Milestone 2

Olympian Tracker

SW Engineering CSC 648/848 Spring 2024

Section 2

03/12/2024

Team 3

Lei Woods (Team Lead)

Eric Kunzel

Chunyin (Alex) Chan (Front End Lead)

Elliot Bullard (Github Master)

Mila Avagimova (Back End Lead)

Oscar Galvez

Riel Orque

History Table

Submission Date:	**Revision Date:**	**Revisions Made:**
03/12/2024		

Table of Contents

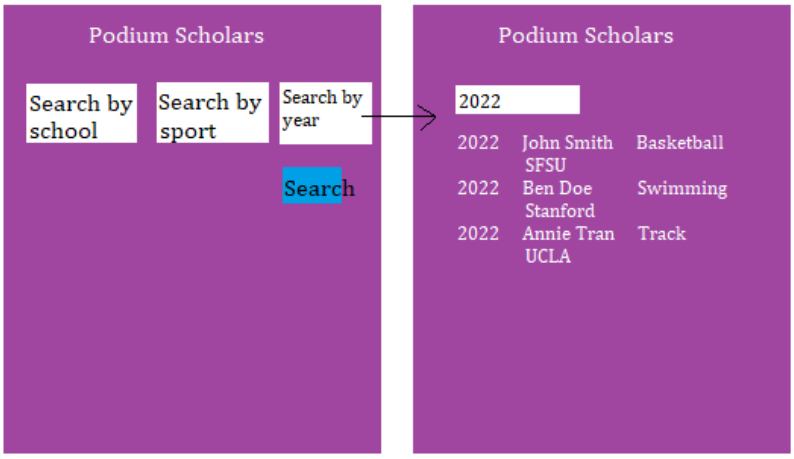
Section I: Functional Requirements

- Website Requirements:
    - 1) Shall provide login and logout functionality.
    - 2) Shall utilize proper account security (password hashing, etc.).
    - 3) Shall verify proper email has been used to register by utilizing email
    - 4) Shall display events in maps and states and provide basic mapping interfaces.
    - confirmation.
    - 5) Shall be media rich.
- General Users:
    - 6) Access shall be limited to only registration, log in and a very basic homepage with event news.
    - 7) General unregistered users shall be able to access the registration page and register as a new user.
    - 8) General users shall be able to access the login page and log in to access the rest of the website.
- Registered users:
    - 9) Shall be able to register as either a university entity, a student/General user (parent), or an instructor from a university.
    - 10) Shall be able to use map functionality to locate each university.
    - 11) Shall be able to recover lost usernames and reset lost passwords.
    - Students:
        - 12) Shall be able to send direct messages to other users/university hub accounts/teachers.
        - 13) Shall be able to search/filter for universities by sport, Olympic admission rate, win percentage, specific Olympic athlete.
    - Teachers:
        - 14) Shall be able to link their account to the university they work for.
        - 15) Shall be able to send and respond to direct messages from students.
    - Universities:
        - 16) Shall be able to create a list of student athletes and enter their information.
        - 17) Shall be able to update/add/change/delete information about athletes related to the university.
        - 18) Shall have “admin” accounts that manage university information.
-

Section II: UI Mockups and Storyboards

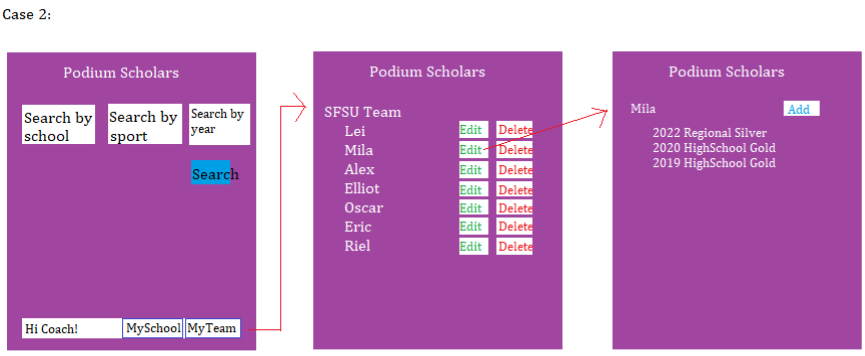
Storyboards:

Use Case 1:



Lei, an aspiring Olympic swimmer, wants to find universities with strong swimming programs that have produced Olympic athletes. She uses the application to filter universities by sport, state, and Olympic participation history by year. The application displays a list of universities with detailed profiles, including athlete achievements, training facilities, coaching staff, medals rewarded, etc. Lei would be able to examine each university by sports and state and is able to decide on which university she would like to attend.

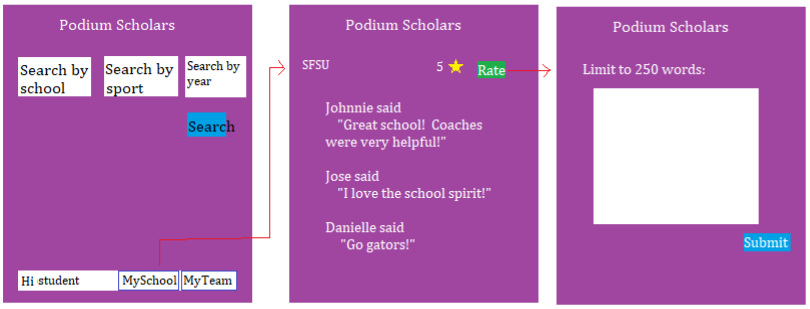
Case 2:



Coach Mila, a successful coach and athletic director, logs into the application to update his athletes' profiles with their latest achievements. She adds new medal wins, track meets, athletes

personal records, training regime, to the website for future athletes to see the success of her training, coaching, and program. Coach Mila is also to make comments and reviews on each of her athletes regarding work ethic and performance.

Case 3:



After participating and attending a successful university athletic program that has produced Olympic athletes, Alex has decided to leave his thoughts and evaluations on his current university athletic program. Alex can write reviews and rate each of his coaches and athletic programs. He can give insight and feedback that will be useful to future athletes who are able to view Alex's reviews and ratings.

Section III: High Level Architecture, Database Organization

Documentation for database

The schema for our database is divided into 8 tables. These tables will contain the data points that will be saved and persist on the server. At the moment the database contains simple text only, but it can be expanded to store images and other media as needed.

Tables

Users – will hold information about user accounts
- user_id
- user_type
- user_email
- password
- first_name
- last_name

Universities – will hold information about individual universities
- University_id
- Name
- Location
- Sport
- Medals
- notable_athletes

Athletes – information about individual athletes, foreign key links to universities
- Athlete_id
- University_id
- Athlete_stats
- Sport_played
- medals_won

Sport – will contain a foreign key id linked to a university and the sport name
- Sport_id
- Name

Rating – user entered ratings/reviews will be stored here
- Rating_id
- University_id
- User_id
- Rating
- review

Students – table for storing student account information
- student _email
- First_name
- Last_name
- School_attended
- Athletic_statistics
- academic _stats

University Staff - table for storing staff account information
- Staff_id
- First_name
- Last_name
- Password
- University_id

Guest Users

Media storage type

The media content for this project will be stored using BLOBs. There is no need for any special data format, as MYSQL can store all required data types using BLOBs. This project will mainly feature image and video files. (type to be determined) Implementing BLOB storage these files will be easily stored, recalled and manipulated.

Keyword search

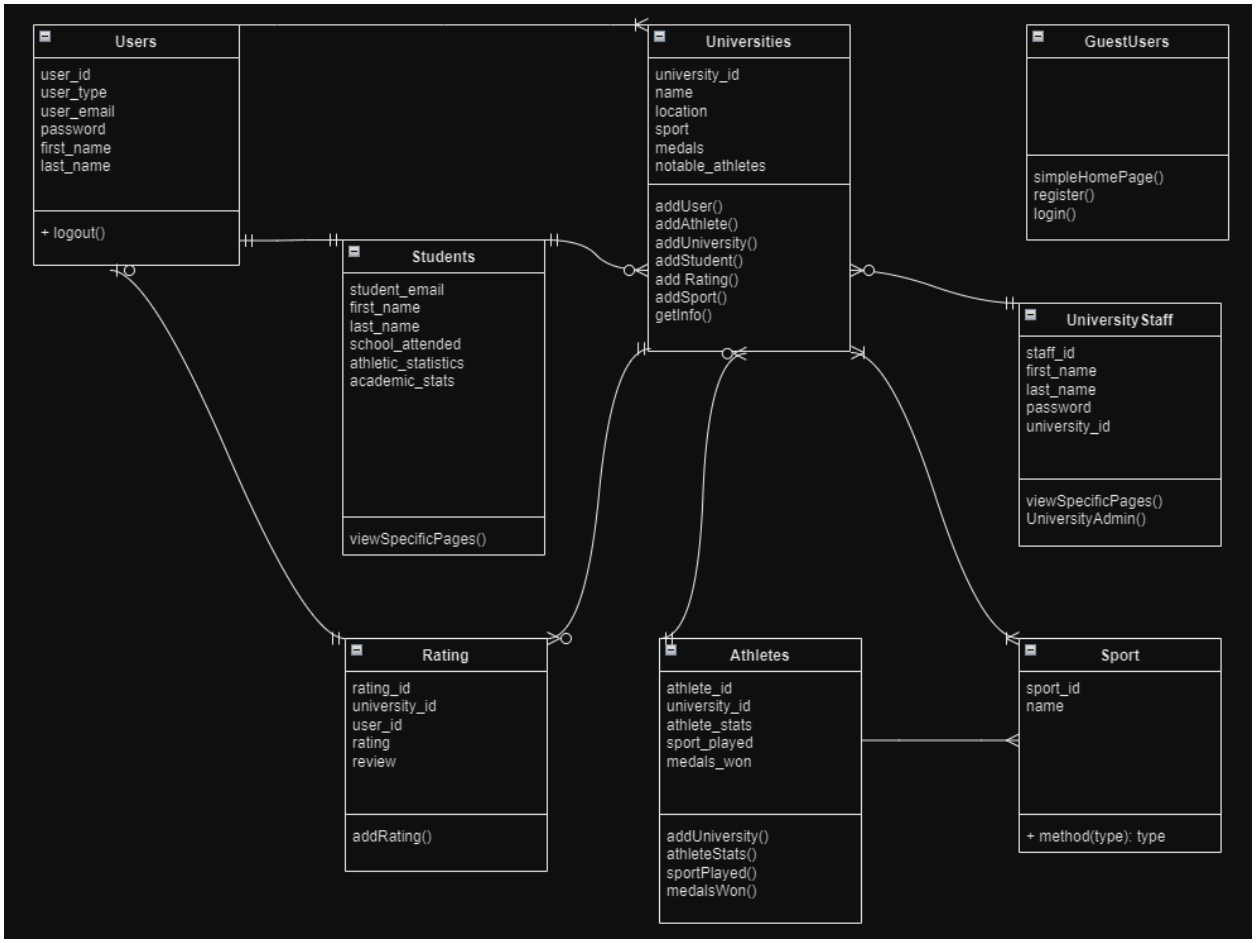
The search function intakes a user keyword limited to one word with a filter dropdown option. The search algorithm uses Javascript and the pg-promise Node library to access the database. The values inside the Universities table will be checked against the user supplied keyword. If a match is found the university will be added into the html of the results page before it is loaded, and upon completion of the search, the results page will be rendered. The steps are outlined below

1. User enters keyword, presses search button
2. Search button triggers search function using keyword
3. Every entry in Universities table is checked, every row and column
   This will be done by making queries to the database using pg-promise and comparing against the keyword
4. Every time the search function finds a college whose database entries contain the keyword, it will create a unique html tag inside the results html doc that displays information about the college, one after the other
5. After all entries in db have been checked, the results page will be rendered with the results
    - We can organize the results in several ways
      *alphabetically
      *by order found in
      *by relevance(how many hits of keyword in database)
    - User can specify which way they would like results to be shown
    - For now the results will be sorted alphabetically, with the ability to

Filter Search

A filter drop-down is shown next to the search bar. This will allow the user to filter searches based on a number of fields which can be expanded over time. At the moment the filter will allow the user to filter based on specific colleges.

Section IV: UML Diagram



Section V: Identify Actual Key Risks

Skills Risks

- Member familiarity with Server-side language
    - Most of the team has a basic understanding of Javascript, but we don't have the comfort level required to immediately set off on a big project requiring it. Out of the six members, three have a self-evaluated comfort level of 3/5. Addressing the issue requires each member to take time to self-study the intricacies of Javascript. Putting in this extra time will allot the entire time to fully understand the nuances of the language, allowing us to complete the product successfully.
- Lack of experience with application deployment
    - The group has not had the opportunity to deploy a web application during our studies. Like with deploying the server, creating this web application together is a new experience for all of us. Mitigating potential problems means exploring these topics together. Multiple perspectives allow us to efficiently address issues, allowing more time to focus on product development.

Schedule Risks

- Meeting with a group of 6 increases the difficulty with scheduling collaborative sessions due to time constraints per person. Different availability made it rather difficult to pin down a time that really worked for every group member. To resolve this, it requires personal sacrifices in order to have a session where everyone is in attendance. For example, all-member meetings now occur on Thursdays at 530PM, despite requiring some members to rush from prior commitments. These compromises allow the group to cut meetings down to one session outside of class meetings.

Technical Risks

- Server Maintenance
    - To every member of the group, deploying a server for a web application was a first. It is logical to assume that any sort of issue that might come up along the development process will pose its own challenge as we don't have experience with it. We will need to research the individual issues as they develop. Team members can research together in order to solve the issue quicker, as we all have access to the server credentials as needed.

Legal/Content Risks

- We need to be cautious about possible licensing problems that we might run into using images other than the ones provided. This could include permissions for school photos, information that complied, and potential expenses such as some that Google Maps API can incur. Mitigating these risks requires us to identify the essential content for our product, and to seek alternatives if the need arises. Most of the information we can compile online appears to be freely available, but we need to exercise caution while verifying this.

Section VI: Vertical Prototype