

Head to Head Records

Dev Environment

- Xampp (run Apache, MySQL)
- Import **mi_melee_h2h.sql** into a new phpMyAdmin database
- Move the folder **mi_melee_h2h** to the xampp directory: **xampp/htdocs**
- Go to **localhost/mi_melee_h2h**

Concept Introduction

I am part of the competitive Super Smash Brothers Melee (2001) community in Michigan. Due to the age of the game, there was no online play for quite some time and as a result, the community has gotten accustomed to meeting in-person at events to compete in tournaments. Many events are held each week in Michigan and many people go to the same events each week to compete. Not too long ago someone in the community decided they would keep track of all the results for all the tournaments in Michigan and enter them into a SkillKeeper ranking system based on TrueSkill. So now we have a Michigan Melee Top 100 ranking that is updated weekly, the owner of which, maintains the results in a google spreadsheet so that anyone can see it. I wanted to use the data that has already been collected to access head-to-head data from a database, displayed on a website. The owner of the data has granted me access to some of the data from the past few months so that I could get started.

Design Considerations

I wanted the page to allow the user to select 2 players, and then display the results and information from matches where the 2 players play against each other. The page would also calculate and display the players' records against each other. The data will be imported to a table in a database from Excel (since that's how I will be receiving the data and updating the database weekly). I can't think of a reason to have more than a single table as the project exists now. The only table would be a table of matches with 5 columns, matchID (not sure that it's necessary but it's the only thing that would be a primary key), tournament name, the winner of the match, the loser of the match, and the date the match was played. The 2 players would either be searched for in 2 separate search bars (preferably with search suggestions) or a dropdown menu (select). I was leaning away from the dropdown since they would likely have thousands of options.

Security Considerations

Since I was heavily leaning toward a search bar, I was concerned about user input interacting with my database, specifically SQL injection. I still believed that searching was necessary since there were

thousands of options, so I began to research a searchable dropdown. This would solve the SQL injection fears since only a fixed set of options could be passed through to a query. I ended up finding a jQuery plugin that modifies the 'select' tag in order for it to be searchable. It also provides suggestions as you type. This seemed like the perfect solution.

UI Design

The screenshot below is a prototype that I made in Google Sheets long before this project was assigned. This version of the design shows 2 dropdowns, one for each competitor, their head-to-head data in the middle columns and on the outside columns it showed their records against common opponents. Record vs common opponents is an interesting bit of data, but I decided to scrap it in favor of readability and simplicity. This Google Sheet used formulas to compute everything on this screen from data in another workbook. Using this method on a Google sheet is not ideal since changing the dropdown on the sheet will change the information for everyone viewing the sheet.

Select a Competitor					Select a Competitor		
Coin			H2H		Aghi		
Current CoinRank 43			22-6		Current CoinRank 83		
Current Coins 6529					Current Coins 5561		
			Set Played At	Winner			
Opponent	Opponent's Coinrank	Record	Umeme 44	#N/A	Record	Opponent's Coinrank	Opponent
248		3-0	SWEET 25 Wave B3	#N/A	1-3		248
			SWEET 25 Wave B3	#N/A			
			SWEET 25 Wave B3	#N/A			
			Umeme 46	#N/A			
			Umeme 46	#N/A	1-0		4Frames
			Megga Smash Vol. 2	#N/A			
ADKnight	62	2-0	Megga Smash Vol. 2	#N/A	0-1	62	ADKnight
Aghi	83	22-6	Megga Smash Vol. 2		SAME	83	Aghi
Aiya		1-0	Megga Smash Vol. 2		1-0		Aiya
			Megga Smash Vol. 2				
			Umeme 47				

The screenshot below shows a mockup of how the UI would look to the user. Two searchable dropdowns that when are selected, display a record in the center box, and fill out a table with data beneath. This is not exactly how the final UI ended up, a key difference being an added submit button, but the underlying idea is still the same.

Player 1

10 - 21

Player 2

Date	Tourney Name	Winner

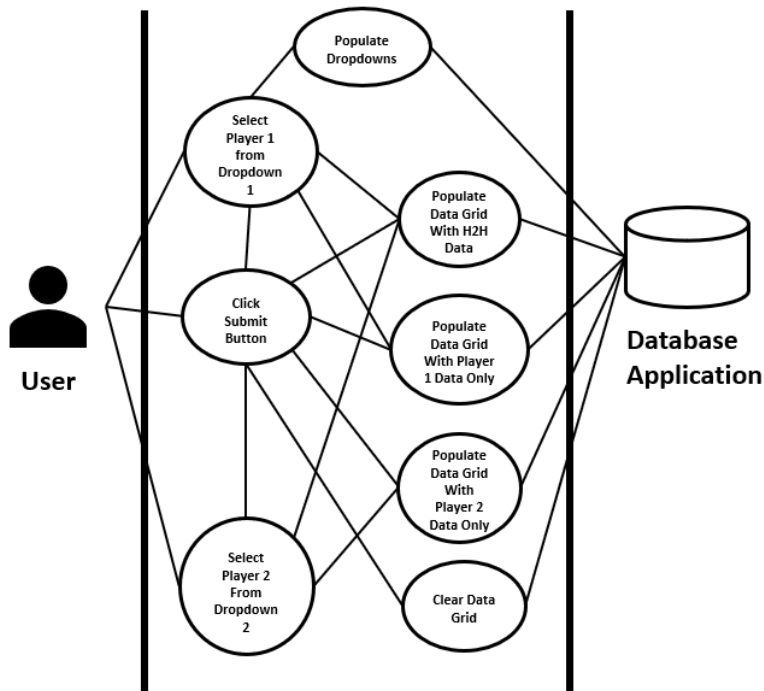
System Components and Functions (UML diagrams)

Database ERD:

I realize that my database is extremely simple, but I thought I would add it anyways. When importing this data into my database I decided not to split the table up into more than one table. Because my dropdowns require a unique list of names, I had thought about adding a table of player tags and tagIDs but seeing as this database needs to be updated regularly, it would just be much easier to append the data to the current table and run a query for the unique tags.

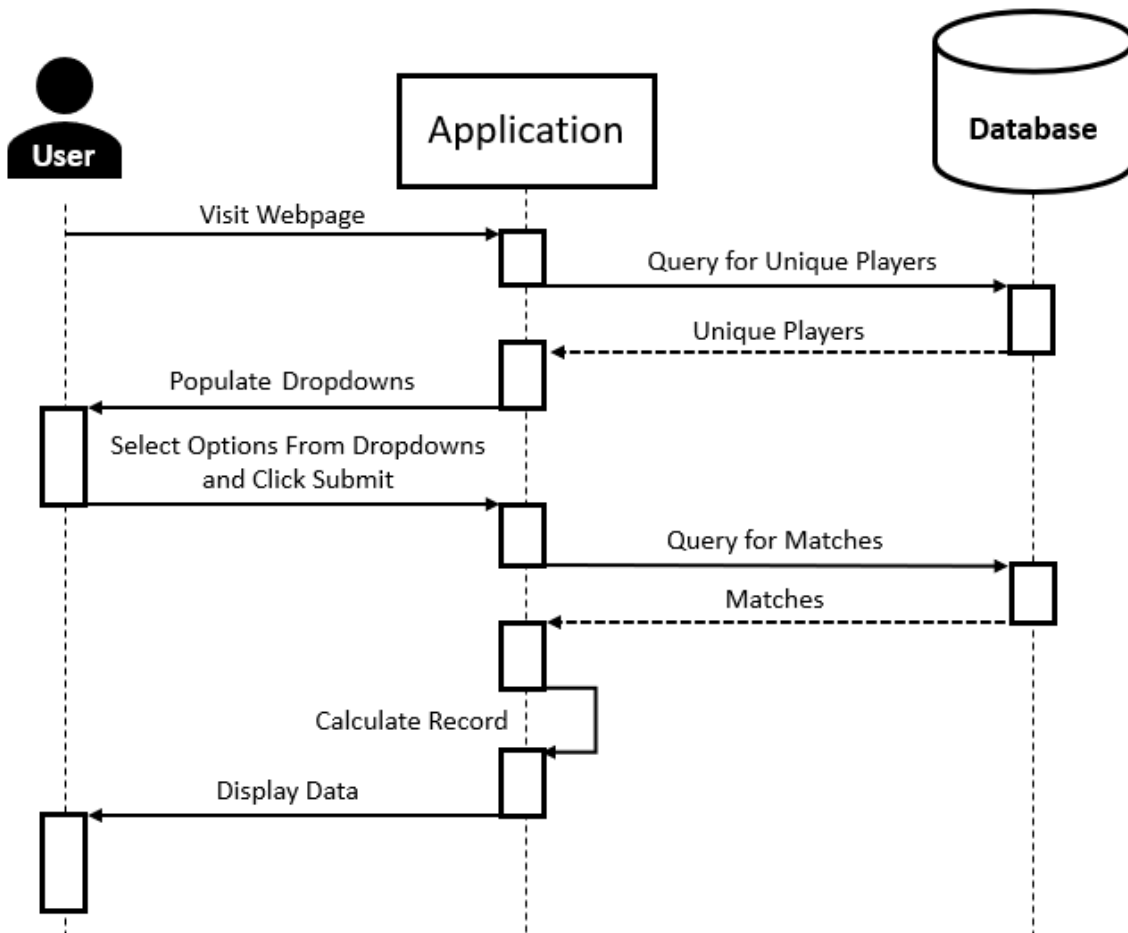
matches		
matchID	Int(11)	PKEY
matchDate	date	
tournament	varchar(25)	
winner	varchar(25)	
loser	varchar(25)	

Use Case Diagram:



This use-case diagram turned out more confusing than I'd hoped for. The one takeaway that I would want someone to have from this is that the user can clear the data grid by only clicking the submit button without selecting either player, the user can display data from only a single player by selecting one player from a dropdown and then clicking the submit button, and that the user can get head-to-head data by selecting players from both dropdowns and then clicking the submit button. The diagram is much more useful when you look at the outcome you want to achieve and then trace it back to what actions you need to complete to get that outcome.

Sequence Diagram:



This sequence diagram displays the order of interaction that occurs between the user, the web application, and the database.

Physical Implementations

Below is a screenshot of the finished page after a request has been made to view a head-to-head record. I designed the web application to query the database on index page load, for unique player tags in order to populate the select dropdowns. If the user selects one dropdown, and clicks the submit button, a separate query is run that selects all the data for one player and displays it in the grid. If the user selects both dropdowns, and clicks the submit button, then another query is run that gets head-to-head data and populates the grid with it. Only in this scenario is the record between the two players computed. This is calculated by a simple 'foreach' that counts how many times each name appears in the 'winners' column and then concatenates them and displays the info. Lastly if the submit button is clicked when

the 2 selects contain the default text 'Choose a Player', then the page is loaded from the beginning, giving the appearance of a cleared grid.

Michigan Melee Head to Head

Choose a Player

Submit

Choose a Player

Rich

3 - 0

Beckett

Tournament	Winner	Loser	Date
Here Be Mid Tiers 4 pool	Rich	Beckett	2018-11-17
WWE #57	Rich	Beckett	2018-11-01
Broken Joystick	Rich	Beckett	2018-10-21

Check out [Coinsheel](#) for that sweet Michigan Melee data!

2018 Rich Stretanski

Conclusion

What surprised me the most about this project is how much time I spent fiddling with CSS as opposed to everything else, which I had a pretty good idea how to do. The CSS was very tedious, especially when a styled object would only appear after a set of actions. What really saved me the most was finding the searchable dropdown jQuery plugin (which has a free use without limitation license that I included in the files). Without it I would have likely had to use a normal search which I didn't want to use for security reasons and developing it myself would have likely been what I would have spent all my time on (by the looks of it). I am glad that I was able to use this project to make something that could (and will likely) actually be used. There are some future improvements and additions that I would like to add that I felt were beyond the scope of this project. One thing I'd like to add is the common opponent's data from the Excel prototype that I made. The only reason I haven't already added it is because I couldn't really settle on how to best display this data. There are many ways it can be displayed: a separate table on the same page, a toggle between the two tables, or even another page altogether with a navigation bar at the top. Another improvement could be integration with the ranking system that is not managed by me. This

would require collaboration with people not in this class and thus, I decided it was not within the scope of the project. The third improvement is an obvious one, hosting. I originally wanted to self-host, however, was unsure of the side effects this would have on my own ecosystem (security, home bandwidth, personal machine resources, etc). One thing I decided, is that I want to pay for hosting eventually, but there is no reason to start now when the larger project (outside of this class) is unfinished. I look forward to improving this web application in the future.