SW Engineering CSC648/848 Spring 2024

Olympic Quest College Search

Section 02 Team 02

Team Lead - Shriya Dandin (sdandin@sfsu.edu)
Front End Lead / System Administrator - Victoria Bialorucki (vbialorucki@mail.sfsu.edu)
Back End Lead - Sam Zandiasadabadi (szandiasadabadi@sfsu.edu)
Front End Team - Erika Chan (echan10@mail.sfsu.edu)
Github Master - Jake Klopukh (jklopukh@sfsu.edu)
Back End Team - Omar Ahmed (oahmed1@sfsu.edu)
Database Engineer - Subodh Khadka (skhadka@sfsu.edu)

Milestone 1
2/27/2024

Date Submitted	Date Revised
2/27/2024	

Milestone 1

1. Executive Summary

Our application, Olympic Quest College Search, aims to simplify searching for colleges, specifically those that specialize in various Olympic sports. We hope to streamline this process, making it simpler for students to be informed about choosing a college. We believe our product will give students an opportunity to consider all of their options; we hope to support and encourage both future and current students toward their goal of being an Olympian.

Various functions and services offered from our application, which aims to enhance user experience. A list of colleges will be generated based on what relates to the student's search, providing information on each college's athletic programs. In addition, our application will include a rating system which would allow students to see the rating of the school's athletic program(s).

Another function/service that is incorporated in Olympic Quest College Search is a messaging system where students and staff can message each other. Our application is designed to be user friendly, ensuring that students can easily navigate throughout the website.

With the goal of hoping to make it an easier college search process for students in mind, our team made sure we included multiple functions/services that would be helpful to students using our application. As our team includes a few members that have been previous student athletes, we feel as we can relate in some ways to students who wish to search for colleges that specialize in multiple different Olympic sports.

As the purpose of Olympic Quest College Search is to provide help with making informative academic and athletic choices, our team can see how this application can go far. Especially an application like this would have been something our team wished we could have had, seeing how useful it is and how it could have helped us make academic decisions.

2. Personas and main Use Cases

Personas:

1. Claire Williams is a 17-year-old high school senior who lives with her parents. She is a sporty person who played sports all throughout her time at school. Currently she is participating in track and field as a long-distance runner. Since she has always been so involved in sports as well as trying to do well in school, she does not have much free time. Like other people her age she knows how to browse the internet and is often on social media websites, but outside of that is not well versed in technology. As a child, she watched the Olympics which drove part of her lifelong huge interest in sports. This childhood event created her persistent dream that she would one day be involved in the Olympics. She is a senior about to apply to college, like many others in her situation she is overwhelmed. Since participating in the Olympics is so important to her, she wants to go to the university which would most help her with that goal. However, there are so many options that she is unsure which school is the best option for pursuing that goal and does not know a good way to find out.

2. Robert Richardson is a 38-year-old man who works at San Francisco State University as a manager on their athletic team. He owns a mortgaged home with his wife and has two children. When he was younger, when he went to university he was part of the swim team while pursuing his degree in business. Between his work and his home life with his kids, he is quite busy, but in his free time he enjoys watching sports. He has basic internet browsing knowledge, but his overall technology knowledge is not high. His biggest pride at work is the amount of medal-winning Olympic athletes that have been in his program. However, even though he has sent a lot of athletes to the Olympics, most students who join the athletic program are unaware of the school's Olympic accomplishments. He wants to make more people aware of the school's accomplishments and use that to recruit more students interested and passionate in participating in the Olympics so he can improve the school's record even more.

3. Will Williard is a 26-year-old college graduate who works as a site admin. He lives alone in his apartment that he began renting only six months ago. In his free time, he enjoys playing video games online as well as taking walks on the beach. His goal is to one day create his own successful website and run it himself. As a web admin, he has a deep understanding of the internet and how websites work from behind the scenes. When he is doing his job as a web admin for a website, his life is made easier when the website is convenient to access, allowing him easy modification of the database and deletion of malicious users.

Use Cases:

1. School searching: Claire wants to search for a college with many Olympic long-distance runner medal winners. She goes to our site and searches by sport and can review a list of searches. The search list can then be filtered and sorted appropriate to Claire's needs. When she finds a desired result, she can expand it and use that to learn more information about the school.

2. Student-School Contacting: Claire wants to get in contact with a school. From a chosen school's page, she can try to contact someone at the school. Before she can message them, she is first prompted to login to access the messaging system. After login, she is given access to a message feed with someone at that school and can talk with them.

3. School Staffer Page Maintenance: Robert wants to change info about his school on the website. To change school information, he must log in as a school staffer of the appropriate school. He is brought to a page that allows him to modify his school's entry. He enters the new information about the Olympic participants, the sports they participated in, and the medals they won, and then the page is updated.

4. School-Student Contacting: Robert wants to get in contact with interested students and recruit them. He can message students that have already reached out through an inbox. He can also look through students that have shown interest in the school and proactively

send them messages.

5. Website Management: Will wants to manage and moderate the website. He can log as an admin, which gives him access to modifying any page's info. He can delete users and schools as appropriate. He can change permissions and monitor activity on the websites such that the website is smoothly running.

3. List of Main Data Items and Entities

Main Data Items:

1. Ratings - A system to evaluate the quality and success of athletic programs in specific sports.

2. Messaging system- Communication between students and staff for inquiries, discussions, etc.

3. Rankings- Rankings of colleges and universities based on the success and quality of their athletic programs in different sports.

Entities:

1. Colleges/universities- colleges and universities offering athletic programs.

2. Athletic Programs- Details regarding athletic programs offered by colleges/universities.

3. Students/Staffs - Individuals using the platform.

4. User profiles- Profiles of individual users, containing personal information, Academic achievements, athletic accomplishments, etc.

5. Academic Programs: Information about academic programs offered by universities.

6. Sports- Details information about various sports.

7. Personnel- Personnel associated with colleges/universities, including coaches, academic advisors, etc.

4. Initial List of Functional Requirements

1. Allow athletes to create accounts - giving each user-athlete the ability to create an account using their name and information to be seen.

2. Allow athletes to log-in to their accounts - giving each user-athlete the ability to log-in to their account given their account registration information.

3. Allow athletes to modify their accounts whenever - giving each user-athlete the ability to add, remove, or update their personal information whenever they need to.

4. Allow recruiters to create accounts - giving each user-recruiter the ability to create an account using their name and information to be seen.

5. Allow recruiters to log-in to their accounts - giving each user-recruiter the ability to log-in to their account given their account registration information.

6. Allow recruiters to modify their accounts whenever - giving each user-recruiter the ability to add, remove, or update their personal information, or guidelines for accepting/scouting athletes whenever they need to.

7. Allow admins to be assigned a unique master account - assigning an admin for the website who would be in charge of maintaining the website.

8. Allow admins to modify all users' posts - giving the admin the ability to modify both the athletes, and the recruiters profiles.

9. Allow athletes to apply/reach out to colleges/recruiters - allowing the user-athletes to reach out and connect to college recruiters to learn more about their program.

10. Allow athletes to apply/reach out to multiple colleges/recruiters simultaneously - similar to the previous requirement, however, ensuring that the user-athlete can contact multiple college recruiters at the same time.

11. Allow recruiters to reach out to athletes - allowing the user-recruiters to reach out and connect to athletes in order to introduce them to their program and talk to them about their needs.

12. Allow recruiters to reach out to multiple athletes simultaneously - similar to the previous requirement, however, ensuring that the user-recruiter can contact multiple athletes at the same time.

13. Allow athletes to search for specific colleges - allowing the user-athletes to search for a specific college to learn about their program or to connect with their recruiter.

14. Allow recruiters to search for specific players - allowing the user-recruiters to search for a specific athlete in order to learn more about them and connect with them if necessary.

15. Allow admins to search for specific players or recruiters - allowing the website admins to search for both user-athletes and user-recruiters in case they need to identify, modify or remove any accounts.

16. Allow website users to sort data based on name - allow all users to sort the data within the website based on the name of the college to easier identify the college they are interested in.

17. Allow website users to sort data based on state - allow all users to sort the data within the website based on the state a college is located in order to showcase all available colleges in that specific state.

18. Allow website users to sort data based on year - allow all users to sort the data within the website based on the year they would be qualified to attend a specific college.

19. Allow website users to sort data based on sport - allow all users to sort the data within the website based on the sport they are interested in playing/recruiting.

20. Generate a list of colleges that matches an athlete's preference/checklist - generating a list of colleges that best matches what a user-athlete is looking for.

21. Generate a list of athletes that matches a recruiter's preference/checklist - generating a list of athletes that best matches what a user-recruiter is looking for.

22. Mark certain words as offensive and unacceptable - marking offensive, discriminatory, and unacceptable words.

23. Ensure no user can type/use such language - deleting or censoring words that have been marked as offensive or unacceptable.

5. <u>List of Non-Functional Requirements</u>

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).

2. Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers.

3. Selected application functions must render well on mobile devices (this is a plus).

4. Data shall be stored in the team's chosen database technology on the team's deployment server.

5. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.

6. The language used shall be English.

7. Application shall be very easy to use and intuitive.

8. Google maps and analytics shall be added.

9. No email clients shall be allowed. You shall use webmail.

10. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

11. Site security: basic best practices shall be applied (as covered in the class).

12. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.

13. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application).

6. Competitive Analysis

Feature	Big Future	NCAA	Ivywise	Educated Quest	Our Future Product
College search	++	+	-	-	+
College profiles	+	+	+	-	+
Student resources	+	+	+	+	+
Messaging between users	-	-	-	-	+
Rating system for colleges	-	-	-	-	++
Olympic pursuits for students	-	-	-	-	+

+ feature exists; ++ superior; - does not exist

1. https://bigfuture.collegeboard.org/
2. https://www.ncaa.org/index.aspx
3. https://www.ivywise.com/
4. https://educatedquest.com/

Planned Advantages of Our Product

The product we aim to develop addresses the challenge of identifying colleges tailored to athletic students, specifically those aspiring to become Olympians in their respective sports. One significant advantage of our planned product is the inclusion of a rating system, allowing students to compare colleges based on their athletic programs' quality and success in specific sports. Unlike existing competitors such as Big Future, NCAA, Ivywise, and Educated Quest, our product will offer specialized focus on athletic programs and ratings.

Additionally, while our competitors lack a messaging system for student interaction, our product will incorporate this feature, allowing communication between students as well as staff. We would like to enable students to make informed decisions regarding their athletic aspirations and academic pursuits. We hope this ultimately allows them to pursue their dreams of becoming Olympians in the future.

7. High-level system architecture and technologies used

- **HTML, CSS, JavaScript:** Client-end logic, styling, etc.
- **Fetch API:** Making HTTP requests to the back-end server
- **Framework:** Node.js with Express.js for the server-side application
  Database: MySQL for storing user data (future students, current students, and staff), profiles, colleges, and olympic records
- **Server:** Using AWS EC2 for Ubuntu Pro instance running version 22.04 LTS
- **Webmail:** Internal website email
- **Maps and Analytics:** Potentially using Google Maps and Google analytics for giving user results
- **Git:** Git with Github
- **IDE:** Visual Studio Code and WebStorm
- **Package Manager**: Node Package manager (npm) for dependency management
- **Collaboration:** Discord for communication and collaborating
- **Documentation:** Google Docs
- **Supported Browsers:** Chrome, Firefox, Edge, and Safari

8. Team and Roles

- **Shriya Dandin:** Team Lead & Document Master
- **Victoria Bialorucki:** Front End Lead & System Administrator
- **Sam Zandiasadabadi:** Back End Lead
- **Erika Chan:** Front End Team
- **Omar Ahmed:** Back End Team
- **Jake Klopukh:** Github Master
- **Subodh Khadka:** Database Engineer

9. Checklist

- Team found a time slot to meet outside of the class **(DONE)**
- Github master chosen **(DONE)**
- Team decided and agreed together on using the listed SW tools and deployment server **(DONE)**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing **(DONE)**
- Team lead ensured that all team members read the final M1 and agree/understand it before submission **(DONE)**
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) **(DONE)**