AA Gymetrics

Arin Aihara

CST-451 Capstone Project Final Architecture & Design

Grand Canyon University

Professor Mark Reha

Revision: 1.0

26 October 2022

**ABSTRACT**

In modern times, mental health of individuals has been on the decline. As more people are diagnosed with a variety of mental and emotional disorders, more attention comes to what can be done to improve mental health. As a result, many people turn to a variety of methods to cope, both healthy and unhealthy. One outlet that many turn to is working out. As a healthy alternative to many unhealthy forms of coping, working out regularly, promotes the advancement of physical health while also supporting a healthy form of self-care, which can in-turn, improve mental health and wellbeing.

As GCU expands, more students are on campus, leading to higher numbers of students utilizing the fitness centers on campus. Although there are quite a few fitness centers available, each individual may have preferences due to machines or tools and their availability in the gym. As aforementioned, with higher numbers of students, there are also certain times with higher concentrations of students working out simultaneously. This project aims to provide students with up-to-date information of the number of students currently in each fitness center. There are a number of students who complain about the crowded gyms on campus. With the high unlikelihood of GCU building more dedicated fitness centers, the hope of this project is to allow regular gym-goers to know in-real time, how crowded a gym is to help them make a decision on when and where to go to work out.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 10/26/22 | Arin Aihara | Initial draft for review/discussion |
| 1/9/23 | Arin Aihara | Update of diagrams to match current functionality |
| 4/5/23 | Arin Aihara |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Design Overview 5

Detailed High-Level Solution Design 6

Detailed Technical Design 9

Appendix A – Technical Issue and Risk Log 18

Appendix B – References 19

Appendix C – External Resources 20

Design Introduction

With the increase in on-ground students at GCU, a corresponding higher number of students are actively utilizing the university’s fitness centers. For those who like to work out without bumping shoulders with people or waiting for a machine to free up, this can pose a problem. As of now, there is no system of keeping track of the live metrics of people in each gym, which can make working out a guessing game when deciding which gym to go to, and at what time. GCU currently employs camera systems in the campus restaurants to keep track of the line, but there is no similar system in the gyms. For obvious reasons, a camera system would pose privacy issues, but there is no reason to completely leave gym-goers in the dark on how busy and crowded a given gym is. Given that everyone has varying schedules, there is no reason that there isn’t a time that wouldn’t work for someone to use the fitness center when it isn’t crowded. The problem is, knowing when that time is, and which gym the time applies to. To solve this, the development team proposes AA Gymetrics.

|  |  |  |  |
| --- | --- | --- | --- |
| External Documents | | | |
| ID | Deliverable Description | Evaluator (internal or external as applicable) | Status |
| 1 | Capstone Proposal | External | Complete |
| 2 | Capstone Requirements | External | Complete |
| 3 | User Stories Document | External | Complete |
| 4 | API Documentation | External | Complete |

Detailed High-Level Solution Design

As seen in the diagram provided below, the development team will be utilizing a number of technologies to provide a solution to the issue mentioned before. AA Gymetrics will be designed to provide information to students to help students decide when and where to go to work out an any given time. A single page app for non-employees will display metrics for each gym on campus, allowing students to stay in-the-know regarding the crowd and current “headcount” (so-to-speak) of each gym. For employees, a secure login will be provided to show another page which allows them to check students in and out of the gym, which will update and reflect the changes in the headcount in real time. The coding environment being used will be Visual Studio Code v1.73.0. Utilizing JavaScript in the context of the Vue v3 framework as well as Tailwind v3.1.8 for CSS purposes, will allow for a reactive and dynamic frontend experience for users. Vue and Node v16.17.1 will also be utilized for the backend services. Postman v10.1.1 will be used to test the API services that will be utilized for the CRUD functions required for this project. Finally, MongoDB and MongoDB Compass v1.33 will be used to house the database and tables required to support the project’s need for data storage.

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | | |
| **Description** | **Rationale** | **Results** |
| 1. Udemy Vue Course Job Search App | Provide insight into Vue framework | In-progress (near completion) |

|  |
| --- |
| Hardware and Software Technologies |
| 1 – USB Handheld Barcode Scanner |
| 2 – Vue.js v3 |
| 3 – Tailwind CSS v3.1.8 |
| 4 – NodeJS v16.17.1 |
| 5 – MongoDB v6 |
| 6 – Postman v10.1.1 |
| 7 – Mongoose v6.8.0 |
| 8 – Express v4.18.0 |
| 9 – Axios v1.2.0 |

**Logical Solution Design:**

**Diagram

Description automatically generateda**

The diagram above shows the different layers of the web application. As highlighted in the frontend services, there are templates, which provide the outline for each component. The components serve as pieces of or the pages themselves, depending on the utilization of each component. Among the many components that will be created (as highlighted further down in this document), there is a navbar component, a button component, and a popup window component. With these components, there is the opportunity for reusability to be implemented across pages. Computed properties allow for real-time logic to affect the page dynamically from user input, or a change in the information provided. The backend services are supported Node and will support the use of a REST API to pull data from the Mongo database.

**Physical Solution Design:**

Diagram

Description automatically generated

The web application will be hosted locally, with cloud hosting being an out-of-scope feature for the time being. The web app will utilize HTTP port 8080 to access the app itself, and the port 3000 will be used by Axios to provide a backend connection to the API, which will allow the team to pull data from the Mongo database. Cloud integration will be done using Heroku’s PaaS to make for easy deployment of the application. It can be found at aa-gymetrics.herokuapp.com. To monitor the uptime of the website, UptimeRobot is utilized, sending alerts when there is downtime to the website.

Detailed Technical Design

**General Technical Approach:**

The goal of the web app is to create an easy to navigate page for users to quickly obtain information on the live metrics of the gyms. To do this, a reactive and dynamic single page application will be utilized to give the user a quick and accurate experience. To do so, the development team decided to utilize JavaScript in the context of the Vue framework. This will be supported by Node for backend development and will provide a smooth experience for users and employees alike.

**Key Technical Design Decisions:**

To support a reactive and dynamic website, the development team decided to use Vue.js 3 as a JS framework as means to create and structure the web application. NodeJS will be used for backend development, and MongoDB will be used to store the data needed to support the website. The decision to use NodeJS reflects the development team’s desire to continue to utilize JavaScript as means to support the application. Since the data being stored isn’t relational, there is no need for foreign keys and key aspects of a relational database that would be commonly seen in MySQL databases. As another key decision, the usual register page that would be commonly found in most web apps has been omitted to prevent unauthorized students from creating privileged accounts. This will ensure that pre-existing authorized employees will be the only ones able to access the employee-end of the website. Finally, a piece of hardware will also be utilized in the application, for scanning IDs. A barcode scanner reads the data off an ID and provides student information that will be used to update the web application.

**Database Schema Design:**

Graphical user interface, text, website, timeline

Description automatically generated

The database will comprise of 4 different tables, allowing for the storage of students, employees, as well as keeping track of active students in the gym, and a table to keep track of login history for employees.

**Sitemap Diagram:**

**Diagram

Description automatically generated**

The sitemap portrays the web application’s navigation. From the landing page, the login and about page are accessible via the navbar. The navbar can also be used to go back to the home page from any of the other pages as well. Although other windows will show up such as popups, since navigation is the priority of this diagram, only the navigable pages are displayed.

**User Interface Diagrams:**

Graphical user interface, table

Description automatically generated

After navigating to the website, users are met with the initial landing page. Although able to access the login page, without a valid login, no action will be made, to keep non-employees out of the employee-end views. Other sections of the website are made accessible through the navbar at the top of the page. Employees that login will be authenticated and will be met with error handling if there is a problem during login, as well as data validation during the entering of their email.

Table

Description automatically generated with medium confidenceAfter successfully logging in, employees will be directed to the page for the associated gym that their shift is scheduled for. As an employee, they can check students in manually, if necessary, as well as check out students and logout. Checking students out will require confirmation, while logging out as an employee will not, since it utilizes a dropdown to access the logout. Again, other pages are accessible via the navbar such as the “Home” page or “About” section.Diagram

Description automatically generated with low confidenceThe last page not covered is the About page. Notice that the navbar options vary depending on the status of logged in or not.**Components Diagrams:**

**Graphical user interface, application

Description automatically generated with medium confidence**

The base home page has a number of components that will be reused in other pages, such as the Navbar and the Button component. The Navbar will stay visible throughout the app. The Button will be utilized throughout the application and will has preset styling. The component itself also serves as a component to limit the code present in the actual view page.

**A picture containing table

Description automatically generated**

Similar to the home page, there are numerous reusable components, such as the dropdown menu for logouts and individual entries for each student. The entire page is also its own component, which will compile multiple reusable components The page itself will then be imported to a view page consisting of only the component tag as its contents.

**Graphical user interface, diagram

Description automatically generated**

The Login area will also be a component, that will utilize the same button component as well as the navbar as seen in the component prior. This will keep a sense of consistency via the navbar and will utilize reusable components to its best ability.

**Graphical user interface

Description automatically generated**The About section will also be a straightforward component, that will again, utilize the navbar component, but also just show a screen of text and information on the goal and purpose of the web application

**Service API Design:**

<https://aa-gymetrics.stoplight.io/docs/aa-gymetrics/branches/main/xohhj6tneh0qr-aa-gymetrics>

**NFR’s (Security Design, etc.):**

The NFR designated for this project will be logging employee logins. During a login, the user information will be gathered as well as the location that the employee is designating themselves to, and the timestamp. This will be saved as an object to be saved to another table in MongoDB. After a period of 2 weeks, the database contents will be saved as a JSON file for easy readability. After the contents have been saved to a file, the database will be cleared to prevent the database from becoming crowded with login data.

**Operational Support Design:**

As mentioned above, logging will be a manual process that will utilize MongoDB to write data, taking user login information, timestamps, and location designation as the data being written. Also, since the website is currently only planned to be locally hosted, there is no need for constant monitoring for uptime. However, if the web app is to be hosted online, it be done with Heroku, and UptimeRobot would be utilized to monitor the uptime of the application.

**Other Documentation:**

N/A. Other documents have been highlighted in External Document table (see above table on pg. 5).

Appendix A – Technical Issue and Risk Log

1. Use the template to identify and monitor project issues and risks.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| R | Vue unfamiliarity causes problems | Great | Udemy Course | Arin | High | 11/19/2022 | 12/10/2022 | N/A |
| R | Barcode scanner technology is too difficult to integrate | Moderate | Online resources and testing of hardware | Arin | Medium | 11/19/2022 | 12/10/2022 | 3/10/23 |
|  |  |  |  |  |  |  |  |  |

Appendix B – References

Appendix C – External Resources

|  |  |
| --- | --- |
| **GIT URL:** | <https://github.com/arin808/aa_gymetrics> |
| **Hosting URL:** | N/A |