

# **SimpleWrestling**

Team 06

**Michael David:** Team Lead/Document Master

Email: mdavid2@mail.sfsu.edu

**Michael Swanson:** Front End Lead

**Huy Nguyen:** Back End Lead

**Heewon (Sarah) Han:** Github Tournament Admin

**Alexander Nunez:** Developer

**Tuan Le:** Developer

**Milestone 1**

3/14/2019

## Table of Contents

<b>Revisions</b>	<b>2</b>
<b>1. Executive Summary</b>	<b>3</b>
<b>2. Personae and Main Use Cases</b>	<b>4</b>
Use Case: Wrestler	4
Use Case: Tournament Administrator	4
Use Case: Coach	4
Use Case: Visitor	5
<b>3. List of Main Data Items and Entities</b>	<b>5</b>
<b>4. Functional Requirements</b>	<b>7</b>
<b>5. Non-Functional Requirements</b>	<b>8</b>
Security:	8
Audit:	8
Performance:	8
Capacity:	9
Reliability:	9
Recovery:	9
Data Integrity:	9
Compatibility:	10
Conformance with Coding Standards:	10
Look and Feel Standards:	11
Internationalization / Localization Requirements:	11
<b>6. Competitive Analysis</b>	<b>12</b>
<b>7. System Architecture and Technologies Used</b>	<b>13</b>
<b>8. Team</b>	<b>13</b>
<b>9. Checklist</b>	<b>13</b>

### Revisions

Milestone	Version	Last Modified
M1	V1	03/14/2019
M1	V2	05/21/2019

# **1. Executive Summary**

The purpose of this website is to solve an ongoing issue that the wrestling community is facing with outdated technologies. Our group has noticed a trend that if your sport is not popular it gets outdated. That is the reason why sports such as Football, Baseball, and soccer get such strong backing. So a sport such as wrestling is often overlooked.

Our website offers modern technology into a sport that is growing. We are able to offer a modern friendly user interface for our users to easily understand. We also offer the Admin team of a wrestling tournament a clean bracket system making running tournaments easier.

We also plan on launching a prototype of this product to the hands of the wrestling community based on one of the members close attachment to the community for a test base during the summer time.

Once we launch we plan on making a simple business plan o the application by making the Admin pay by the match.

Our mission is to just offer a modern spin on an industry that is old and one that no one cares about. We also plan on making a huge positive impact on this sport as it continues to grow.

## 2. Personae and Main Use Cases

### Use Case: Wrestler

**Wrestler:** Huy, a college wrestler, is competing in a tournament. He would like to know his position in the tournament bracket and would also like to see who his next potential opponent may be. He logs on to the website, selects the tournament for which he is already registered for, and views the brackets. Huy has no issues navigating the website because he has been using computers and the internet for the majority of his life.

### Use Case: Tournament Administrator

**Tournament Admin:** A retired SFSU wrestling coach, Jose, has volunteered to manage a wrestling tournament that Huy will be competing in. At the end of the tournament, he must log in to the website, choose the tournament for which he is volunteering for, and update the brackets depending on the results of that day's matches. Jose deletes the losers accordingly and moves the winners into a new bracket. Jose has basic WWW skills and has done all of this by hand in the past. He's grateful to have a tool like this website to do most of the work for him.

### Use Case: Coach

**Coach:** SFSU's wrestling coach has just welcomed a new batch of freshman wrestlers, and is saying goodbye to Huy and the rest of his senior class. He logs in to the website, navigates to his roster of wrestlers, and begins to delete his old wrestlers, including Huy, and adds his new wrestlers. Once it is time to compete, he searches for a tournament and makes a request to join. He has basic WWW skills and knows how wrestling tournaments work so he has no real issues working with the website.

### **Use Case: Visitor**

**Visitor:** Huy's CSC 648 team would like to show support during one of his wrestling matches. They do not remember the time or day of the tournament. They visit the website without having to log in and search for the tournament. Once they find it, they select it and are prompted with general information about the tournament including the time and day. Huy's CSC 648 team also consists of college students, and they navigate the website with ease.

## **3. List of Main Data Items and Entities**

**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**. Tournaments are platforms that allow 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.

**Matches** - Sub-entities of **Brackets**, Matches contain 1 or 2 **Wrestlers**, will contain a flag **Bye** and a **Score**. The matches are used to display the score of two wrestlers to update the bracket.

**Mats** - Entities that contain **Matches** in an ordered sequence. Mats are used to make sure the wrestlers have a place to compete and they need to compete in order to update the brackets.

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

**Scorecard** - Entity that displays the connection between **Wrestlers**, **Matches**, and **Mats**. The scoreboard is used to determine the winner of the match and how the wrestlers won. A wrestler can win in any of the following ways: Minor Decision, Major Decision, Tech Fall, Pin Fall, Overtime, Bye Injury, default, and Forfeit.

## **4. Functional Requirements**

### **1. Tournament Admins**

- 1.1 Tournament Admins** shall create a new table of Tournaments
- 1.2 Tournament Admins** shall delete a table of Tournaments
- 1.3 Tournament Admins** shall modify Tournaments
- 1.4 Tournament Admins** shall create Brackets within Tournaments based on Weight Pools
- 1.5 Tournament Admins** shall assign Wrestlers to Brackets based on the Wrestler's Weight and the Bracket's Weight Pool
- 1.6 Tournament Admins** shall review and edit Scores of Matches
- 1.7 Tournament Admins** shall approve or decline Coaches request
- 1.8 Tournament Admins** can assign Matches to Mats

### **2. Coaches**

- 2.1 Coaches** shall join Tournaments
- 2.2 Coaches** shall add Wrestlers to Rosters
- 2.3 Coaches** shall delete Wrestlers from Rosters
- 2.4 Coaches** shall modify Rosters
- 2.5 Coaches** shall manage multiple Rosters
- 2.6 Coaches** shall make a request to join a Tournament
- 2.7 Coaches** can track which Mats their Wrestlers need to go to next

### **3. Wrestlers**

- 3.1 Wrestlers** shall view Brackets
- 3.2 Wrestlers** shall view Tournaments

### **4. Visitors**

- 4.1 Visitors** shall search tournaments by Date, Team, and Wrestler
- 4.2 Visitors** shall subscribe to a Wrestler or Team

## **5. Non-Functional Requirements**

### **Security:**

1. Login shall be mandatory for host and Tournament Admin.
2. Username shall be the user's registered email
3. Password shall be encrypted before saving it in the database
4. User's session shall be ended after 30 minutes of inactivity
5. User's session timeout limit shall be set up by the Tournament Administrator
6. User's session shall only be ended by code design
7. Uploaded content from any part shall be audited by the Tournament Administrator
8. This site shall not accept third-party cookies

### **Audit:**

1. New registrations shall be audited by the Tournament Administrator
2. New registrations shall be approved by the Tournament Administrator
3. The site Tournament Administrator shall be the only person authorized to configure the host interfaces
4. The host shall not be allowed to modify any web configuration files
5. The host shall not be allowed to login into the Tournament Administrator page
6. Users, registered or not, shall not be able to login into Host and Tournament Admin pages

### **Performance:**

1. The site loading time shall be less than 1 second for all the screens
2. The search function shall be executed in a background thread for improving performance
3. Query shall be executed in a background thread for improving performance



**Capacity:**

1. The total data storage allowed by the web site shall not exceed of 80 % of the server capacity for this site
2. The web site shall be prepared to support scalability for adding future new features
3. The web site shall be capable to handle at least 50 users

**Reliability:**

1. Downtime for maintenance shall be less than 1 hour per month
2. Downtime for maintenance shall not affect the main functionality of the site
3. In all cases, downtime for maintenance shall be informed to the users either by email
4. In all cases, downtime for maintenance shall be informed to the users by publishing an announcement in a visible place of the main page.

**Recovery:**

1. In a total failure case, the whole site should be put down to revision
2. If broken, the mean time to recovery shall not exceed one day

**Data Integrity:**

1. Database tables shall be backed up every week
2. Tournament Administrator shall be able to execute a recovery when needed
3. Images size shall be limited up to 1 megabyte
4. Images shall be uploaded in the correct format (jpg or jpeg)
5. Video content shall be uploaded only by Tournament Admins in a compressed format such as mpg4

**Compatibility:**

1. The site shall be compatible with the last version of Safari browser:  
12.1.1
2. The site shall be compatible with the last version of Firefox browser:  
66.0.5
3. The site shall be compatible with the last version of Chrome browser:  
74.0.3729.159
4. The site shall be compatible with at least an old version of all the browsers listed above
5. Third party applications shall not be able to modify any content that may affect the site compatibility
6. The site shall be ready to support with any or minimal changes any other compatibility that may be added in future versions
7. The site should be compatible to escalate to new databases

**Conformance with Coding Standards:**

1. Architecture and design standards shall meet all the requirements listed under the High-Level Architecture section of this document
2. Only working code that meets all the code standards shall be submitted to the project repository
3. Any working code shall be tested and debugged before being considered to be working code
4. Any internal errors or exceptions returned by the code shall be stored in a log
5. Any error that may affect the functionality of the site shall be reported to the user
6. Any error shall be handled in a way that does not affect the functionality of the site.
7. The whole production cycle of this site shall be finished 2 weeks before the delivery date
8. This site shall not be launched without all the priority one featured finished and tested
9. This site shall be tested and debugged as a whole 2 weeks before the delivery due date

**Look and Feel Standards:**

1. The application and its layouts shall look professional
2. The site shall be enough simple to handle by all the parties involved
3. Elements on the screen shall have the correct density to meet the compatibility standard of the browsers
4. Elements on the screen shall have the correct size to meet the compatibility standard with all mobile devices
5. Elements on the screen shall have rich and beautiful colors for user delight
6. The site shall be able to work correctly without mouse interaction
7. The site shall be able to work correctively without keyboard interaction
8. Elements in screen shall be resized automatically without user interaction when being loaded in all the different platforms supported by the site

**Internationalization / Localization Requirements:**

1. The default language of this site shall be English
2. The site shall support scalability to add more supported languages in future versions
3. The site shall support geolocation to locate tournament addresses in a map.

## 6. Competitive Analysis

Competitors	TrackWrestling	InterMat	FloWrestling	TheOpenMat	Us
Search by Wrestler	-	-	-	-	+
Dynamic Bracket Seeding	+	-	-	-	+
Live tournament Mat Update	-	-	-	-	+
Following Individual Wrestlers	-	-	-	-	+
Tournament Maintenance Portal	-	-	-	-	+

The **search by wrestler** functionality is something that our competitors are missing. A lot of the time, family members or friends of the wrestler who want to show support, do not know anything about the wrestling world. All that they may know is the name of their friend or family member who is competing. That is why we saw it important to have a feature that allows you to search a wrestler and see what tournaments they will be participating in. **Dynamic Bracket Seeding** is a popular form of managing brackets and is intuitive. **Live Tournament Mat Updates** are important to us because sometimes matches may be cut short, not even happen, or be delayed. Having live updates can help viewers gauge how long they have until the wrestler they wanted to watch, will compete. **Following Individual Wrestlers** will help fans stay updated with their favorite wrestlers. The **Tournament Maintenance Portal** is important when having to deal with technical issues.

## 7. System Architecture and Technologies Used

Server Host: *Amazon Web Services*

Operating System: *Debian Ubuntu 16.04*

Databases: *MongoDB, PostgreSQL 11.2*

Web Server: *NodeJS 10.13.0*

Server-Side Language: *JavaScript ES6*

Additional Technologies: Web Framework: *React 16.7*

Node Library: *Express 4.16.4*

IDE's: *Sublime, VS code, Atom*

## 8. Team

- Michael David: Team Lead/Document Master
- Michael Swanson: Front End Lead
- Huy Nguyen: Back End Lead
- Heewon (Sarah) Han: Github Tournament Admin
- Alexander Nunez: Developer
- Tuan Le: Developer

## 9. Checklist

- The team found a time slot to meet outside of the class  
**ON TRACK**
- Github master has been chosen  
**DONE**
- The 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

### **ON TRACK**

- Team lead ensured that all team members read the final M1 and agree/understand it before submission

### **ON TRACK**

- Github organized as discussed in class (e.g. master branch, development branch, the folder for milestone documents, etc.)

### **DONE**