CS CAPSTONE TECHNOLOGY REVIEW

27 OCTOBER 2017

ROBOTIC WHEELCHAIR DATA COLLECTION AND ANALYSIS

PREPARED FOR

OREGON STATE UNIVERSITY

MATTHEW WILLIAM SHUMAN _____ _____
Signature *Date*

PREPARED BY

GROUP 4

WHEELCHAIR DATA COLLECTION TEAM

HADI RAHAL-ARABI _____ _____
Signature *Date*

Abstract

This document will describe the technology that will be utilized in the development of the user-interface portion of the Wheelchair Data Collection Project. The technology discussed is broken up into three sub-pieces, the server back-end, the server front-end, and the webpage style. When combined, these technologies will be a web-based graphical user interface that a tester can use to collect data during the wheelchair operation trials.

CONTENTS

1 Server Back-end

1.1 Introduction

The first sub piece of this project is the backend webserver portion, which will provide the foundation on which the user interface will sit. This review considered three options: apache, nodejs, and nginx. Each webserver solution has different drawbacks and advantages, but all are feasible to implement, and any could be compatible with the other pieces of technology discussed in this review.

1.2 Apache

Apache server is extremely compatible with PHP for server-side scripting. The typical apache server stack is referred to as LAMP (Linux, Apache, mysql, PHP). Since the other components of Project Chiron will require a database like mysql, and the LAMP stack is ubiquitous enough to have several resources for configuration, apache server would be a natural candidate for the server-side scripting portion of this project.

1.3 Nodejs

Nodejs allows for both back-end and front-end scripting to be done in JavaScript, which will provide maintainability and longevity to this project by limiting the amount of required expertise. A popular argument for nodejs is that it is easily scalable, but that isnt an important advantage in the scope of this project. Node provides a simple package manager, npm, which will make installing and managing dependencies and libraries exceptionally simple. Furthermore, nodejs will provide portability, as the server can be run in multiple environments without changes to the code. However, to run the server in different operating systems, we will need custom system-level calls for each environment that the server will run in. A workaround for this is to deploy the node server in a docker container, a process that is quite simple and highly documented for nodejs.

1.4 Nginx

Ngnix (pronounced engineX) is a web server renown for its load-balancing and scalability. It is well suited for large scale enterprise environments. Nginxs stability while scaling comes from its low memory footprint, allowing many more active simultaneous connections per gigabyte of RAM than the competitors listed in this review. The webserver was specifically written to outperform Apache with less resources, and can handle up to four times more requests per second than an apache server on identical hardware. When developing a web-based application that is resource-limited and must support many user connections, nginx is the obvious choice.

1.5 Review

For this project, we will be using nodejs. While the maturity of apache server would provide many learning resources, it would limit the server to running in a linux environment, and additionally add PHP as a technology necessary for this project. Although Nginx would scale well and be efficient with resources, these advantages are not as desirable for this project as simplicity and maintainability. Nodejs will provide several advantages, the most significant of which is not technical: it limits the web scripting portion of project Chiron entirely to JavaScript. Any other webserver, including those not listed in this document, would require configuration and development in a new language, even though invariably the front-end of the application would require some JavaScript. This would mean additional complexity in development during the project, and complexity in maintenance after the project is complete.

2 SERVER FRONT-END

2.1 Introduction

The front-end scripting portion of the project will be small in scope, but still warrants a discussion of available technology. Since the user interface will be a webpage, the front-end logic will be entirely in JavaScript. However, there are extensions to native JavaScript that can be utilized for additional functionality. The two extensions that will be discussed are Angular and React. There is a multi-dimensional problem with selecting a technology for front-end development however, as Angular and React provide significant functionality, but this functionality is only necessary if the front-end will be doing significant amounts of logic and data-management. So selection of a front-end technology may significantly impact the architecture of our interface.

2.2 JavaScript

JavaScript is an interpreted programming (scripting) language for websites and web-based applications. It is the foundation of all of the front-end scripting options that will be discussed in this document. JavaScript code can be embedded directly into the markup of a webpage, or it can be written separately and referenced. It is compatible with all modern browsers, although there is occasionally a case in which a fringe functionality is only available on some browsers. JavaScript will facilitate all interactions between the front-end and the back-end of our web application, but the syntax behind these interactions may vary depending on whether the frameworks discussed below are adopted.

2.3 Angular

AngularJS is a JavaScript framework developed by Google. It provides simple methods of creating responsive webpages, which would otherwise require significant amounts of code in native JS. Angular utilizes the model-view-controller (MVC) architectural model. In MVC, the model is the central component which stores data and contains logic. The view is the user-interface. The final piece, the controller, accepts input from the user, and passes it to the controller to execute logic, or to the view to render a change. Angular provides the MVC structure to JavaScript, where normally native JavaScript is more unstructured and does not adhere well to large-scale front-end projects.

2.4 React

ReactJS is a JavaScript library developed by Facebook. Like angular, it utilizes the MVC architectural model. However, while angular is a framework that requires a specific syntax, react can instead be seen as an extension of JavaScript with a smaller learning curve.

2.5 Review

For this project, we will native JavaScript. While React and Angular would provide significant functionality on the front-end, we will be keeping most of the application logic on the back-end for the sake of simplicity, as housing significant amounts of logic in two different components of our interface negatively affects maintainability. Additionally, if the project is extended and used by Permobil after project Chiron, keeping the logic on the backend will provide additional data security. However, since the stretch goals of this project may involve significant front-end logic for data modelling, it is important to note that Angular may eventually be installed on top of the existing JavaScript. In this case, we would choose Angular over React since the strict structure of the controllers would assist in managing a large set of dynamic views.

3 WEBPAGE STYLE

3.1 Introduction

Webpage style, such as element positioning, size, and color are managed by Cascading Style Sheets (CSS). However, like JavaScript, writing CSS natively can be a painstaking process. Dynamic style is necessary for responsive webpages, and writing individual pieces of logic in JavaScript to change CSS based on user input is very time consuming. For this reason, there are style frameworks for webpages. The frameworks that will be discussed in this tech review are Bootstrap and Skeleton.

3.2 CSS

Using native CSS does have advantages for small projects. It is entirely compatible with both modern and outdated browsers, and it is extremely unlikely that a page styled with CSS will appear differently across browsers. Additionally, assuming that aesthetic is important, native CSS has the quickest initial set-up time, as it requires no dependencies and can even be embedded directly into the markup of a webpage.

3.3 Bootstrap

Bootstrap is an open-source web framework for managing style, originally developed by Twitter. Bootstraps primary selling point is that it is incredibly mobile-friendly. Pages styled with bootstrap will automatically scale to the size of the screen they are displayed on. Bootstrap also provides a style-set including buttons, forms, typography, and navbars. A quick install for bootstrap is available in the node package manager, and it is highly compatible with both Angular and React.

3.4 Skeleton

The Skeleton CSS boilerplate is named for its barebones approach to CSS. The entirety of the framework is 400 lines of CSS, and most of the functionality comes from the Skeleton grid. The grid provides a simple method of arranging elements on a webpage into a form of table. The appeal of Skeleton is that it is incredibly quick to stand-up a page with a professional aesthetic, but there are limited options for customization without a significant amount of custom code. Skeleton is an appealing option if the goal is to very quickly develop a clean webpage style, without requiring the installation of any package or framework.

3.5 Review

Although style is not a critical component of our project, it is important that the page appears polished. A well designed page will improve ease-of-use, and a professional appearance improves the likelihood of adoption of our interface by Permobil. While there are merits to each technology for styling our webpage, we will be using bootstrap. Creating a professional aesthetic using barebones CSS will be significantly more time consuming, and the Skeleton extension simply does not provide as many options for modifying the style of the page. Bootstrap will provide many sets of responsive and attractive buttons form options, which is critical since most of the style will be written for an interactive form. It has not yet been decided where the webserver will be hosted, but if it is not running locally on the machine of the user, bootstrap will also allow for the testers to be running trials directly from their mobile devices, eliminating the need for a desktop or laptop entirely.