Mariam Adegbindin

CIDM 6395-70/MS-CISBA Capstone

Professor: Dr Jeffery Babb

FLAG FOOTBALL PLAYER PERFORMANCE & INJURY TRACKING SYSTEM

INTRODUCTION

Casual flag football players often juggle busy schedules, informal pickup games, and limited access to performance science, yet they still want to stay safe and improve. My capstone project, the Flag Football Performance & Injury Analytics Platform, was born from my own experience playing flag football on campus and wishing I had a simple way to track my stats, monitor injuries, and learn from my habits.

In this paper, I will walk through how I applied and connected the four core areas of the MS-CISBA curriculum; Software Systems (SS), Business Analytics (BA), Data Management (DM), and Cybersecurity & Networking (CN) to build a working prototype that demonstrates real value for recreational athletes. I'll describe my approach, show the prototype components, highlight early insights, and reflect on how each discipline played a role in the final artifact.

CONTEXT

Most sports-tracking tools focus on mainstream sports (football, basketball) or high-end wearables. Few cater to informal leagues where players self-manage. A lightweight, player-centered analytics platform can fill that gap by combining easy data entry, clear visual feedback, and basic protections for personal health data.

SOFTWARE SYSTEMS

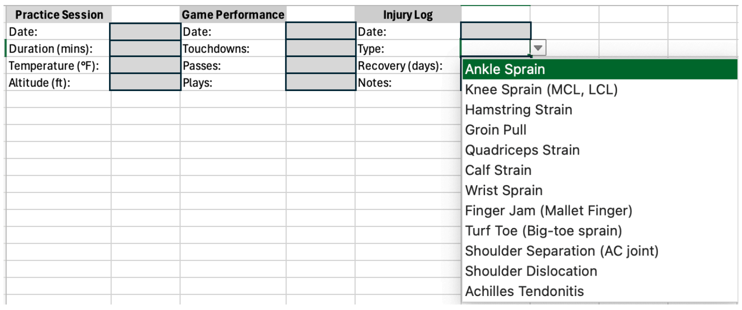
Software Systems is about designing and building technology that works for people. It's not just about writing code it's about creating a solution that makes sense, is easy to use, and solves a real problem. It also means thinking through how everything connects behind the scenes, from how the system is structured to how the user interacts with it on the front end. At its core, Software Systems is about functionality, usability, and good design working together.

For this project, I wanted to create a simple, intuitive system where players could enter performance stats and injury-related information without needing any technical experience. I started by outlining what the user experience should look like: who's using it, what kind of data they're entering, and how they'd want to see their stats. From there, I drafted mock-ups and planned out the structure of the system keeping things lightweight, especially since the users are everyday players, not professional analysts.

The system is being built using tools I've worked with before, like Excel for early mockups and potentially python for more advanced functionality. It includes input sections for logging practice sessions, game performance, and injury data, and it provides visual feedback through charts or tables. Even though this is a prototype, the foundation I'm laying out can easily grow into a mobile or web app with more interactive features down the line. This ties directly back to my work in the *Software Engineering* course, where I learned about design patterns, architectural styles, and the importance of thinking through how users interact with technology.

One project I remember especially helped me understand how to break systems down into components and map out their functionality step by step before writing any code.

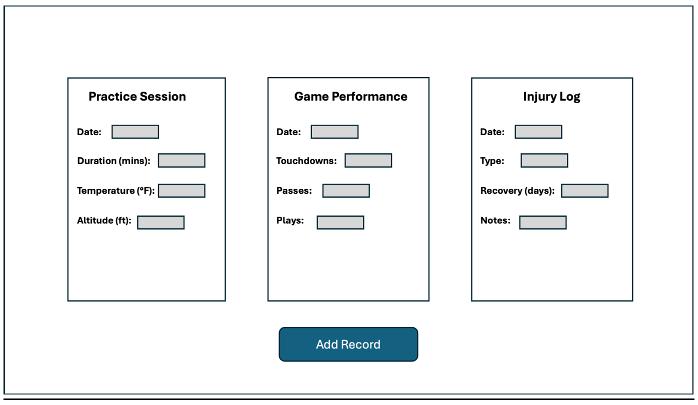
Methodology
User workflow design: I sketched an Excel mock-up showing three input panels. Practice Session, Game Performance, and Injury Log so no one ever feels lost.



Practice Session		Game Performance		Injury Log	
Date:		Date:		Date:	
Duration (mins):		Touchdowns:		Type:	
Temperature (°F):		Passes:		Recovery (days):	
Altitude (ft):		Plays:		Notes:	

Ankle Sprain
Knee Sprain (MCL, LCL)
Hamstring Strain
Groin Pull
Quadriceps Strain
Calf Strain
Wrist Sprain
Finger Jam (Mallet Finger)
Turf Toe (Big-toe sprain)
Shoulder Separation (AC joint)
Shoulder Dislocation
Achilles Tendonitis

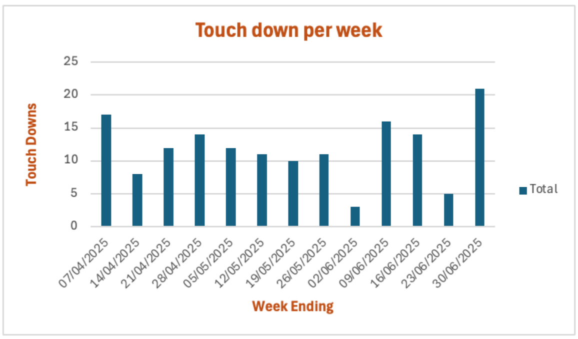
Wireframes: From those sketches, I generated a desktop mock-up and a companion mobile wireframe using simple shapes in PowerPoint.



Prototype code: To demonstrate the end-to-end flow of our system, I implemented a lightweight Python loader script called **ss_loader.py** that reads the CSV exported from the Excel mock-up and prints out each practice record.

<u>BUSINESS ANALYTICS</u>
In this project, analytics plays a central role. The goal isn't just to collect stats, but to help players understand how their habits, performance, and recovery are affecting their overall progress. Using the data logged like the number of games played, injuries sustained, or practice duration I apply basic analytics to highlight patterns that matter. For example, a player who practices more than two hours in high temperatures might show a higher risk of injury Or, someone recovering from an ankle sprain may need to ease back into full play to avoid re-injury. I used tools like Excel and Tableau to explore different ways of presenting these insights. Charts and dashboards make it easier for users to digest information at a glance. From identifying high-performing periods to flagging risky practice trends, these analytics help users make decisions that protect their health and improve their gameplay over time.

This directly connects to my experiences in the *Seminar in Data Analytics* and *Data Mining Methods* courses. I developed a solid foundation in tools like RapidMiner, Excel, and Tableau to clean, prepare, and visualize data. I also learned how to move beyond surface level metrics to uncover deeper insights that drive real value. In my earlier projects, I worked with customer behavior data and business trends this time, I'm applying those same skills in a personal and meaningful context through sports.



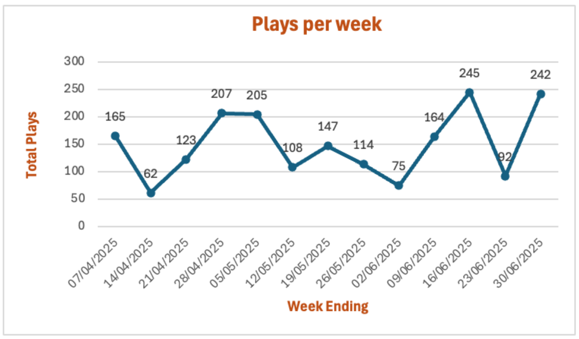


Figure 3A
Figure 3B

To turn my raw game-performance logs into a clear, action-oriented view, I first grouped every seven days of data into "weeks ending" in Excel. In **Figure 3A**, a clustered column chart shows total touchdowns scored each week—peaking at 21 in the week ending June 30, 2025, and dipping as low as 3 in early June. In **Figure 3B**, I plotted the total number of plays per week as a line graph, revealing that one of the lightest workload weeks (75 plays ending June 9) immediately preceded one of the highest touchdown weeks, suggesting periods of rest may boost scoring efficiency.

By viewing these two metrics side-by-side, players (or coaches) can quickly spot performance drivers and potential over- or under-training cycles. If heavy-workload weeks consistently coincide with slumps or injuries, training plans can be adjusted downward; conversely, if moderate weeks yield high touchdown totals, it may indicate an optimal balance of practice intensity and recovery. Together, these BA visuals demonstrate how simple weekly aggregation and charting transform raw CSVs into insights that guide smarter, data-driven decisions on practice schedules and game-day preparation.

DATA MANAGEMENT

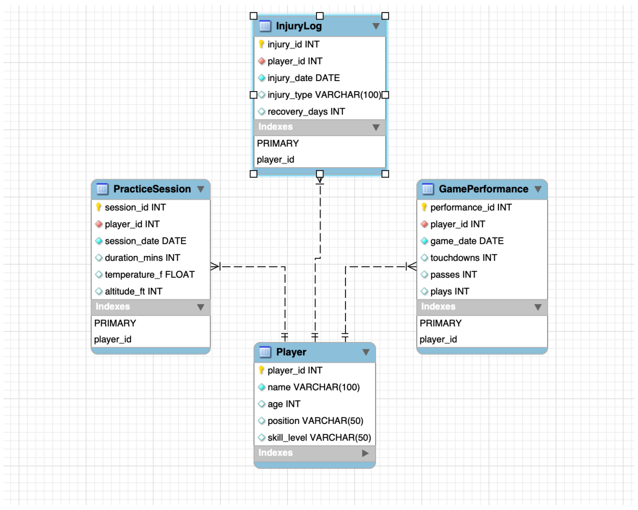
Data Management, to me, is all about structure and responsibility. It's not just about where you store your data it's about how well it's organized, how easy it is to retrieve, and how carefully it's protected. I treated each type of data game stats, practice logs, injury reports, and even environmental conditions as something that needs to be handled carefully. I started by deciding what fields were necessary, how they'd be formatted, and how the data would be stored in a way that made analysis easier down the line.

For this project, I modeled my data as a classic relational schema in SQLite. I started by sketching an **Entity–Relationship Diagram (ERD)** that captures the four key entities:

- **Player** (who's playing)
- **PracticeSession** (when they practice, environmental conditions)
- **GamePerformance** (scoring and play stats per game date)
- **InjuryLog** (what went wrong and how long recovery took)

Each of the latter three tables has a foreign key back to **Player**, ensuring that every session, performance record, and injury is always tied to a valid player.

All three transactional tables (PracticeSession, GamePerformance, InjuryLog) use a one-to-many relationship back to Player (each player can have many practices, performances, and injury records). By enforcing foreign-key constraints, the design guarantees referential integrity—every session, game stat, or injury must belong to a valid player.



This schema supports reliable, scalable data management. It keeps data consistent (no orphan records), organized (clearly defined tables and columns), and query-ready (e.g. joining through player_id to analyze how practice conditions or injury history affect game performance). As the prototype grows, new tables—like teams or equipment—can be slotted in with the same pattern, ensuring the model remains robust and maintainable.

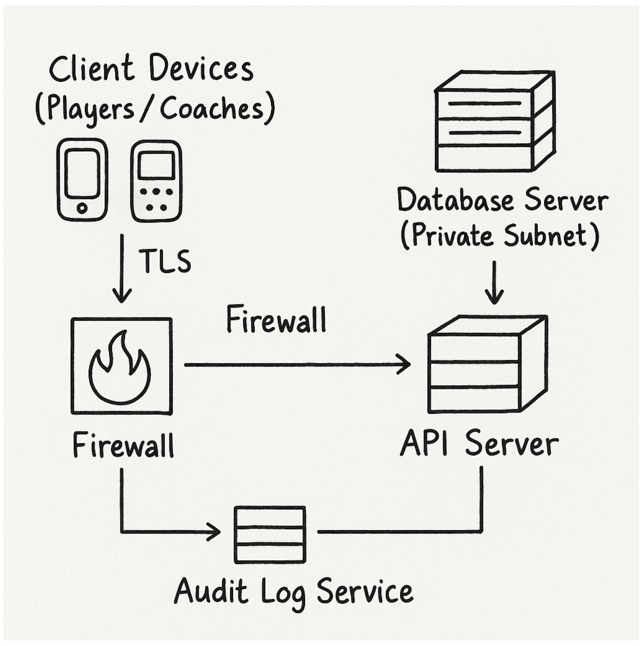
This part of the project ties directly to what I learned in *Data & Information Management*. That class taught me how to design ER diagrams, create relational schemas, and write SQL queries to manage data effectively. One of the key takeaways from that course was understanding the importance of planning your data model before you even think about building the system itself.

CYBERSECURITY

Even though this project is a prototype, I still made cybersecurity a priority. Injury and health-related data can be sensitive, so I included basic security practices in my system design. I planned for user access levels, for example, a player can view and input their own data, but only a coach or team leader might be able to view group analytics. I also considered how data would be stored securely, and how it would be protected if the project scaled into a real application.

I wrote a tiny Python service (flagfootball_demo.py) with decorator-based guards

Client Devices (Players/Coaches) connect over TLS to the API server. All traffic passes through a Firewall before reaching the API Server. The API Server resides in a separate DMZ from the Database Server, which lives in a private subnet. The API writes audit events to an Audit Log Service, ensuring every access is recorded for future analysis.



To demonstrate my understanding of security risks, I documented a simulated breach scenario something like a user accidentally gaining access to another player's injury log and explained how that kind of issue could be prevented with proper role-based access, data encryption, and system audit logs. Even simple steps like protecting the data entry sheet or locking formulas in Excel show how cybersecurity thinking can be applied at all levels of system design

My courses in *Cybersecurity* and *Network Management* helped me understand not only the technical tools like Wireshark and Nessus, but also the importance of policies, risk planning, and system audits. I learned how to identify vulnerabilities, plan for contingencies, and prioritize security even during the early stages of system development. In past projects, I practiced setting up network scans and writing incident response plan; skills that directly shaped how I thought about this project from a security point of view.

CONCLUSION

This Flag Football Performance & Injury Analytics Platform brings together my MS-CISBA toolkit into one prototype. I used Software Systems methods to map user workflows, build Excel and Python demos, and plan a scalable architecture. I applied Business Analytics to turn raw game and practice logs into weekly touchdown and play-volume charts that uncover training insights. A clean Data Management schema enforces integrity and readies the system for future growth, while Cybersecurity & Networking practices role-based access, encryption, segmented network design, and audit logging ensure that personal health data stays protected.

Each discipline reinforces the others: secure, well-modelled data fuels reliable analytics, which in turn guides system design and interface decisions. Although still a prototype, this project proves you can blend software engineering, analytics, data governance, and security into a cohesive tool that helps recreational athletes train smarter and stay safe. A fully realized version could become a mobile app with real-time alerts and coach dashboards but even today, the platform demonstrates the power of a true interdisciplinary synthesis.