part of the launch vehicle.

1.1.2 Option 2

The second more complex option for rover collision avoidance is to include a type of non contact sensor. Such as SONAR, LIDAR, or RADAR. Depending on the sensors chosen the detection ranges can range by several feet. The benefits of using a non contact collision avoidance system is that it can be paired with ROS (The robot operating system) to dynamically creating mappings of its surroundings. These mappings can detect if the rover were to move into a dead end area. Addtionally the scope of the area the rover could view would be a 60+ degree cone from the forward facing direction of the rover. Allowing it to detect multiple objects at once, enabling it to weave throughout them seamlessly. As opposed to ramming into objects in its path and turning 90 degrees in the other direction (the method the simple bumpers would use.) Moreover SONAR, LIDAR, and RADAR sensors have a significantly smaller form factor to their bumper actuator cousin in Option 1 for rover collision avoidance.

1.1.3 Option 3

The most complex option in rover collision avoidance as well as destination mapping, would be fitting the rover with a GPS module sending itself coordinates of its current position

2 AVIONICS

2.1 Option 1

blah blah

2.2 Option 2

blah blah

2.3 Option 3

blah blah

3 WEBSITE

3.1 Overview

Aside from the launch vehicle and payload itself, the website serves as a the main medium of giving deliverables to NASA and an excellent resource for potential USLI members to join and gain information. The website itself must be easily refactored by computer science and non computer science members of team in case of emergency updates, as well have very little down time in between updates.

3.2 Front-end

Overall the front end technologies are the most important part of the website because of these three key factors: maintainability, ease of use, and refactorability. Anything displayable on the website that the end user NASA or OSU students can view must be easy to maintain as well as easy to change at any time. The website can change due to specific team leads needs and wants for the website since it is a vital component of OSU USLI's documentation delivery to NASA and subject to the USLI competitions Best Website award.

3.2.1 Static HTML

One of simplest and easiest ways to put up a website is to statically serve pure HTML and CSS. This option is simplistic and wouldn't require much of the USLI computer science students to do, while also allowing other USLI team members and volunteers to contribute to the web project. The main draw back to using static HTML is the minimal degree of functionality. This option limits the page to purely serving content without the ability of enabling outside libraries like JQuery for Dom manipulation or relatively more complex POST/GET requests to the server side than simple HTML forms.

3.2.2 React

On a higher level of web development React can be used for a clean method for DOM manipulation. React excels with its ability for easily maintainable and readable interface code, However there are some large draw backs to the framework. For example React is best used in combination with Typescript a subset of Javascript created by Microsoft. This language requires the use of multiple libraries like Webpack from previous experience takes time to setup correctly. Additionally most underclassmen as well as seniors have little to no Web development experience let alone experience with React, leaving those who exclusively wanted to contribute to the teams website a significant hurdle to jump over in terms of learning the React framework.

3.2.3 Handlebars

The middle ground between static HTML and React would be the use of Handlebars or a Handlebars like templating engine. The benefits of using a templating engine is if the DOM on load is affected by passed in data from the server (this is most useful for applications utilizing CRUD operations for a database back-end). Additionally Handlebars is widely taught at Oregon State University's web development classes providing web development experience for underclassmen about to take the web development class as well as after. The primary drawback of this framework is the OSU USLI website would not utilize any sort of back-end database.

3.2.4 Conclusion

Since the goal of the USLI website is ease of maintainability and accessibility to other members of the team, the best options for front end development would be a mix of Handlebars and Static HTML. Because the website mainly serves non dynamic content and these technologies are taught by professors as Oregon State University.

3.3 Back-end

The main purpose of a websites backend is to act as the main component for serving web pages to the end user, as well as a a mid point for any database that the end user may need access to. In context to the OSU USLI website the server side will solely be used for content serving for the end user (NASA and OSU USLI team members.)

3.3.1 Django

The easiest option in terms of setup out of the three options for frameworks would be Django because of its extensive documentation, functional and easy to use language (python), and vast collection of libarays to choose from for templating, database, etc. And the ability for projects to be setup immediately with only a couple commands to the terminal. However the frameworks largest problems is the painfully slow and inefficient processing times compared to its alternatives. As well as none of the OSU USLI computer science students have used the platform nor have any of the computer science underclassmen, requiring a small amount of research and tinkering of those working on the website.

3.3.2 NodeJS

The NodeJS framework offers alot of support and documentation for developers indicating an excellent choice for the OSU USLI team. NodeJs's primary language is JavaScript which is loosely typed allowing for quicker development of all simple server side processes such as routing and simple data manipulation. However if library are used most tend to be asyncronous requiring knowledge and development experience using promises within the NodeJs framework. as well as debugging can be trouble some in JavaScript because the language was developed in a short time frame and is loosely typed.

3.3.3 Dotnet Core

Dotnet Core is the most efficient of the three frameworks because of its primary language C#. Additionally the framework offers cross platform development allowing any person on the OSU USLI team the ability to help develop whether it be on Mac, Windows, or Linux. Any allows for any web application to be easily scalable if the website where to gain a significant amount of views end users over the internet. However the platform does require sometime to setup and a little bit of a technical hurdle for underclassmen to learn C# as well.

3.3.4 Conclusion

Because of its large amounts of documentation and ease of development NodeJs will be our choose platform for back-end web framework. Again this was the best option because this is the primary framework taught by Oregon State University professors and is widely used within the web development for smaller scale project intend for little growth.

3.4 Hosting Services

Because its unreliable and costly for individual members of the OSU USLI team to host the team website with a personal server the following choices were devised:

3.4.1 Amazon Web Services

the Amazon Web Services(AWS) is a great platform for developers to have complete control over their web application in every way. Allowing the developer to choose what type of operating system will be used to host, and customized to any configuration. AWS also supports ease of scalability with different payment plans. The drawback to this option is although it may be cheap the system will need to be tested and updated at least once every month for content changes to the website as well as simple operating system updates.

3.4.2 Heroku

Heroku works similarly to Amazon Web Services how ever some of the operating system configurations are obfuscated to the developer leaving less room for customization. However this is not a big factor for the OSU USLI website because of its simplicity. Additionally the platform offers a Git like interface for updating the codebase quickly, and can provide a great learning experience for computer science underclassmen on the USLI team. The drawback is the free option on herokus platform form will only run the web app for 1000 hours out of the month and will make the sever sleep after 30 minutes of inactivity. Leave room for the website to potentially be offline when NASA checks the website.

3.4.3 Github.IO

Github IO is the most promising out of all the platform choices because of free unlimited hosting of a single website under a given project. Additionally the codebase would be stored with the rest of the teams work under the OSU-USLI-18 Github Organization leaving any member of the team access to the website at anytime. This platform also obfuscates any and all server side code leaving limitations in terms of server side data handling. However since the website only needs to serve the html and pdf files Github IO is the perfect option.

3.4.4 Conclusion

The website is intended to only be active for the duration of the competition. Therefore any long term or complicated hosting solutions are not required. The main quality for hosting services is robustness and ability to be updated quickly. Github.IO provides the best of both those qualities because the website is directly tied to the source control itself, the pages are automatically updated with each pull request, the website will always be online as long as Githubs systems are online, and the back-end is abstracted from the developer leaving more time for higher priority projects.