**PennWest California Vulcan Activity Tracker**

**CMSC-4900-001-Senior Project I**

**Fall 2025**

**Project Requirements**

**October 20th, 2025**

**Margo Bonal**

**John Gerega**

**Luke Ruffing**

**Instructor Comments/Evaluation**

**Table Of Contents**

[**Abstract** 3](#_Toc211512508)

[**Introduction** 4](#_Toc211512509)

[**Background:** 4](#_Toc211512510)

[**Constituents / Team Details & Dynamics** 5](#_Toc211512511)

[**Application Domain** 6](#_Toc211512512)

[**Project Context** 6](#_Toc211512513)

[**Initial Business Model** 7](#_Toc211512514)

[**Operational Environment** 7](#_Toc211512515)

[**Description of Data Sources** 8](#_Toc211512516)

[**Initial Requirements** 13](#_Toc211512517)

[**Functional** 13](#_Toc211512518)

[**Nonfunctional** 13](#_Toc211512519)

[**Documentation** 14](#_Toc211512520)

[**Testing / Revisions** 16](#_Toc211512521)

[**List of References** 17](#_Toc211512522)

[**Appendix I: Technical Glossary** 18](#_Toc211512523)

[**Appendix II: Team Details** 20](#_Toc211512524)

[**Appendix III: Workflow Authentication** 21](#_Toc211512525)

# **Abstract**

The Vulcan Activity Tracker is intended to serve as an athletic activity management system for PennWest California students. By entering student email credentials, a free interactive environment is accessible. The question arises, there are many current athletic activity trackers available, what makes The Vulcan Activity Tracker different? Apps on the market right now are activity specific, alienating both uncommon and unpopular exercises from using the app features. For example*, Nike Run Club App (2025)*, *Runna (2025),* and *Strava* *(2025)* are directed towards running athletes and users only. Whereas, *Strong (2025)* and *Hevy (2025)* are designed for weightlifting activities. The purpose of the Vulcan Activity Tracker is to bring together the PennWest student population into one athletic app that allows for the tracking of all campus sports, cycling, weightlifting, calisthenics, swimming, running, climbing, equestrian sports, and general exercise categories. While many other activity trackers may provide exclusive features and smart watch connectivity, they are costly and detrimental to a student’s overall budget. In one combined platform, the Vulcan Activity Tracker aspires to meet student organization needs, social interaction, and exercise goal planning in a free accessible way.

The overall projection for this document is to serve as a technical guideline for project development requirements. Software components will be discussed thoroughly to gain scope of project. All hardware required will be specified. Technical development toolsets will be determined. Team scheduling and leadership roles will be created for project organization. Overall, this document will establish a clear blueprint for senior project phases, software development, and testing to produce a deliverable in undergraduate Spring Semester 2026.

# **Introduction**

## **Background:**

A common problem for all undergraduate students is the need for organizational systems. Students balance multiple classes, jobs, friendships, family life, and workout schedules. Whether a student is a Vulcan athlete with a rigid exercise plan or has recreational and personal goals to meet, finding a place to track progress is often hindered by apps that require monthly subscriptions. The Vulcan Activity Tracker is the result of identifying both the problem of organization and access to athletic activity planning. This senior project application aims to replace exercise tracking apps, such as Strava. The Vulcan Activity Tracker aspires to fulfil the features of recording workouts, viewing performance analysis, and following leaderboards. With the social aspect of friendly competitions, this application seeks to bring the students together into an athletic community.

**Overview/Objectives of Project:**

The objective of the Vulcan Activity Tracker is to implement various features that encourage athletic participation and are helpful for recording training schedules. The Vulcan Activity Tracker is intended to be a web-based application that implements a user-friendly dashboard combined with contained database system for student credentials and activity data. The front-end or client side of this application will have 3 main components. A User Profile will allow students to add activities, statistics, and notes regarding their workout progress. Students will be able to view past activity history. The second client-side feature will be group interaction. Students will be able to make groups of other Vulcan Activity Tracker users, filtering by friends. clubs, and activities. Users will be able to record who participated with them in the activity. The third client-side feature will be a leaderboard. This leaderboard will show top users and the most popular activities, adding a fun competitive feature to the Vulcan Activity Tracker. Fun friendly competition will inspire users to perform their best at selected activity to get a spot on the application leaderboard.

On the back end of the Vulcan Activity Tracker, a server, API framework, and database will be implemented to power this web-based application. The use of internal functions and frameworks eliminates the security flaw of outsourcing code. By maintaining and controlling a database for the Vulcan Activity Tracker, data injection can be catered to PennWest student needs. An API framework will allow for fast data transfer, considering all https protocols. Organizing the server by APIs, services/data handlers, and database schema will allow for a streamlined development strategy.

## **Constituents / Team Details & Dynamics**

Teamcoordination and planning is a crucial part of the Vulcan Activity Tracker’s development. Every member of Group 2 Team is foundational and valued. Development roles will be both allocated and shared for this project to both succeed and achieve the proposed goal. Leadership roles will be assigned to each member based on skillset and knowledge in designated topic. The following tabular representation demonstrates the selection each member has chosen to lead. These phases are to show initiation of team member leadership that is enhanced by prior experiences. The individual strengths of each team member are fundamental in the creation of The Vulcan Activity Tracker. Each member is listed below:

|  |  |  |
| --- | --- | --- |
| **Team Member:** | **Major:** | **Leadership Phase:** |
| John Gerega | CS | Specification Analysis/Design |
| Margo Bonal | CS | Implementation |
| Luke Ruffing | CS | User Manual & Final Presentation |
|  |  |  |

Regarding project organization, the team will stay in constant communication through a Discord server, which is the primary source of communication. Phone numbers are saved as well in case of technical issues with Discord. Team meetings will mostly be virtual, however overlapping scheduling allows for frequent in-person meetings. For remote collaboration, GitHub is used to implement a shared team repository for project source code, documents, research, brainstorming outlines, and all other related resources. Every member is given contributor access. GitHub allows for source control, project history, and member participation mapping.

# **Application Domain**

## **Project Context**

The Vulcan Activity Tracker will be designed to help promote physical activity within the PennWest California community. This web-based application will provide access to all its features to anyone with a valid PennWest email address, whether they are students or faculty. While there are already so many popular activity tracking apps that have good features, they are too often hidden behind a hefty subscription that locks these valuable features. This creates a problem for a lot of people, especially students who are already short on money. The Vulcan Activity Tracker aims to eliminate this problem by providing a free, easily accessible, community-focused alternative.

With features such as group workout logging, campus-based organization groups, and campus-wide leaderboards, the Vulcan Activity Tracker encourages the PennWest students and faculty to stay active and engage with their campus communities. Our tracker aims to promote personal wellness and community involvement, which cannot be said for all the other popular activity trackers.

# **Initial Business Model**

## **Operational Environment**

The Vulcan Activity Tracker will view the PennWest California campus as its operational environment. Various paths around campus will be designated as athletic training areas for running and cycling paths. The Campus gym will be the physical environment for other activities, such as workout equipment, pool, running track, and climbing wall. Various athletic facilities, such as fields or indoor courts, will be used to track specific sports depending on the team or club the student is participating in. Users will represent the adjoining view of the operational environment. The proposed users of the Vulcan Activity Tracker are currently enrolled students with a PennWest email address. Demographics of these users are students who are student athletes, participate in club sports, or personal athletic training. Limiting access to only approved PennWest California campus students allows for common ground, shared interests, and an opportunity for students to connect through exercise. Additionally, it allows for fair competition on the same athletic courses.

## **Description of Data Sources**

The Vulcan Activity Tracker will handle and access multiple data sources to provide users with refined athletic information. Users will manually enter activity data into their user profile. This data will be represented in activity name, time, date, description, calories burned. Users will also be able to record what environment was used to participate in their activity. For example, a student ran for 5 miles, on the Riverview Road loop. This student would record time, distance, energy expended, and the path they chose to run on. Users can also record if it was a solo exercise or participated in a group.

The Vulcan Activity Tracker will store these records to implement in 2 ways. The first use will be an Activity History Viewer. Incorporated into a student profile, they can easily see what past activities they participated in and following statistics. The second use of the athletic records will be the overall platform Leaderboard. Students will be able to compete in challenges for overall athletic performance on the Campus Leaderboard feature. This Leaderboard will pull data such as distance to calculate the highest value between users, recording them on the leaderboard. The Vulcan Activity Tracker will use both numeric and alphanumeric data to supplement platform features. Systems of filtering, storing, and fetching data will be important to this project’s functionality.

**Use Case UML Diagrams & Descriptions**

The section below encompasses a visual representation of the Vulcan Activity Tracker. Unified Modeling Language (UML) is used to specify, visualize, and construct software architecture, designs, and frameworks. A complete blueprint of project features are mapped out and organized into manageable subdivisions. This allows for project complexity to be reduced, aiding team development. With these subdivisions, sections can also be allocated to team members for development. These UML diagrams were developed by *Lucidchart-flowchart, ERD, and UML designer (2025).*

Included in the Vulcan Activity Tracker UML are four sections: User Interface Interaction, Data Storage, Data Migration, and User to User Interaction. These sections were carefully formed to provide clarity of the main working components that drive the Vulcan Activity Tracker. The following UML incapsulates how a user is intended to interact with the platform and how input data is stored. Additionally, these diagrams show the linking of both server and client sides of the application. Finally, the UML shows how users can interact socially with other users through internet connection.

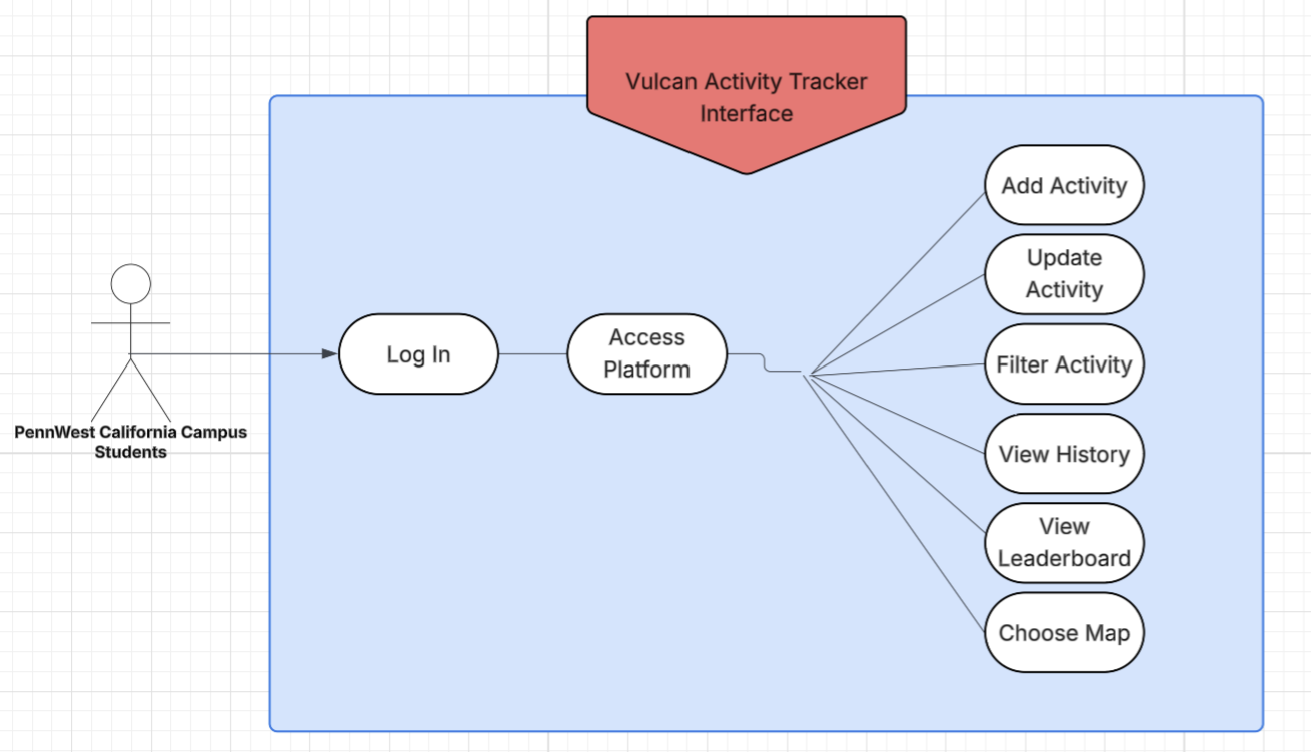


Figure 1: Overview of the Vulcan Activity Tracker application and user interaction.

Description:

The UML actor represented in Figure 1 is the PennWest California Campus student. A student user will log in by entering a validated email address and username in the appropriate data entry field. When the student user is authenticated, the Vulcan Activity Tracker Platform becomes accessible. The user will be able to navigate between pages. Each page will hold a designated feature. These features include a user profile, training areas, and a campus activity leaderboard. The user will be able to access the features of adding, updating, and filtering activities from the User Profile. The user will additionally be able to view activity history. When the student user navigates to the Training Area page, they will be able to view running and cycling maps around campus. They will be able to view gym amenities and indoor climbing areas. Finally, the leaderboard shows highest user participation and statistics in certain athletics.

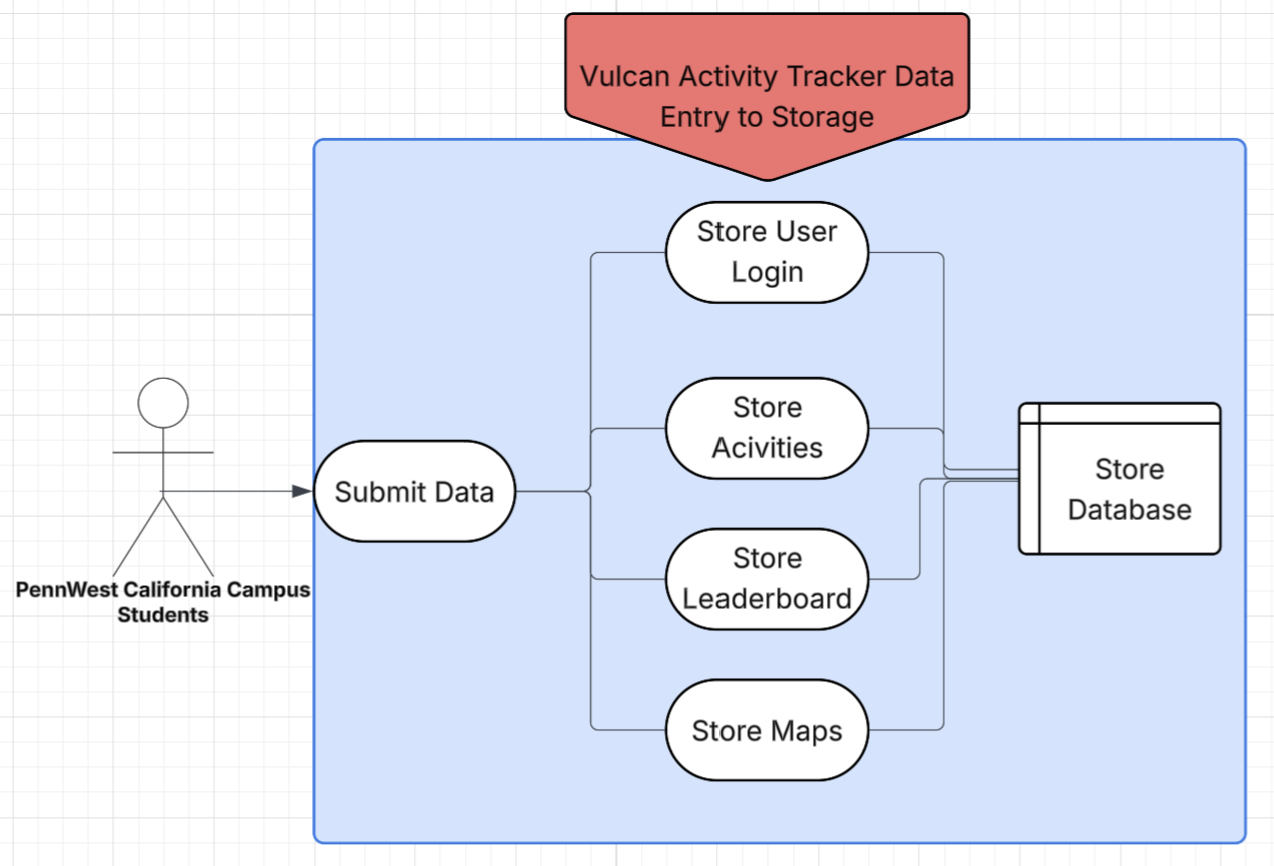


Figure 2: Overview of the Vulcan Activity Tracker user data entry and database storage.

Description:

In Figure 2, the data entry and database storage system are represented. The PennWest-California student will enter many types of data into the Vulcan Activity Tracker. This data will be activity statistics, calories burnt, miles reached, etc. This data can be numeric, alphabetic, or alphanumeric. This data will be parsed and prepared for database entry. Methods will be built on both server and client sides to pull and propagate data. See Figure 3 for further details. The data will then be stored in table format based on topic. For example, user login table, activity table, leaderboard data table. These tables will be further broken down into fields to hold specific dates, times, and other data from the student users. Database cross-references will be built in schema to link users to their specified tables of data, activities, and login credentials. PennWest student users will not have direct access to the application database as demonstrated by Figure 2 UML. The user will need to interact with the client-side interface to add, delete, or modify data in the database.

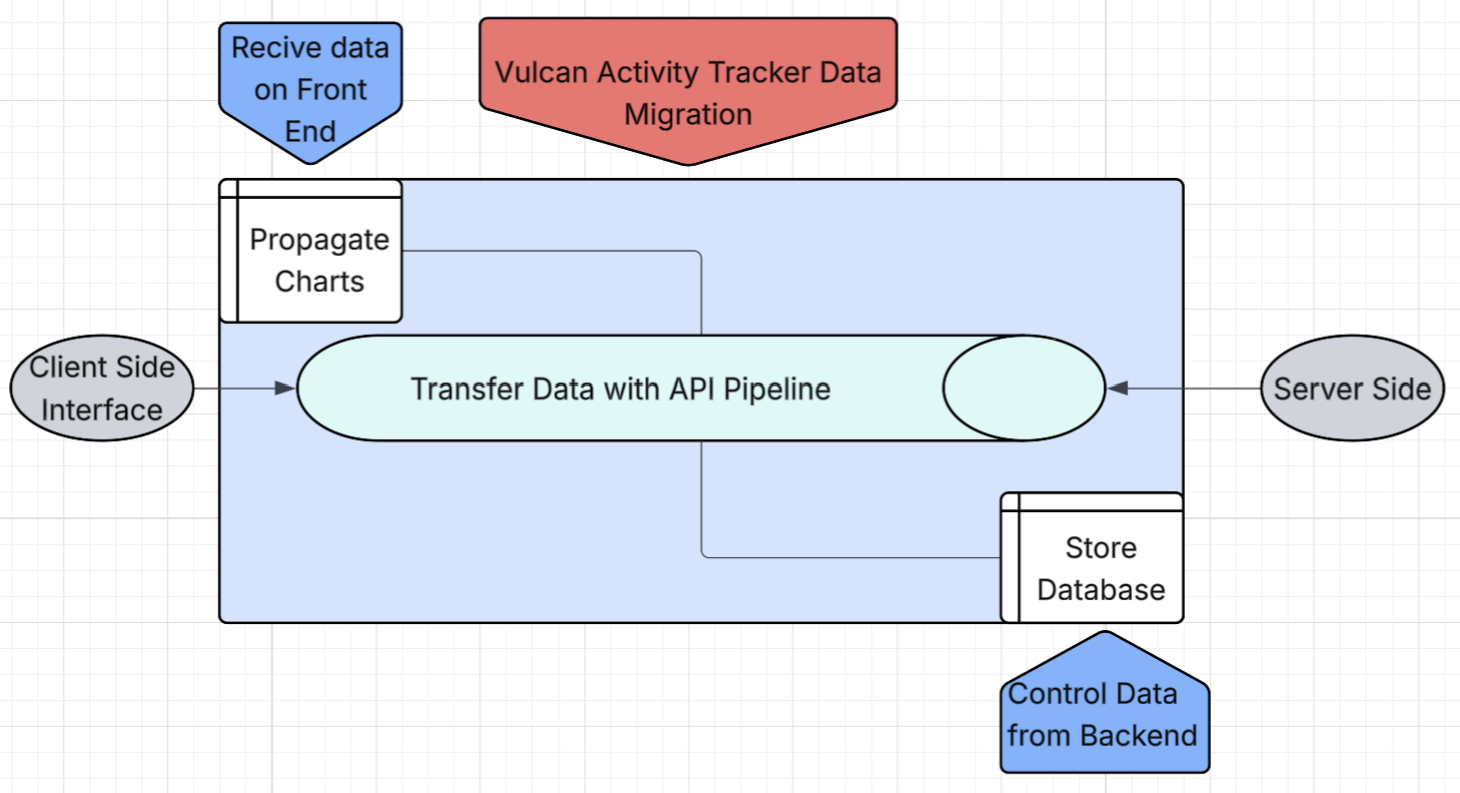


Figure 3: Overview of the Vulcan Activity Tracker data migration from client to server.

Description:

Data migration is a complex aspect of the Vulcan Activity Tracker. Figure 3 is the UML graphical representation of this software architecture. Data entered by users needs to be prepared then transported to the back-end engine of the application. This data then needs to be stored in an internal database that is specifically designed for student data. The front-end interfacing will be designed to implement methods that receive data and fill the appropriate charts and tables with existing information. Likewise, on the backend data is received from front-end forms and controlled by data handlers. Vulcan Activity Tracker data is then inserted into the database with proper logic and schema. The link between the client and server sides of the application will be the API pipeline. This logic framework sends data across the network using HTTP protocols.

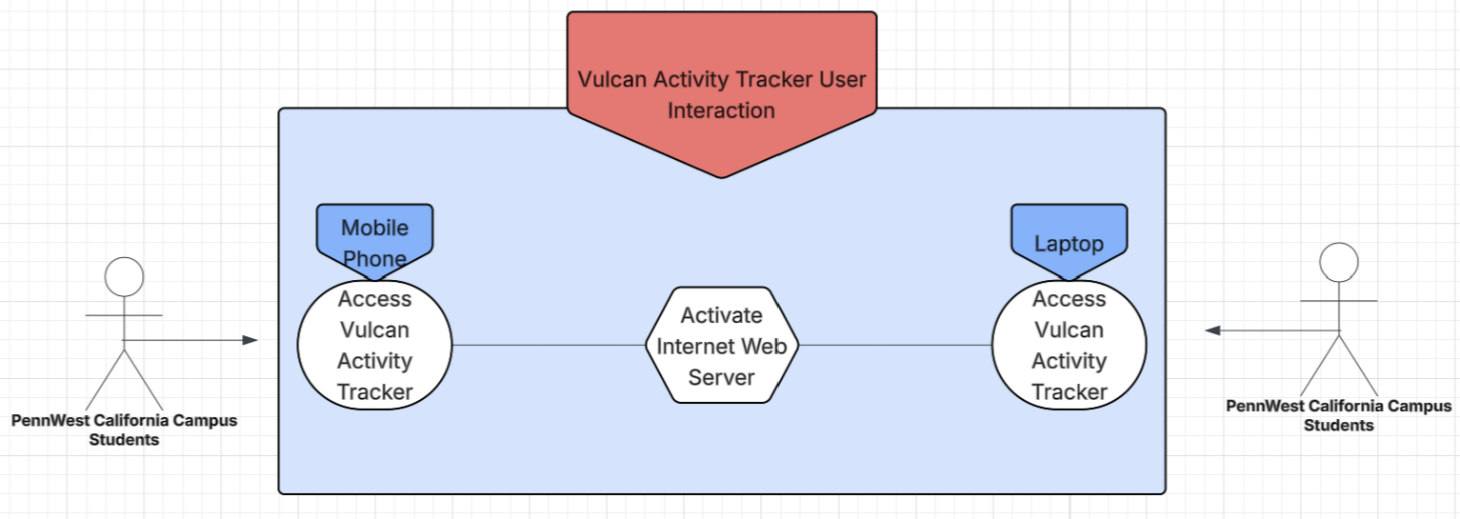


Figure 4: Overview of the Vulcan Activity Tracker user social interaction.

Description:

Figure 4 UML shows the interaction between student users of the Vulcan Activity Tracker. The application can be accessed by either mobile device or computer as demonstrated in Figure 4. The application is accessible if connected to an internet network. The PennWest Wi-Fi network system is sufficient for testing throughout development and demonstration purposes. Students will be able to interact with each other through two features, group activities and campus leaderboard. PennWest-California campus students will be able to do group exercises and activities with friends or clubs. Users will be able to view activities statistics for their group and see who participated in the activity. The Leaderboard will be available to all Vulcan Activity Tracker users, allowing fun competition between users on workouts, sports, and activities.

# **Initial Requirements**

## **Functional**

This project has several functional requirements, the first being a successful database. The database will ensure that a user’s activities can be tracked and that total values can be calculated, which can then lead to the accurate data being displayed for all of those users’ friends on the app to view. The database will also securely hold a user’s login information, like their email and password.

The first thing a user will do before accessing anything is log in to the system, whether they are a new or returning user. Once the user enters credentials, the program will search the database to either ensure the credentials are correct or create a new account with the provided credentials. Once the credentials are verified or the new account is created, the user will be entered into the activity feed section, which will display other users' activities.

## **Nonfunctional**

This project also has several nonfunctional requirements that help to ensure that it is reliable, secure, and easy to use. To provide a smooth user experience the system should respond within a couple seconds under typical conditions and support at least one hundred users at once without lag. Users should be able to upload their workouts and view other activities without delay or performance issues.

Security is an especially important part of the system. All users’ data will be encrypted while being transferred and securely in the database while at rest. It will also have built-in safeguards against data injections and unauthorized access to protect every user and their data from data breaches. Additionally, the system will be developed with maintainability in mind and will be expected to maintain around 99% uptime, excluding downtime for maintenance.

## **Documentation**

The following outline demonstrates the needed documentation as outlined by CMSC-4900-001 Professor, Dr. Chen, for all Senior Project I and II development phases. Defined below are the several forms of documentation needed to bring the Vulcan Activity Tracker from concept to actuality:

1. Proposal Document
   * The Proposal is the initial document phase, it is used for project mapping, planning, and definition after the Project idea is approved.
2. Requirements Document

* The Requirements Document is the secondary document used to represent overall goals of the project, development strategy, and visual UML to aid in structuring of main components needed to create application.

1. Specifications Document

* The Specifications Document is the third document phase that formulates actual requirements, standards, testing, revisions, and parameters of final deliverable project.

1. Design Document
   * The Design Document is the fourth planning document phase that describes project layout, appearance, features, functions, and allows integration of cohesive themes.
2. Project Log
   * The Project Log documents the Implementation phase. It is intended to track development, software and planning faults, and debugging solutions. Project Log also follows timeline and scheduling to ensure deliverability is completed, and all goals are met.
3. User Manual
   * The User Manual is the final form of documentation, used to provide insight on project functionality and defined method of use. It will provide education to proposed users in the format of a step-by-step guide.

# **Testing / Revisions**

To ensure the highest quality deliverable, this document was carefully revised. Team discussions were held to unite project scope, actions, and development path. Each team member added both valuable insight and documentation to this Requirements Document. Multiple drafts were formed, discussed, and revised to produce this final document. Digital communication was key in this process since it was developed primarily remote, however, in-person meetings were held to discuss specifics and hold progress updates.

By implementing Git/GitHub, a similar procedure has been planned for the software development phase of the Vulcan Activity Tracker. A ticketing party will be held to assign each member to tasks and frameworks. Branch policies will be implemented. Main branch will hold a working deliverable project at all times. This means that the code on the Main branch must be executable, resulting in documented project history as well. For all new features and updates, each member must make a branch off the Main branch for development. These branches must be formatted in *memberName/feature* for progress tracking. When a feature is completed, a Pull Request (PR) must be initiated to merge the team members branch into the Main branch. This must only be approved when code is thoroughly tested, executable, and bugs logged. The team member who initiated the PR is not sanctioned to approve their own code into Main Branch. Another member must review and approve their PR request. This will lead to team democracy and an egoless programming approach. Likewise, digital communication and regular in-person meetings will be held to map out and discuss project scope and design. Any development disagreement will be held to an overall vote and gaining advice from professor.

# **List of References**

Hevy Studios S.L. (2025). ***Hevy – Gym Log Workout Tracker*** (Version 2.4.7) [Mobile app].

<https://www.hevyapp.com/>

Lucidchart Software Inc. (2025). ***Lucidchart*** [*Website and Software Application]*

<https://www.lucidchart.com>

Nike, Inc. (2025). ***Nike Run Club: Running Coach*** (Version 7.70.0 for iOS) [Mobile app].

<https://www.nike.com/nrc-app>

The Run Buddy Ltd. (2022). ***Runna: Running Training Plans*** (Version 8.5.0) [Mobile app].

<https://www.runna.com/?gad_source=1&gad_campaignid=22530268929&gbraid=0AAAAABfUKAUcs2aA7jayQhjZYrJnvDftf&gclid=CjwKCAjwuePGBhBZEiwAIGCVS1dtaMKRlGDzWrJsTvy5TKab0O_LodHzXuZie-qwF0uvQrYvfe_-CRoC89MQAvD_BwE>

STRONG FITNESS PTE. LTD. (2025**). *Strong Workout Tracker Gym Log*** (Version 6.2.1) [Mobile app].

<https://www.strong.app/>

Strava, Inc. (2025, Sept 11). ***Strava* (Version 428.0.1)** *[Website and Mobile App].*

<https://www.strava.com>

# **Appendix I: Technical Glossary**

**API –** Short for Application Programming Interface, a connection between computers or computer programs that offer services to other pieces of software. (site for citation - [API - Wikipedia](https://en.wikipedia.org/wiki/API))

**Back-End –** In software development, back-end refers to the data management and processing behind the scenes. Back-end developers focus on the server side of websites. They use technical skills to perform the behind-the-scenes work that creates a website’s structure and overall functionality, allowing a site’s front-end to exist.

**Branch –** instruction in a computer program that can cause a computer to begin executing a different instruction sequence and thus deviate from its default behavior of executing instruction in order. May also refer to act of switching execution to a different instruction sequence as a result of executing a branch instruction.

**Database** – a structured set of data held in a computer, especially one that is accessible in various ways

**Deliverable –** means result or software product, designed document, or asset of project plan that can be submitted to customers, clients, or end-users.

**Front-End –** In software development, front-end refers to the presentation layer that users interact with. Front-end developers ensure that visitors can easily interact with and navigate sites by using programming languages, design skills, and other tools

**Git –** A distributed version control software system that is capable of managing versions of source code or data. It is often used to control source code by programmers who are developing software collaboratively. (site for citation - [Git - Wikipedia](https://en.wikipedia.org/wiki/Git))

**GitHub –** A proprietary developer platform that allows developers to create, store, manage, and share their code. It uses Git to provide distributed version control and GitHub itself provides access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project. (site - [GitHub - Wikipedia](https://en.wikipedia.org/wiki/GitHub))

**Maintainability –** determines how easy and profitable it will be to maintain, update, and do upgrades in that software system.

**Pull Request (PR) –** A pull request is a request to merge changes from one branch into another branch in a Git repository. Typically, pull requests are used in collaborative workflows where multiple developers work on different features or fixes in separate branches. (site for citation - [Git Pull Request - GeeksforGeeks](https://www.geeksforgeeks.org/git/git-pull-request/#what-is-a-git-pull-request))

**Repository –** specialized storage systems designed to manage source code and other software development assets. They are active, intelligent systems that track every change, coordinate collaboration between developers, and maintain the entire history of your project.

**Server –** A computer that provides information to other computers called “clients” on a computer network. Servers can provide various functionalities, often called “services,” such as sharing data or resources among multiple clients of performing computations for a client.

**Ticket –** A ticket is a special document or record that represents an incident, alert, request, or event that requires action

**Ticketing System (Ticketing Party) –** a tool that organizes requests and automates service requests.

**UML –** Unified Model Language (UML) is a general-purpose modeling language with the main aim to define a standard way to visualize the way a system has been designed. (site - [Unified Modeling Language (UML) Diagrams - GeeksforGeeks](https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-introduction/))

# **Appendix II: Team Details**

This requirement document was developed through the collaboration of all team members of group 2, under the leadership of Margo Bonal. The document was made with the combined efforts of the entire team over the course of the project timeline. While each member contributed across various parts of the document, the primary contributors for each section are listed below:

* Margo Bonal:
  + - Abstract, Background, Overview, UML diagrams/Descriptions, Team Dynamic, Operational Environment, Data Sources, Testing/Revisions, Documentation, References
* John Gerega:
  + - Abstract, Project Context, Team Dynamic, Glossary
* Luke Ruffing:
  + - Functional, Nonfunctional, Documentation, Title Page

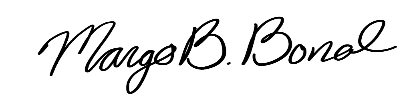
The rest of the major sections of the document were completed collaboratively with all members participating equally in discussions, decision making, and writing. This collaborative approach taken to completing the document allowed the group to refine ideas and reach a consensus before making any big decisions.

In addition to the writing of each section, all team members took part in proof reading and formatting ensuring the creation of a clear, concise and well organized document. Together the team worked to check for proper grammar, proper citation, and adherence to all formatting guidelines.

# **Appendix III: Workflow Authentication**

I, Margo Bonal, confirm that all work I have contributed to this document has been to the best of my knowledge, completed with integrity and in accordance with the objectives and collaborative standards established by the team. I attest that I will maintain these standards in all future contributions to this project.



Signature:  Date: 10/20/25

I, John Gerega confirm that all work I have contributed to this document has been to the best of my knowledge completed with integrity and in accordance with the objectives and collaborative standards established by the team. I attest that I will maintain these standards in all future contributions to this project.

Signature: A black text on a white background

AI-generated content may be incorrect. Date: 10/20/25

I, Luke Ruffing confirm that all work I have contributed to this document has been to the best of my knowledge completed with integrity and in accordance with the objectives and collaborative standards established by the team. I attest that I will maintain these standards in all future contributions to this project.

Signature:A close-up of a signature

AI-generated content may be incorrect. Date: 10/20/25

**Appendix IV: Report from Writing Center**

Writing Center Instructor: Chayse Lizon

Appointment: 10/13/2025 2:00 PM EDT

*The following notes related to your 10/13/2025 2:00 PM EDT appointment with Chayse Lizon have been shared with you:*

You and your group came in for a review of your paper overall and in comparison, to the rubric and the example papers provided. Content wise I think you did well in the explanation of the app and the process of how you all worked together came to conclusions, in addition to just general communication. I think the biggest point that I noticed for you guys to work on is the formatting of the paper, consistency is important. Make sure that everything is times new roman, 12 point, and double spaced.

You asked about if your names should be on the paper and I would say yes, you might want to include your group number too (I believe I read that you are group two); if you are unsure if you should have names or group number on your title page because it is not reflected on the example page or the instructions, I would suggest talking to your teacher.

I had deleted a few spaces that were unnecessary once the size of the font was changed to 12 point, but I would suggest that if you have a space before one of the sub-titles then you proceed to do that for all of them since sometimes you do it and other times you do not -- this is not the case for the big titles, because your group tends to keep a space between them. Make sure your indentations are the same size, for the abstract they are about 2 indent spaces while everywhere else throughout the paper is just one indent. There were no major misspellings that I have noticed, just check the underlined red portions to make sure everything is grammatically correct (there are often instances where "the" or "a" should be preceding a word, so make sure those are in agreement).

I noticed in some areas you have "main branch" written as "Main Branch, Main branch, main branch," try sticking to one way of writing it. As it gets later in the paper there are more cases where commas need to be included, this will come up on the underlined red words, but you can also try reading aloud the paragraphs and putting a comma where one would naturally take a breath in a sentence (I think some are needed in the area where you signed your names for example).

Take out the "and" in "And I attest...," and make sure that if you have a punctuation following a quote, that that punctuation is within the quotation marks. Finally, you and your group discussed if you needed more citations and decided you were okay, but you should go over the citations in the glossary whether you will keep them or not; in the case that you keep the in-text citation, make sure there is a matching citation on the list of references. Also, on the references page, check if those citations should be bolded, because I don't think they need to be -- also please make sure your citations are in APA format and are consistent with one another (such as .5 inch indent for the hanging indent). If you have any following questions, please feel free to email me at [liz58193@pennwest.edu](mailto:liz58193@pennwest.edu)