

# **Milestone 3**

## **SimpleWrestling**

### **Team 6**

Michael David: Team Lead

Huy Nguyen: Back End Lead

Michael Swanson: Front End Lead

Alexander Nunez: Back End Developer

Sarah Heewon: Front End Developer

April 4, 2019

# Table of Contents

<b>Revisions</b>	<b>2</b>
<b>1. Data Definitions</b>	<b>3</b>
<b>2. Functional Requirements</b>	<b>4</b>
Priority 1 Requirements:	4
<b>3. UI Mockups + Storyboard</b>	<b>5</b>
Landing Page:	5
Sign Up Page:	6
Login Page:	7
Admin Dashboard:	8
Coaches Dashboard:	9
Coaches Team Registration:	10
Tournaments Search Page:	11
<b>Story Board</b>	<b>12</b>
Use Case 1:	12
<b>Use case 2:</b>	<b>14</b>
<b>Use case 3:</b>	<b>16</b>
<b>4. High-Level Architecture, DB Organization</b>	<b>18</b>
Business Rules (8):	18
ERD Based on Business Rules:	19
Database ERD Based on ERD Above:	20
<b>5. High-Level UML Diagrams</b>	<b>22</b>
<b>6. Key Risks</b>	<b>24</b>
<b>7. Project Management</b>	<b>25</b>

## Revisions

Milestone	Version	Last Modified
M2	V1	04/04/2019
M2	V2	05/21/2019

## **1. Data Definitions**

**Wrestlers** - One of our main users of interest. Wrestlers are used to populate the data of a tournament.

**Weight** - Attribute of **Wrestlers**. Weight is used to categorize wrestlers.

**Coaches** - Entities that **Wrestlers** belong to. Coaches are used to organize teams of wrestlers to compete in a tournament.

**Teams** - Entities that **Wrestlers** and **Coaches** belong to. Teams are used to categorize groups in tournaments.

**Tournaments** - Entities that are created by a **Tournament Admin**, comprised of **Brackets** and **Teams**. **Tournament** is a platform that allows all data to follow together connecting all our data sets.

**Dates** - Attribute of **Tournament**.

**Brackets** - Sub-entities of **Tournaments**, comprised of **Matches**. The Brackets are used to display the winner of the match to have relevant information to update the bracket.

**Score** - Attributes of **Wrestlers**, and **Matches**, created by **Scorecard**. The score is to determine which wrestler wins.

## **2. Functional Requirements**

### **Priority 1 Requirements:**

#### **1. Tournament Admins**

**P1, 1.1: Tournament Admins** shall create a new table of Tournaments

**P1, 1.2: Tournament Admins** shall delete a table of Tournaments

**P1, 1.3: Tournament Admins** shall modify Tournaments

**P1, 1.4: Tournament Admins** shall create Brackets within Tournaments

**P1, 1.7: Tournament Admins** shall approve or decline Coaches request

#### **2. Coaches**

**P1, 2.1: Coaches** shall join Tournaments

**P1, 2.2: Coaches** shall add Wrestlers to Teams

**P1, 2.3: Coaches** shall delete Wrestlers from Teams

**P1, 2.4: Coaches** shall modify Teams

**P1, 2.6: Coaches** shall make a request to join a Tournament

#### **3. Visitors**

**P1, 3.1: Visitors** shall search tournaments by Date

**P1, 3.2: Visitors** shall search tournaments by Team

**P1, 3.3: Visitors** shall view Brackets

**P1, 3.4: Visitors** shall view Tournaments

### **Priority 2 Requirements:**

#### **1. Tournament Admins**

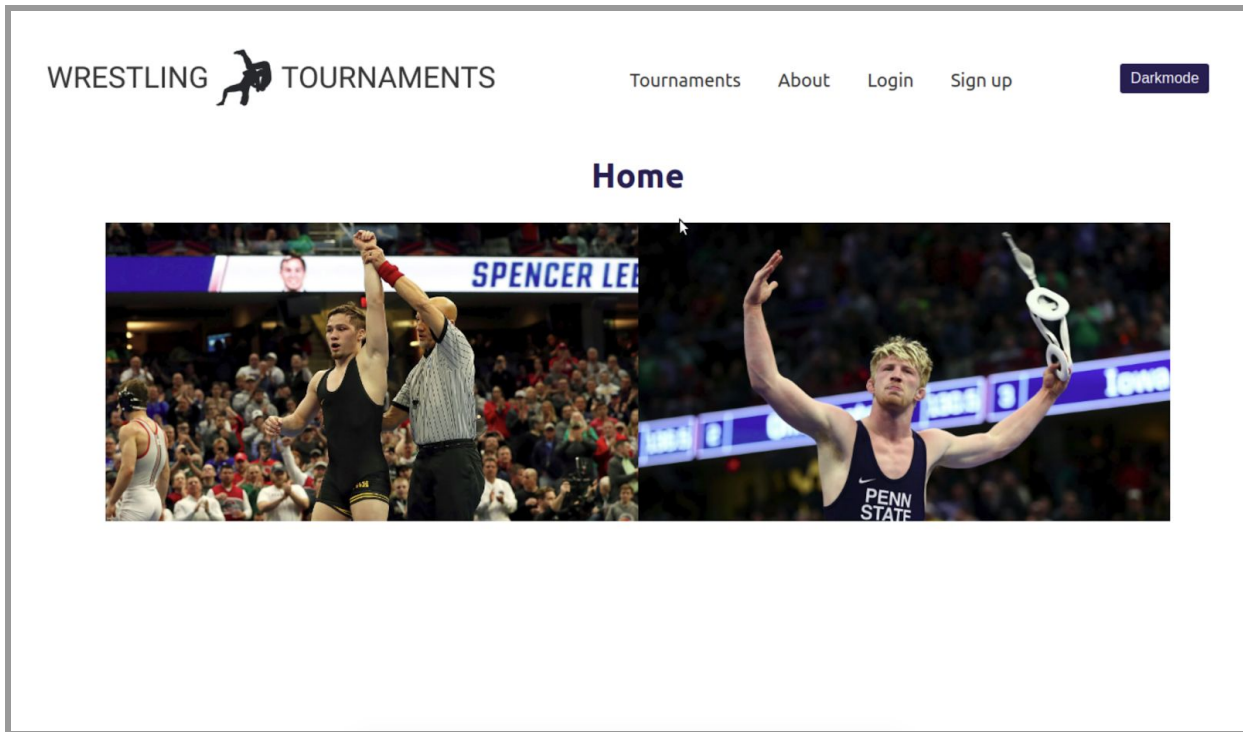
**P2, 1.9: Tournament Admins** shall manage Tournaments from multiple regions

#### **2. Coaches**

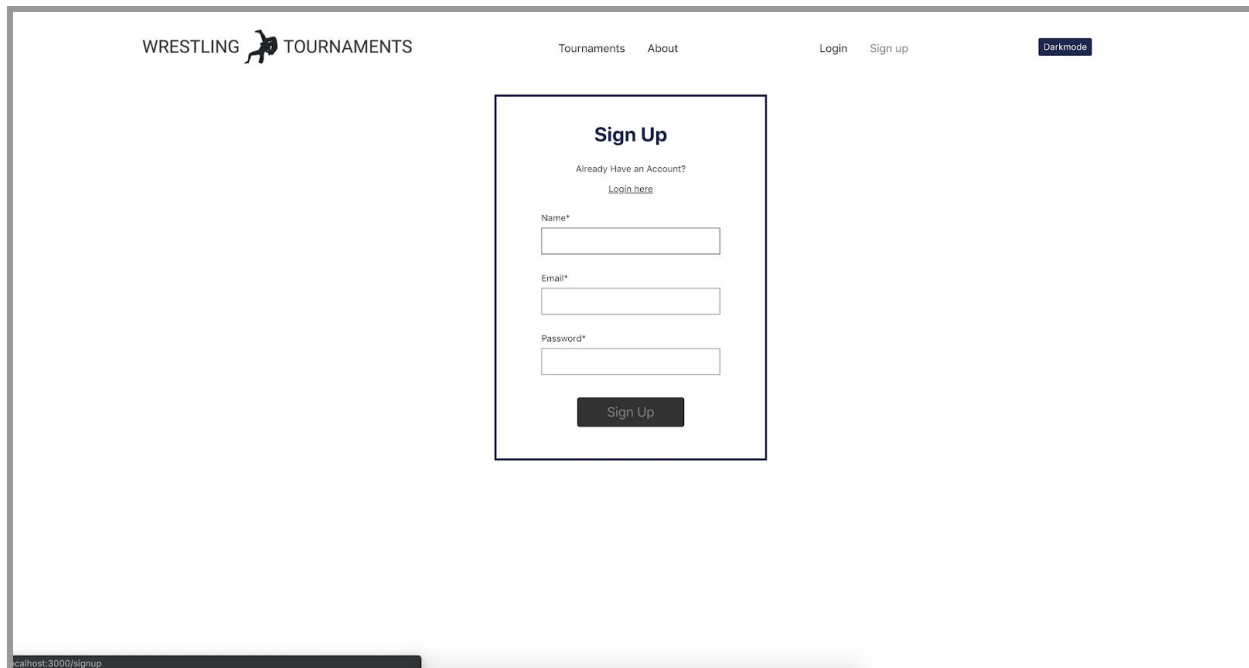
**P2, 2.7: Coaches** shall manage multiple Teams

### 3. UI Mockups + Storyboard


Landing Page:



## Sign Up Page:



The screenshot shows a web page for "WRESTLING TOURNAMENTS". The header includes the site name with a logo, navigation links for "Tournaments" and "About", and user links for "Login" and "Sign up". A "Darkmode" toggle is in the top right. The main content area features a "Sign Up" form with the title "Sign Up", a link for existing users to "Login here", and input fields for "Name\*", "Email\*", and "Password\*". A "Sign Up" button is at the bottom of the form. A terminal bar at the bottom left shows the URL "localhost:3000/signup".

WRESTLING  TOURNAMENTS

Tournaments About Login Sign up Darkmode

### Sign Up

Already Have an Account?  
[Login here](#)

Name\*

Email\*

Password\*


Sign Up

localhost:3000/signup

Fixed Sign Up page by removing the “Forgot Password?” option as it was not needed for new members to see.

## Login Page:

We are giving three different options for registering. That is one for coaches, one for tournament administrators, and the last one is for the wrestlers themselves. Once users have their information registered with us, and they have logged in, they will be able to access their dashboards as well as all other features of the site.

WRESTLING  TOURNAMENTS

Tournaments About Login Sign up 

Darkmode

Login

Don't Have an Account?  
[Sign up here](#)


Email\*

Password\*

Login

[Forgot Password](#)

# Admin Dashboard:

WRESTLINGTOURNAMENTS

TournamentsAboutLoginSign upDarkmode

Admin Dashboard


Northern CaliforniaSouthern California

Tournaments Dashboard				Save
Name	Location	Picture	Background	
San Francisco Classic	San Francisco, CA	<a href="https://imgur.com/CyatTnp.png">https://imgur.com/CyatTnp.png</a>	<a href="https://imgur.com/Uoudtg.jpg">https://imgur.com/Uoudtg.jpg</a>	X
California State Something	Sacramento, CA	<a href="https://i.imgur.com/4eeUuAC.jpg">https://i.imgur.com/4eeUuAC.jpg</a>	<a href="https://i.imgur.com/00Tsf8A.jpg">https://i.imgur.com/00Tsf8A.jpg</a>	X
Chad's House	Chad's House	<a href="https://i.imgur.com/hrDntr.jpg">https://i.imgur.com/hrDntr.jpg</a>	<a href="https://i.imgur.com/Zwv6Zof.jpg">https://i.imgur.com/Zwv6Zof.jpg</a>	X

Add Row



Coaches Dashboard:

WRESTLINGTOURNAMENTS

TournamentsAbout

LoginSign up

Darkmode

Coach Dashboard

The Teststerinos

Team 2

Team 3

The Teststerinos Dashboard

Name	Age	Weight	Class	Record
Michael	20	170	160	7/2
Mark	22	180	180	6/1

Name

Age

Weight

Class

Record

| Bob | 18 |  |  |  |
| Bill | 19 |  |  |  |

Name

Age

Weight

Class

Record

| Larry | 21 | 170 | 160 | 7/2 |

Add Row

# Coaches Team Registration:

Tournament name

Enter a tournament

wrestler name

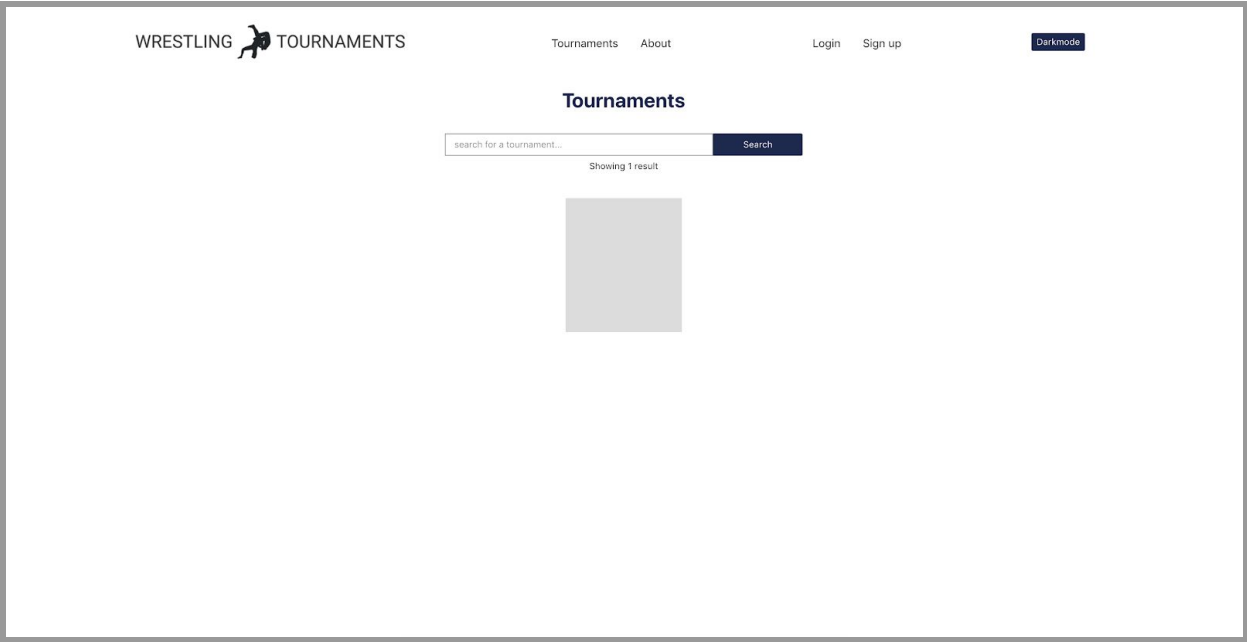
find profile

wrestler age

wrestler weight class

seeding information

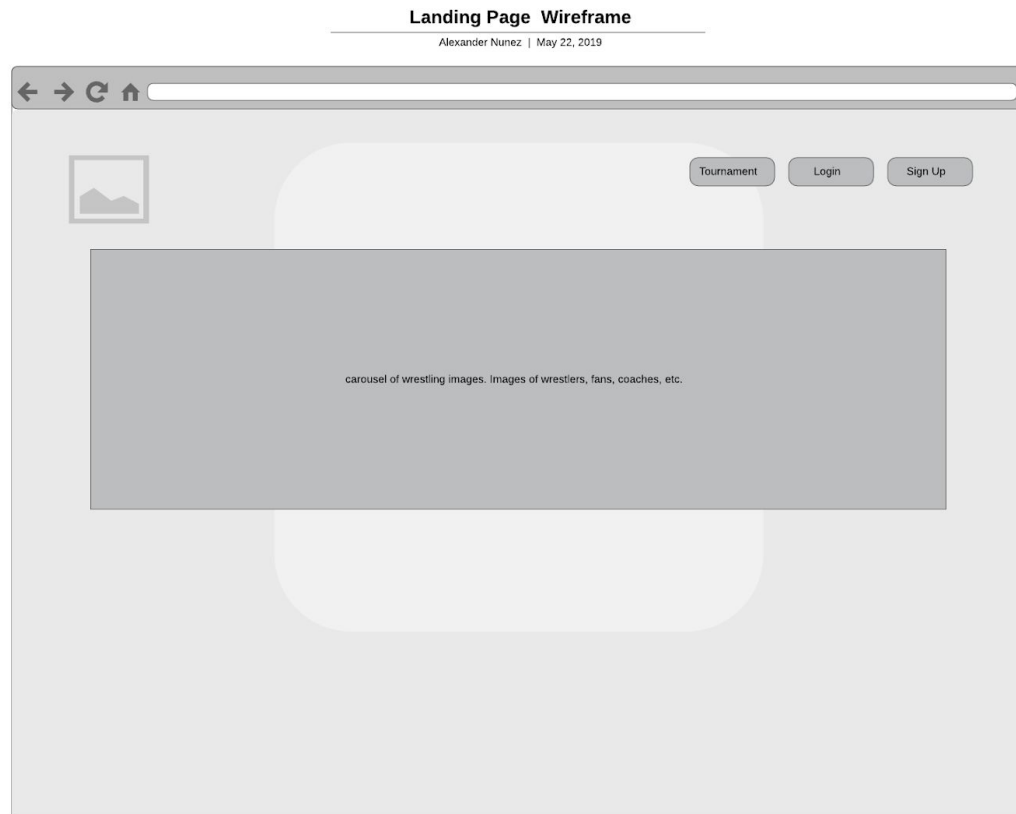
Tournaments Search Page:



# Story Board:

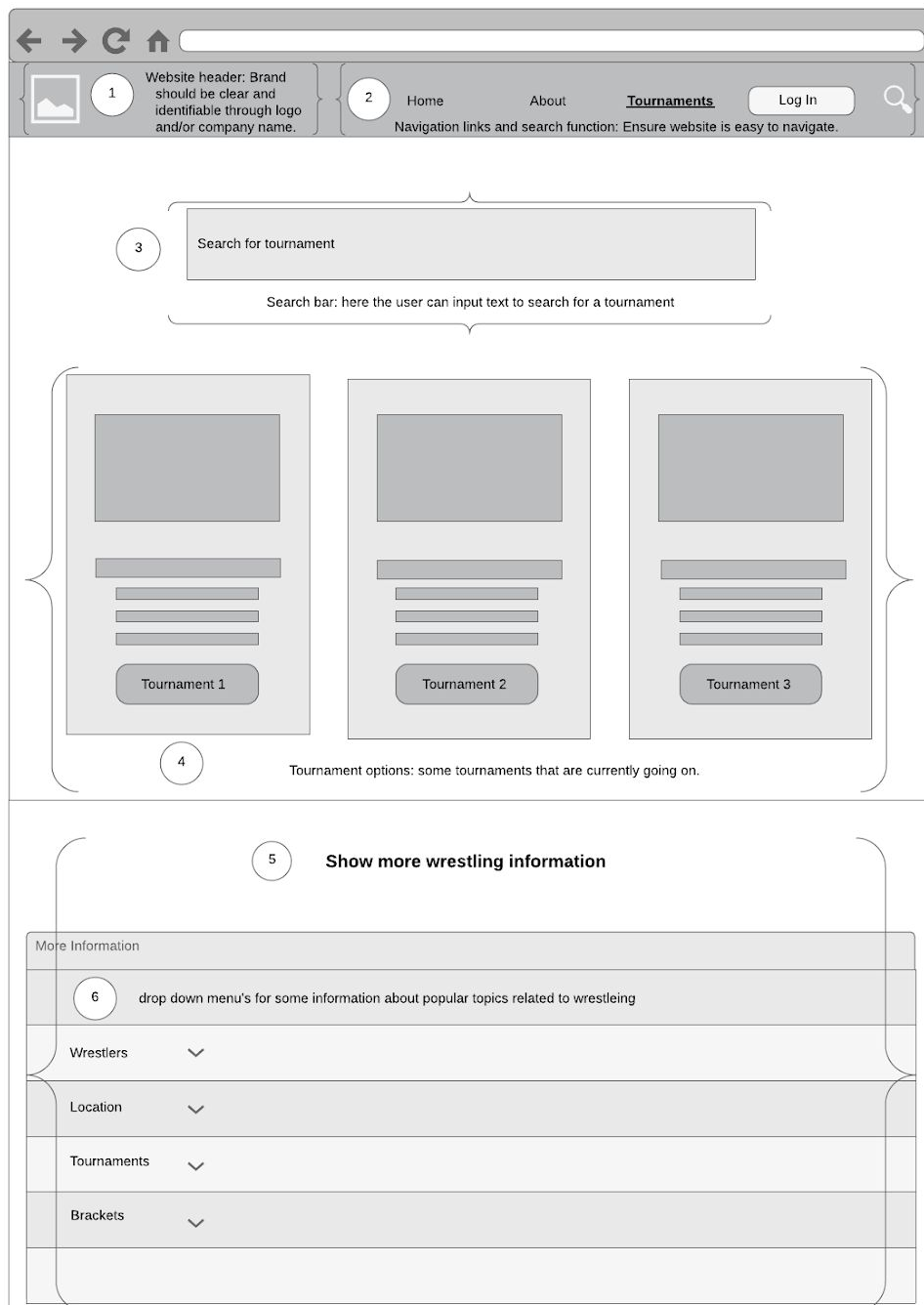
## Use Case 1:

A family member of a wrestler wants to know when the wrestler is going to compete. They visit the website, reach the landing page, select tournaments, search for wrestlers name.



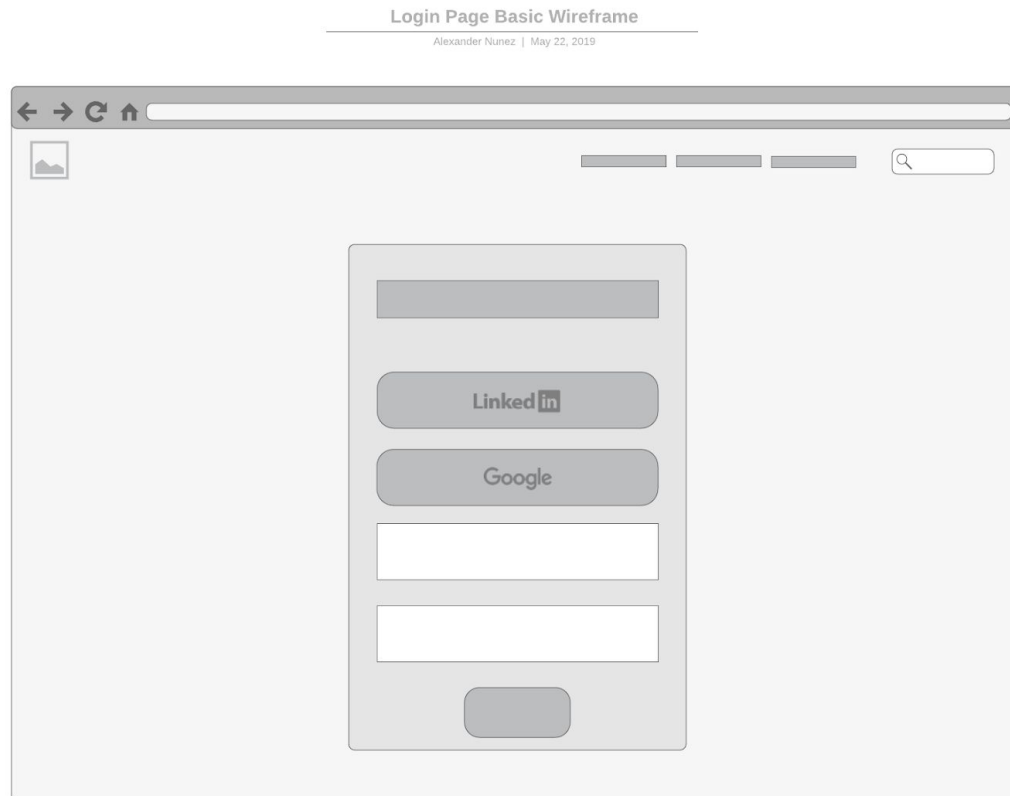
## Tournament Page Wireframe

Alexander Nunez | May 22, 2019



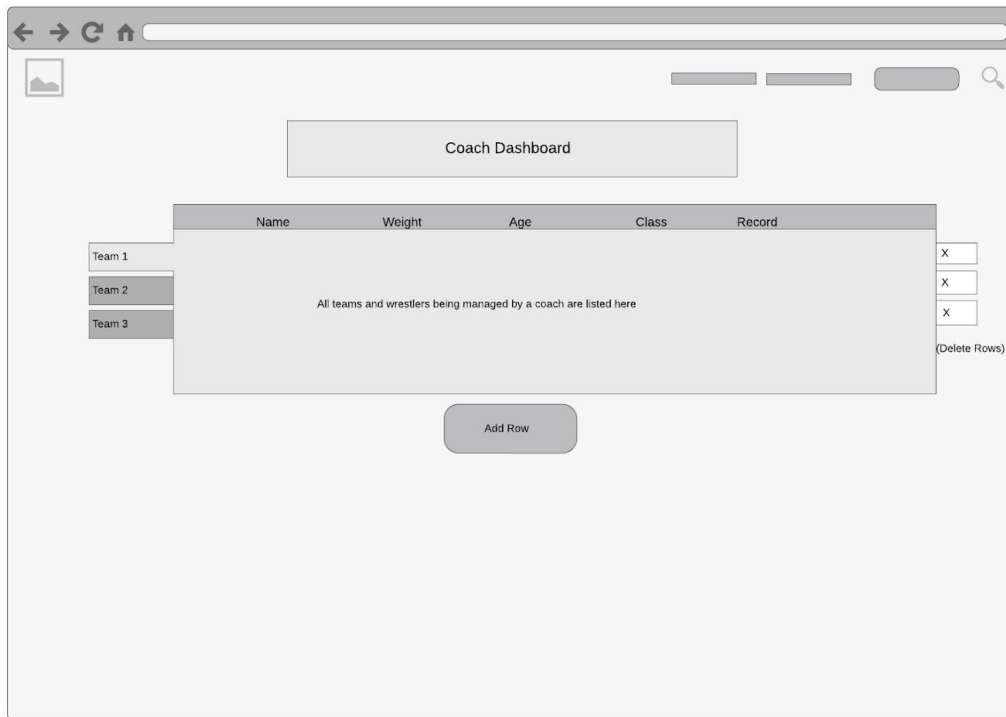
## Use case 2:

A coach wants to add a new wrestler to his team. The coach visits the login page, enters his credentials, and is directed to the coach's dashboard page where he can add, delete, modify



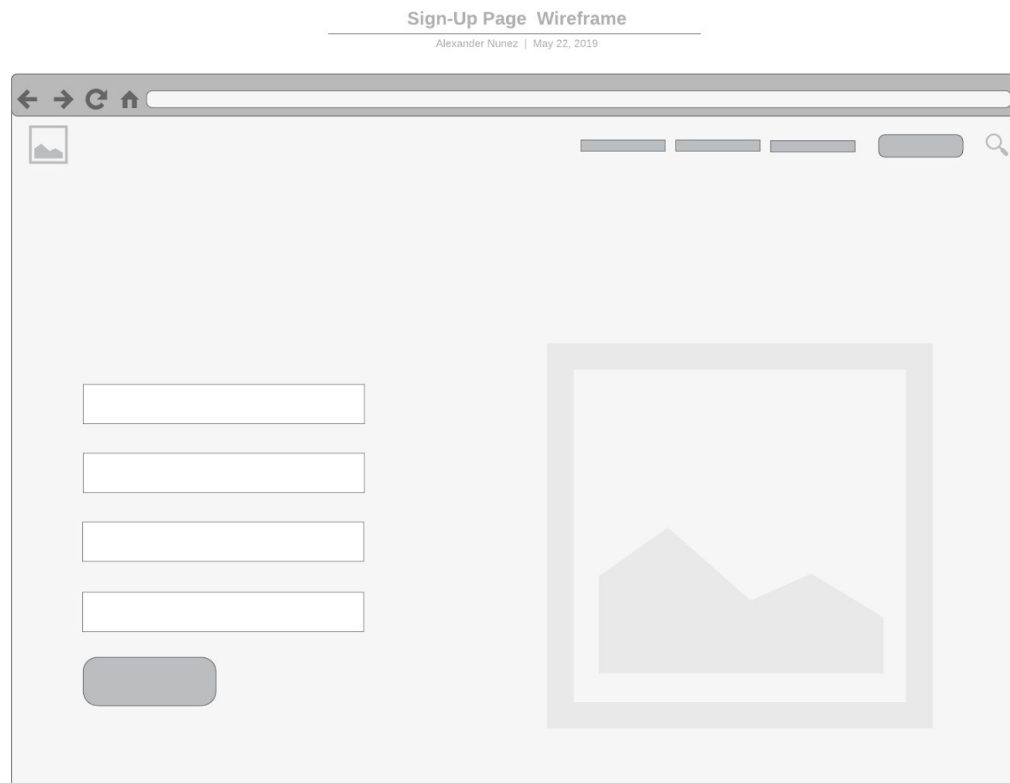
## Coach Dashboard Wireframe

Alexander Nunez | May 22, 2019



### Use case 3:

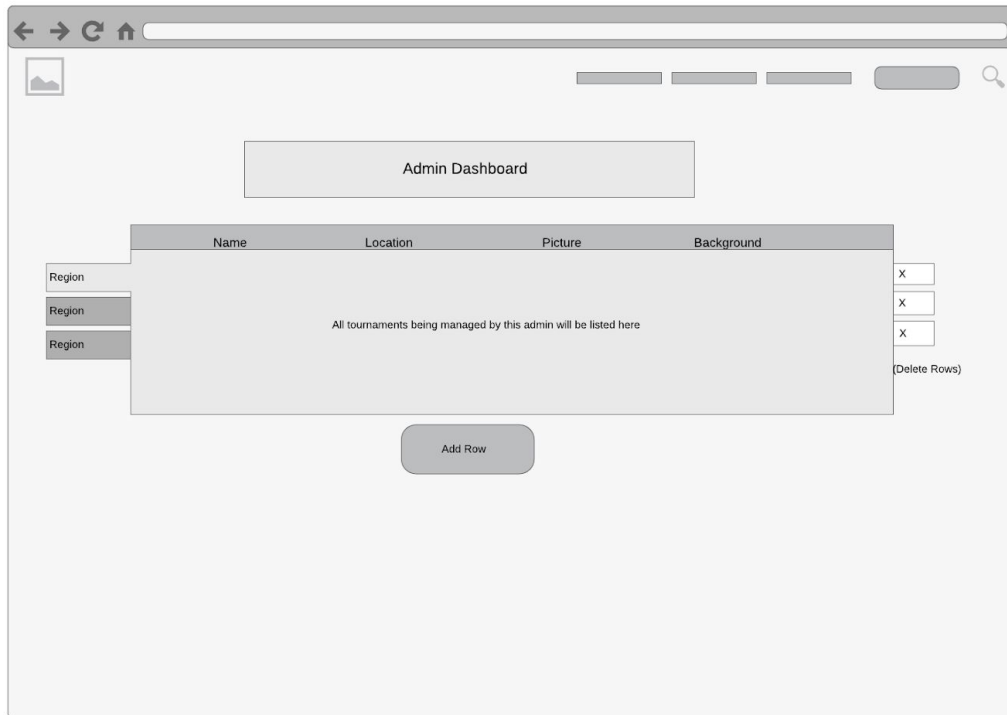
An administrator wants to use the website to manage some tournaments. She visits the Sign Up page, enters credentials, and is then granted access to an administrators dashboard





## Admin Dashboard Wireframe

Alexander Nunez | May 22, 2019

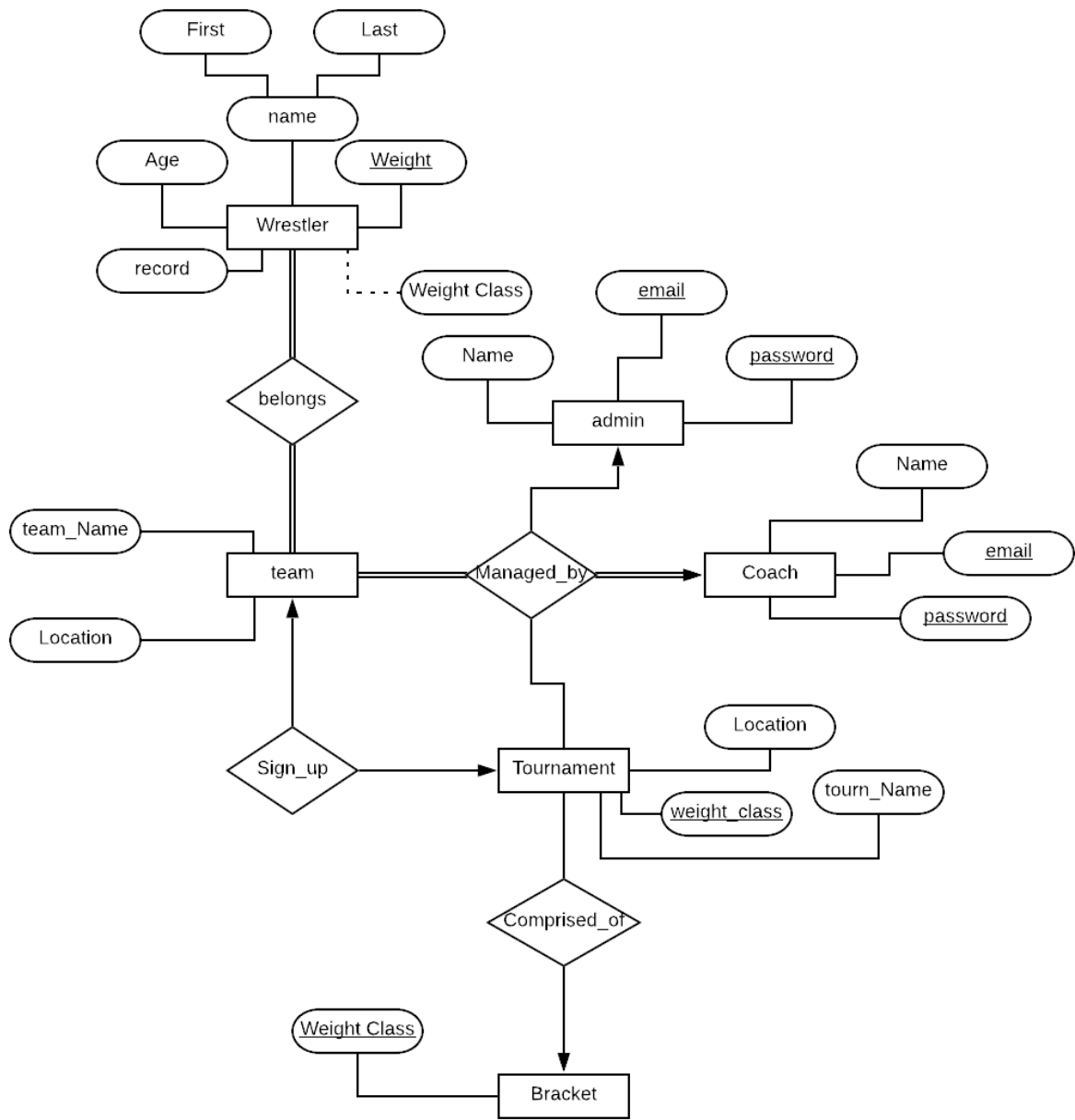


#### **4. High-Level Architecture, DB Organization**

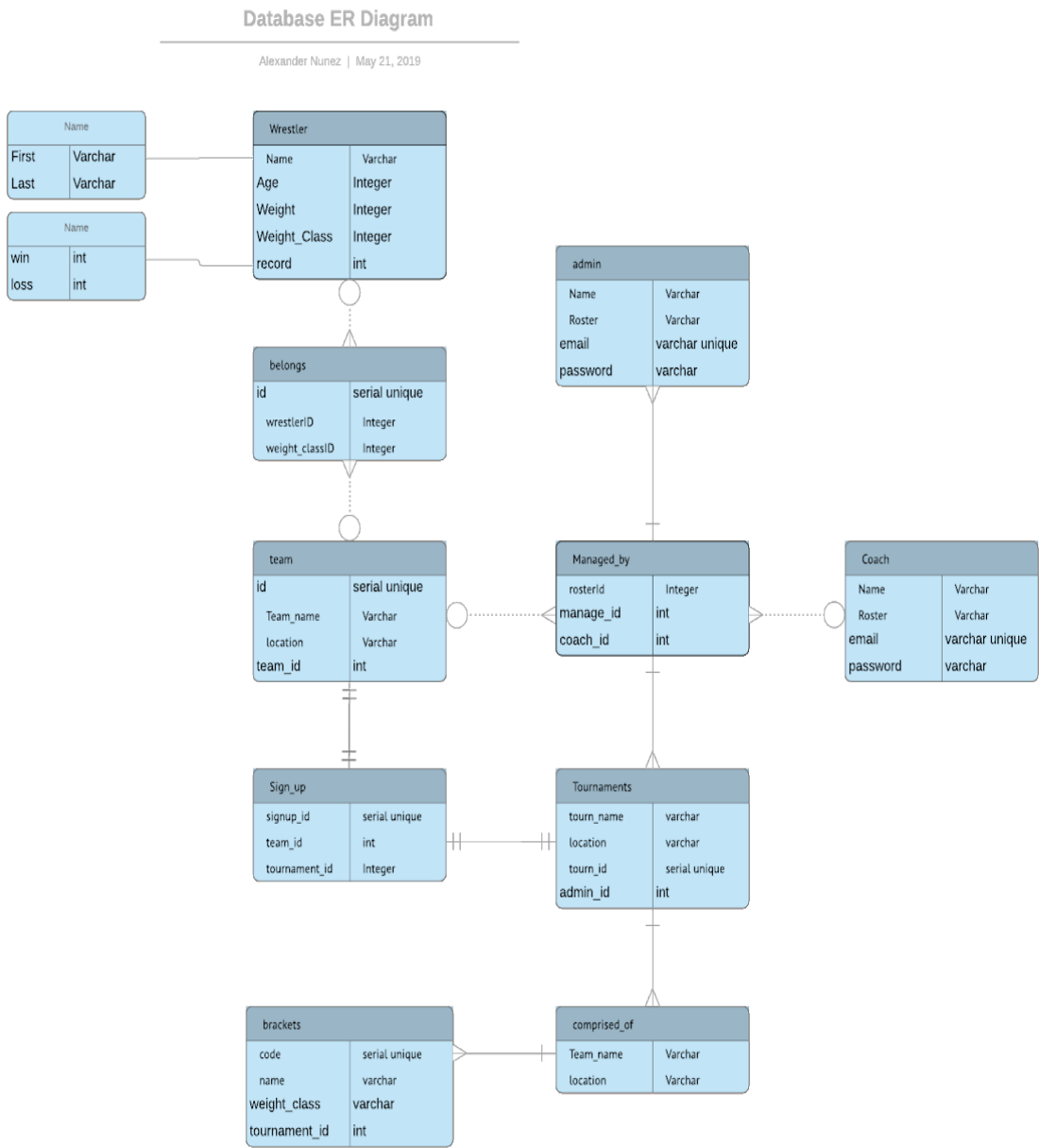
##### **Business Rules (8):**

1. **Tournament Admins** shall create a new table of Tournaments
2. **Tournament Admins** shall create Brackets within Tournaments
3. **Tournament Admins** shall assign Wrestlers to Brackets based on Weight Class
4. **Coaches** shall be able to join Tournaments
5. **Coaches** shall be able to add Wrestlers to Rosters
6. **Coaches** shall be able to log the Wrestlers weight, age, and name in

**ERD Based on Business Rules:**



Database ERD Based on ERD Above:



We will use PostgreSQL as our DBSM because it is a DBSM that some of our team members were more familiar with, which reduced the learning curve for other team members.

Media Storage: all media will be stored in a file system.

### **Search/filter architecture and implementation:**

We will be using Full-Text Search for PostgreSQL. Searching provides the capability to identify natural-language *documents* that satisfy a *query*, and optionally to sort them by relevance to the query. The most common type of search is to find all documents containing given *query terms* and return them in order of their similarity to the query. Notions of query and similarity are very flexible and depend on the specific application.

The searched items will be organized in a way that the most relevant (relevant to the input from the user) will appear first at the top of the results.

The DB terms to be searched are “Coach name”, “tournament name”, tournament date”, “wrestler name”, “tournament location”

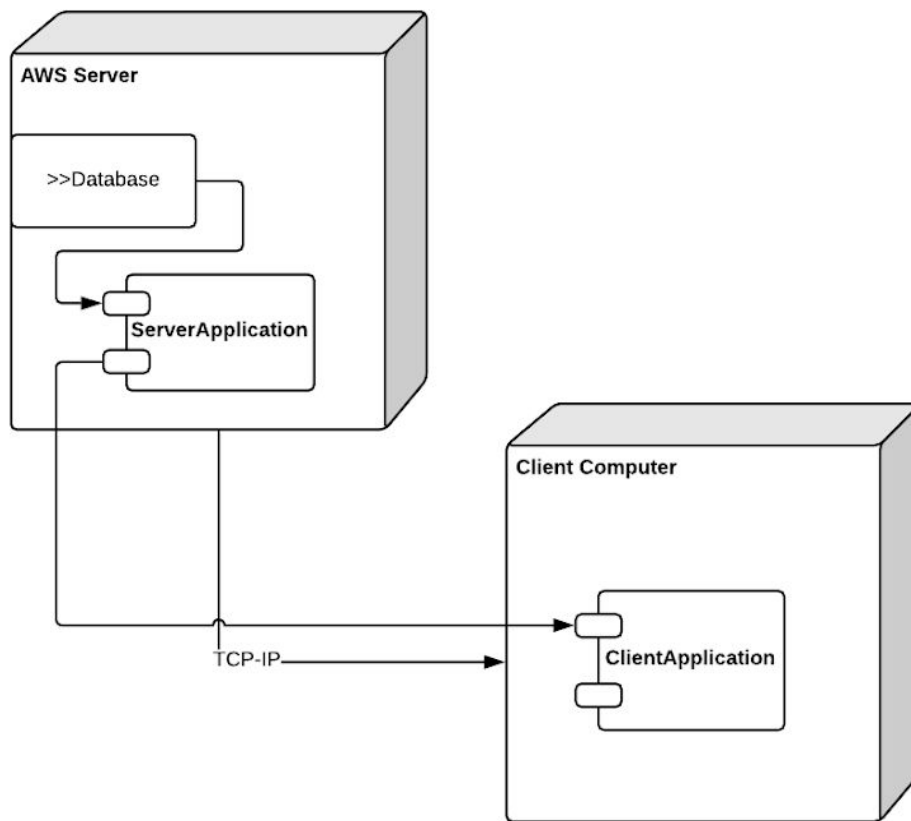
DB Organization:

```
SELECT ' (Wrestler name) & tournament'::tsquery @@ 'Miguel Hernandez
collegiate finals tournament'::tsvector;
?column?
-----
f
```

## 5. High-Level UML Diagrams

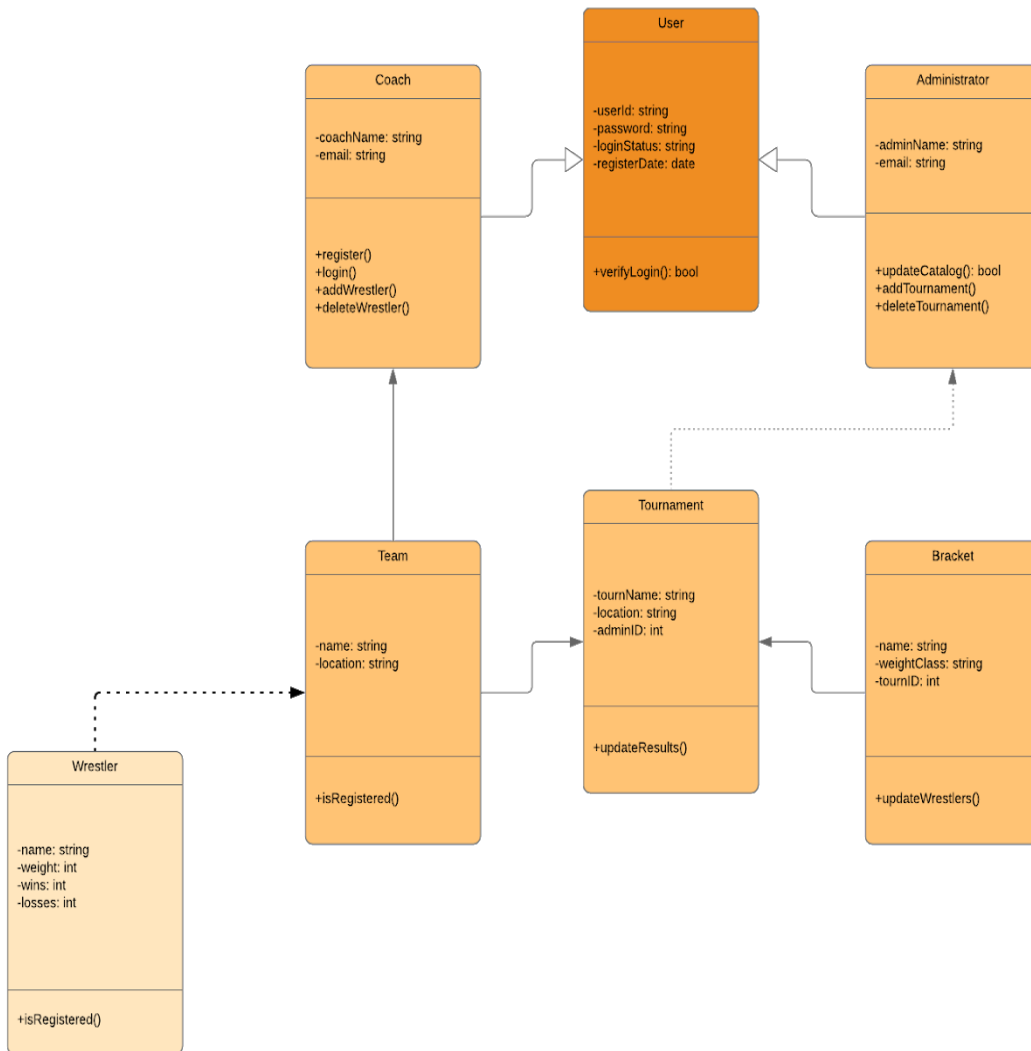
### Deployment Model

Michael David | April 4, 2019



## UML Class Diagram

Alexander Nunez | May 21, 2019



## 6. Key Risks

### **Skill Risk**

The Skill Risk we will have are not having the technical skills learn the technologies in the time we are given. We are all new to the web development industry. I think it is extremely difficult for the team to learn a new language and framework to complete this project. The team is capable of learning these technologies however in the given time allowed it is extremely difficult. On top of that, our website is unique and not everyone on the team was aware of how wrestling tournaments operated.

Our solution is to educate others on what we know and explore the technologies together. The goal is to gain skills independently then share that information with the rest of the team so we constantly learning and improving. We are taking advantage of many online resources to learn the necessary skills.

### **Schedule risk**

The Schedule risk is always going to be an issue because we mapped out our timelines of when to each our goals however as young developers we do not know when we are going to hit a roadblock or bug. These issues can take days or weeks to fix. So the time from of the project is was going to be an issue as time gets closer to the deadlines.

Our solution to this issue is trying to communicate well with the team to bring up our issues as they come up. We have to communicate well with our team when we can not meet our deadlines. So can help in areas where we need help. This issue is something we saw happening as the semester progressed. We understood that everyone has busy schedules and other obligations. So a solution was to only use Javascript-based languages because switching from React.js to Node.js is much easier than React.js to Python. This means it is much easier to help out the areas that need the most help because everything is done in Javascript.

### **Teamwork Risk**

The risk of working in any team of strangers is developing trust. The issue is that we do not know each other and we are unaware of each other's development skills and soft communications skills. Sometimes these risks are unavoidable. Also, given that everyone is a student and has different schedules, it becomes difficult to communicate with everyone

Our Solution is solving this issue is clear communication. The constant struggle of making sure everyone stays on task would be to communicate often with the team. I think it would be extremely helpful to the team to just speak up when you're having trouble or just general questions or thoughts, but also have the team lead to check in to make sure everyone is on task in case the group is too shy to speak up.

### **Legal/content Risk**



The Legal and Risk problems of this project would be the copyright of possibly using similar software to Trackwrestling who is the current king of the market. I can see some legal issues that they might try to pursue once users begin to realize our product is much more user-friendly and a better experience.

The solution to this issue that our team came up with is to take a deeper look into Trackwrestling and other potential legal issues with this type of software because if we plan on launching this product to the market we have to be clear it is safe and legal. After our research, we can be approved with proper licensing as there are no restrictions.

## **7. Project Management**

We were able to manage the task for M2 in an okay manner we assigned check-ins every few days to make sure everyone is on the same page development, however, this comes with some unavoidable issues being that many students are very busy outside this course. It was hard to keep track of everyone's task. We are in the process of attempting the process of doing a pyramid style of management. The style being having the Frontend and Backend Lead run their own teams and report to the Team Lead. However, I think in the future we will use Trello just to make sure everyone can clearly see what everyone is doing. I think it would also be more helpful to see where everyone is on the development cycle that way we can locate which areas need help and help out.