CS495 Software Project Proposal: SAUCE: Real-Time Hockey Analytics and Coaching Dashboard

Stuart Belvin & Fabrizio Guzzo

2026-02-11

1 Client Information

By sharing this client information and the rest of this document, you are stating that this client has provided this project as something they want (not something you created and asked if they wanted), and that they are interested in having you complete this project for your capstone.

1. Client name: Matthew Barrow
2. Client title: Energy Consultant / Hockey Coach & Analytics Lead
3. Client email address: matthewabarrow@gmail.com
4. Client employer: Switch (Tech–Energy Sector), coaches the Loyola Maryland Hockey team
5. How you know the client: Two of Fabrizio's roommates are the captains of the Loyola Maryland hockey team. They put us in contact with their coach since they knew he was in need of statistical software.

2 Project Description

2.1 Overview

The goal of this project is to design and prototype a software platform called **SAUCE**, a real-time hockey analytics and coaching tool intended primarily for the Loyola Hockey team and also for other club, junior, and university hockey programs. Many teams currently rely on handwritten notes or disconnected spreadsheets to track key statistics such as hits, face-offs, shots on goal, and time of possession. This process is slow, error-prone, and fails to provide coaches with the instant insight they need to make in-game decisions or evaluate long-term trends.

The client has identified major gaps in the existing analytics process: the team lacks centralized tools, visual dashboards, and streamlined ways for multiple assistants to log data collaboratively during games. The purpose of SAUCE is to replace the current error-prone system built inside Google Sheets with a dedicated, multi-user platform that enables real-time data entry, clean visualizations, and trend analysis over time.

The client wants this project because there is a clear need among club hockey programs—over 500 teams nationwide currently lack accessible analytics tools. Existing commercial solutions (e.g., Mad Puck, ITrack Hockey) do not provide effective visualizations or are too complex. SAUCE aims to be a simpler, cleaner, more intuitive system tailored to the core statistics coaches rely on.

2.2 Key Features

At this point you should have a basic understanding of your client's needs. List out the key features of the software system the client wants you to build at a level that makes it possible for the department to determine if we can approve it.

- Multi-user live data entry (hits, shots, possession time, face-offs, etc.) during games

- User login system to assign the tracking of specific statistics to individual users
- Real-time dashboard for coaches showing ongoing game metrics
- Data visualization tools (graphs, possession charts, trends over time)
- Ability to review historical games and track long-term player/team performance
- Simple, intuitive UI designed to be used quickly during high-speed gameplay
- Exportable reports for coaches to use in reviewing game recordings or practicing planning
- Cloud-based storage to keep team data organized and accessible

2.3 Why this Project is Interesting

Why did you decide this project was interesting enough to you to be a capstone project? What about this project is enticing? Why should anyone care?

This project is interesting because it blends sports, technology, and real-time data analytics to provide a real, tangible benefit to the client. It presents a real-world challenge: designing software to be used in fast-paced, high-stress situations where usability and speed matter. The opportunity to create a tool that could serve hundreds of hockey programs outside of Loyola makes the project meaningful beyond the classroom.

Additionally, the client is genuinely invested in the solution and sees potential for future expansion and possibly even funding (venture capital or philanthropic). Building SAUCE gives us the chance to work on an innovative project with tangible impact, real users, and long-term value.

2.4 Areas of CS required

What subfields of computer science seem most likely to be relevant to your project? A capstone must involve multiple.

This capstone will involve several areas of computer science, including:

- **Software Engineering:** Requirements gathering, architecture design, iterative development
- **Web Development:** Multi-user front-end interface and backend system
- **Databases:** Managing game statistics, user accounts, and historical data
- **Data Visualization:** Graphs, charts, and dashboards for coaches
- **Human-Computer Interaction (HCI):** Designing a clean, usable UI that works during live sports events
- **Cloud Computing:** Hosting, authentication, and data storage

2.5 Potential Concerns and Questions

Is there any aspect of this project that makes you unsure if it will work, either due to your own interests/background, or that you aren't sure if it fits the requirements? Are there questions you have about this project that you want instructor feedback about?

There are a few uncertainties at this stage. First, the scope may need to be cut back to fit the timeline of a semester-long capstone. The full product vision is larger than what can realistically be built in one term. We may need guidance on scaling down to a feasible MVP (minimum viable product).

2.6 Summary of Efforts to Find a Project

(Not necessary for 482) [Briefly list out when/how you've discussed with this client, and if you've discussed with other clients who either didn't work out or didn't respond. If you considered a different project and it didn't work out, why didn't it work out?]

2.7 Comparison to Draft

This project is the same one that Fabrizio proposed. Our proposal is essentially the same as the draft. We were lucky that our client has a very clear idea of what he wants, especially since he has years of experience with other statistical tracking applications. This streamlines the process of gathering our project requirements and will make subsequent communication with the client clear and efficient.

3 Requirements

3.1 Non-Functional Requirements

[Non-functional requirements are just as important as functional requirements. Dont forget to specify them.]

Table 1 presents the NFRs for this software project.

ID	NFR Title	Category	Description
NFR1	Customizable Team Branding	Usability	The system shall provide at least 3 pre-designed color schemes that the client can select from to customize their team's dashboard interface.
NFR2	Rapid Navigation	Usability	All core application functions shall be accessible within 3 clicks or taps from the main dashboard to ensure efficient workflow during live games.
NFR3	Real-Time Performance	Performance	The system shall maintain data synchronization latency under 5 seconds during live game tracking for all users.

Table 1: Non-Functional Requirements

3.2 Functional Requirements (User Stories)

[In CS482, all functional requirements are written as User Stories. In CS496, some projects may use a different template to write the requirements. The table below is an example of writing the Stories. Adapt accordingly to different templates or if you want to display more info.]

Here's the revised table with properly formatted descriptions that fit within the page width:

ID	Story Title	Points	Description
U0	User Login	2	As a User, I want to login to the system so I can access and exit the application securely.
U1	User Profile	3	As a User, I want to create and edit my profile so my account is identifiable and I can contribute to my team.
U2	Landing Page	1	As a User, I want a landing page when entering the app to be greeted and directed appropriately.
U3	About Page	1	As a User, I want an about page to understand the app's purpose and functionality.
U4	Password Reset	2	As a User, I want to securely reset my password to avoid creating a new account if I forget it.
U5	Contact/Feedback	2	As a User, I want to provide feedback to admins so issues can be identified and fixed.
C0	Overall Game View	3	As a Coach, I want a simplified view of all game stats to make quick, game-winning decisions.
C2	Create Game	2	As a Coach, I want to create a game to establish a place for data collection and tracking.

C3	Assign Team Members	2	As a Coach, I want to assign stats to specific team members for more accurate data collection.
C4	Edit/Delete Game	3	As a Coach, I want to edit or delete game data to correct or remove inaccurate information.
C5	Team CRUD	8	As a Coach, I want to manage team data to prevent unauthorized users from editing my team's information.
C6	Data Export	3	As a Coach, I want to export game data in various formats for post-game analysis and review.
C7	Overall Data View	3	As a Coach, I want graphs of historical stats by opponent to identify trends in team performance.
C8	At-a-Glance Summary	2	As a Coach, I want a quick data summary to get an overview without analyzing detailed graphs.
C9	Player Specific Stats	2	As a Coach, I want to see individual player stats to make informed player assignment decisions.
C10	Individual Stat Graph	2	As a Coach, I want graphs of specific game stats to understand detailed aspects of gameplay.
C11	Period-Specific View	2	As a Coach, I want stats for specific game periods to see how my decisions affect gameplay over time.
C12	View Settings	2	As a Coach, I want to customize the stat view page to display only the statistics I need.
C13	Historical Opponent View	3	As a Coach, I want historical stats against specific opponents to prepare for future matchups.
C14	Historical Player View	3	As a Coach, I want historical player stats to provide targeted feedback for player improvement.
C15	Historical Stat View	3	As a Coach, I want specific stat trends over time to track team progress and development.
TM0	Historical Data View	2	As a Coach, I want to view historical game data to identify long-term team trends.
TM1	Notifications	3	As a Team Member, I want notifications when assigned to track stats to remember my responsibilities.
TM2	Multiple Tracking	3	As a Team Member, I want to track multiple stats simultaneously to maintain data collection with limited personnel.
TM3	Account Verification	3	As a Coach, I want to verify accounts before assigning stats to ensure only authorized users edit data.
DE1	Data Entry	2	As a User, I want to edit game stats to contribute to my team's decision-making process.
	Total:	73	

Table 2: Functional requirements as User Stories.

4 System Design

4.1 Architecture

Our team will be following a simple layered web MVC model for this project. Our main modules will be user authentication, game management, real-time data entry, and the live graph view of game data.

4.2 Diagrams

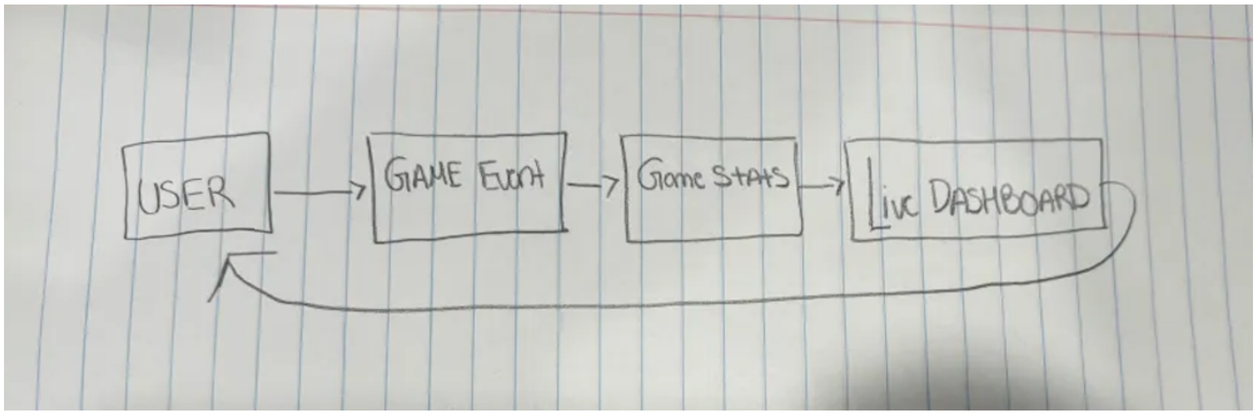


Figure 1: Simplified Class Diagram

4.3 Technology

For this project we will be using the MERN stack with typescript, mongodb, express.js, react, and node.js. For our testing framework we will be using jest and node.js. We will also be using ReCharts for our graphing components.

4.4 Coding Standards

- Limit the use of global variables.
- Use **CamelCase** for variables and function names.
- Final code should not be pushed unless it has been tested beforehand or explicit notice of an issue has been given.
- Comments must be written to help the entire team understand the code.

Header Format for Each Module Each module should begin with the following header information:

- Name
- Date
- Author
- Synopsis
- Variables

4.5 Data

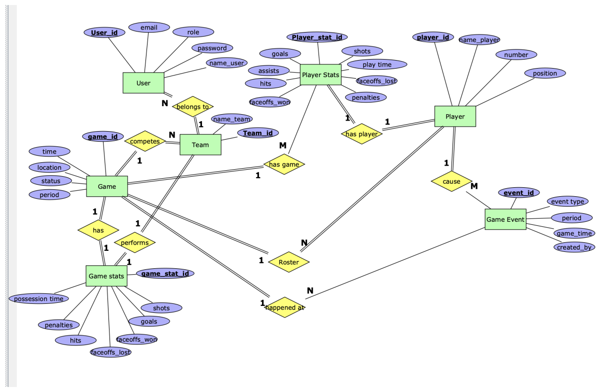


Figure 2: ERD

4.6 UI Mocks

[Define and draw/sketch/code the main UIs your user will interact with in your software. Add your UI mocks here and a short caption about it. Do not forget about the main forms and CRUD UIs.]

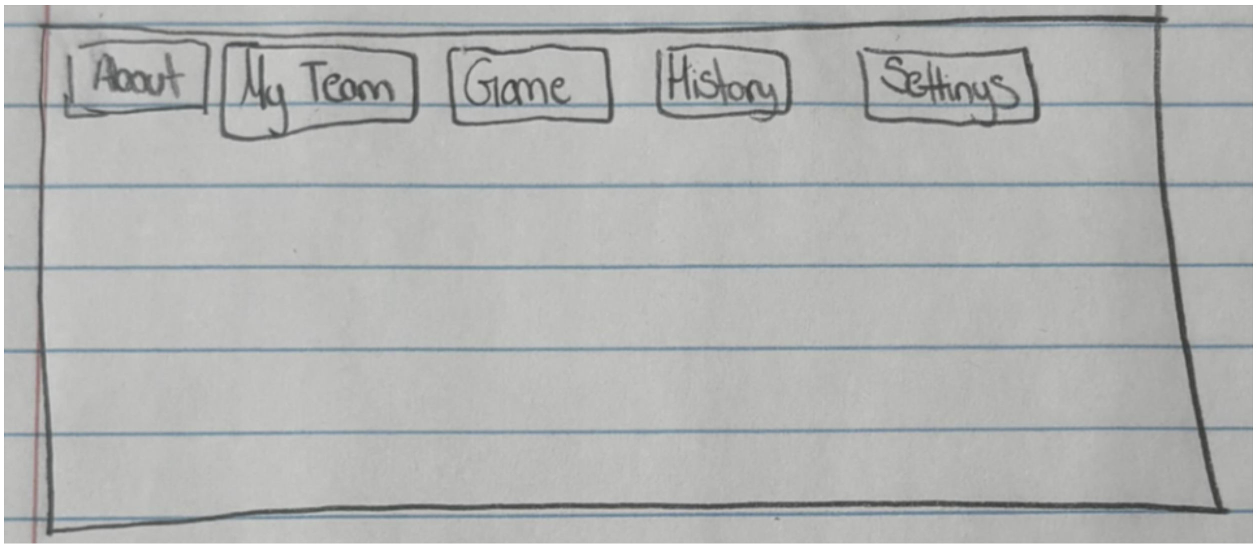


Figure 3: MOCKMENUBAR-TOP MENU STRUCTURE

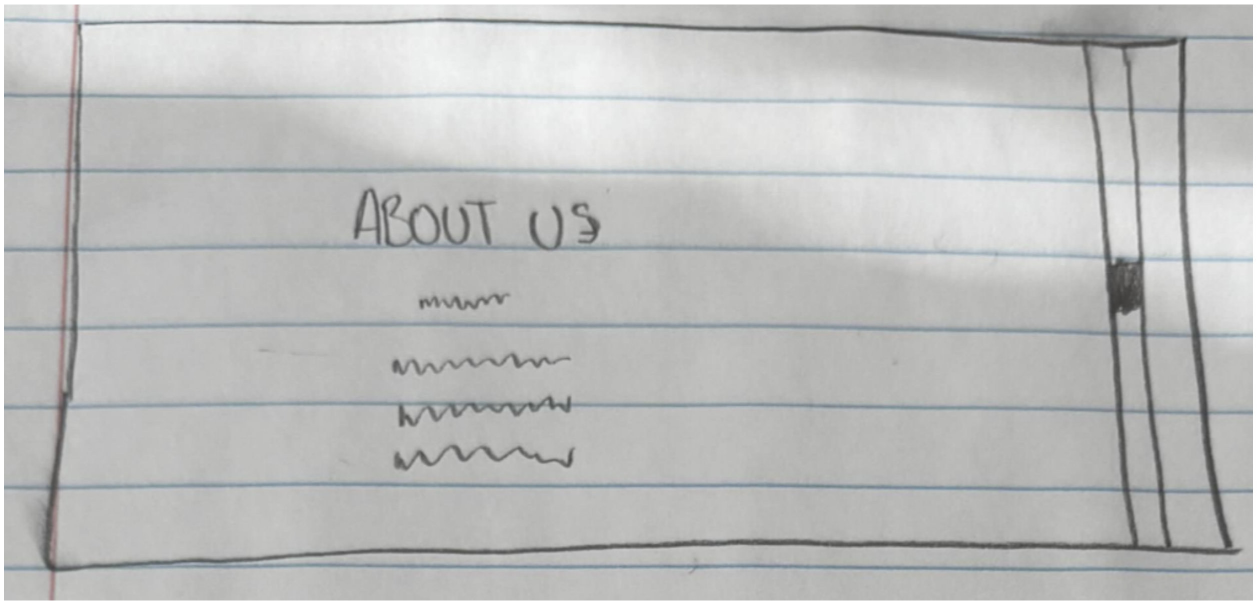


Figure 4: MOCKLANDING-LANDING PAGE

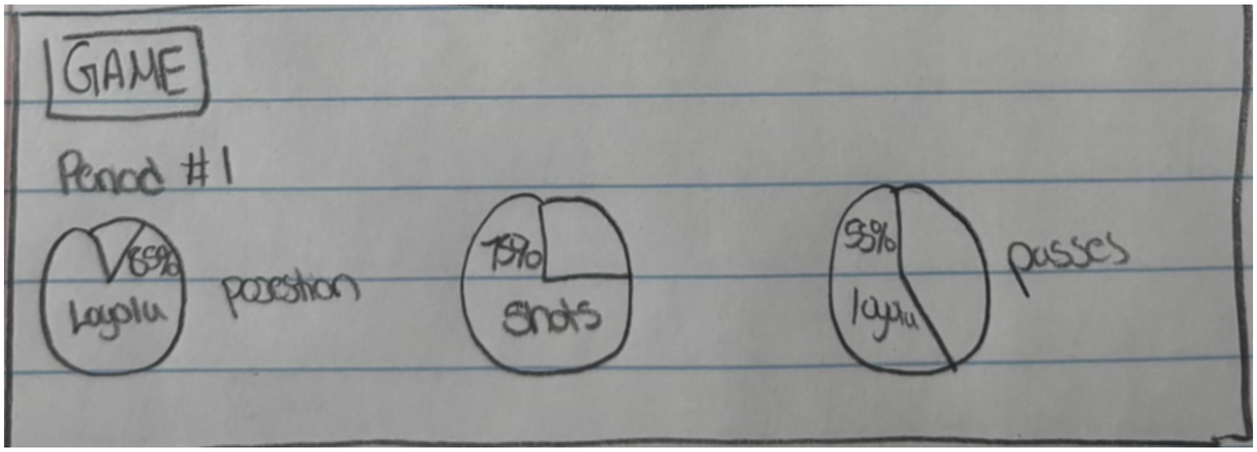


Figure 5: MOCKGAMEPAGE

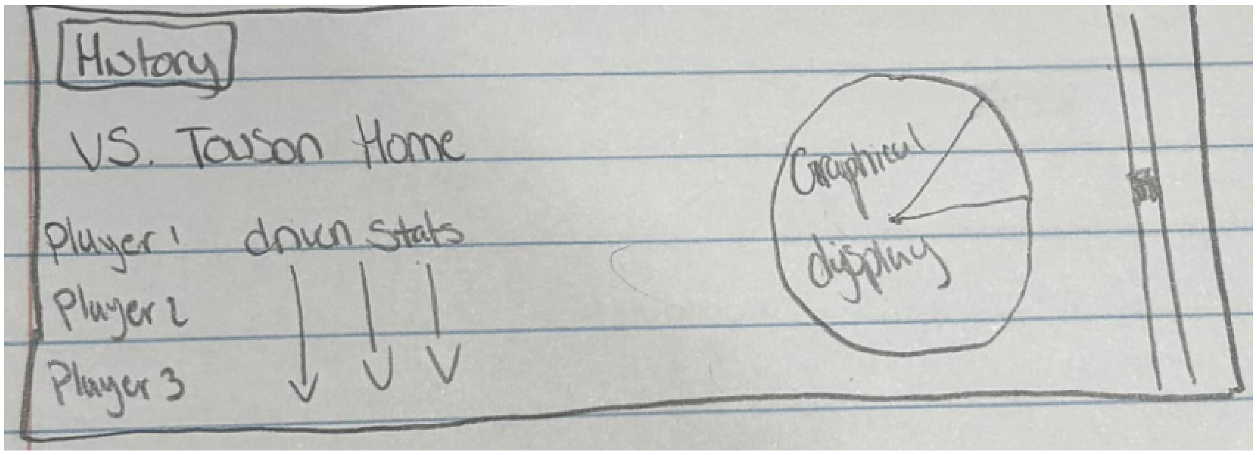


Figure 6: MOCKGAMESPEC-Specific Games

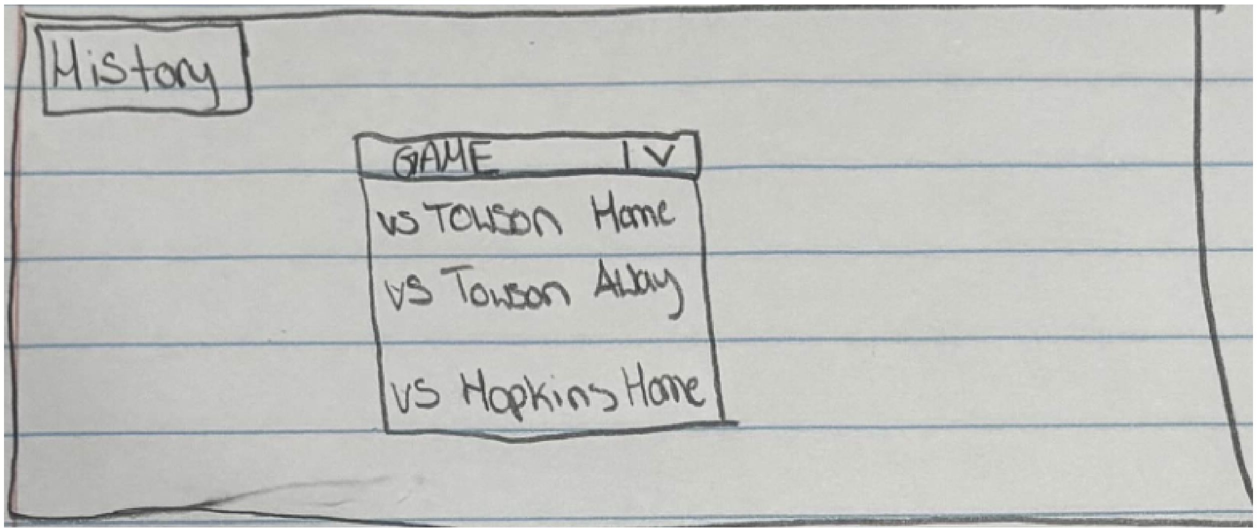


Figure 7: MOCKUIGAMEHIST-Game HistoryPage

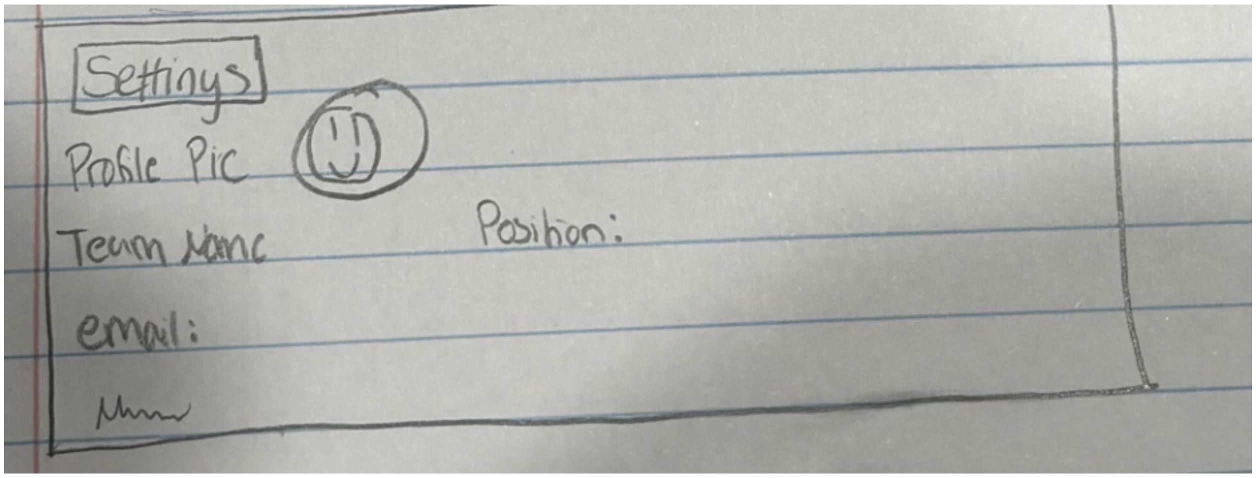


Figure 8: MOCKUIPROFILE-Setting Page

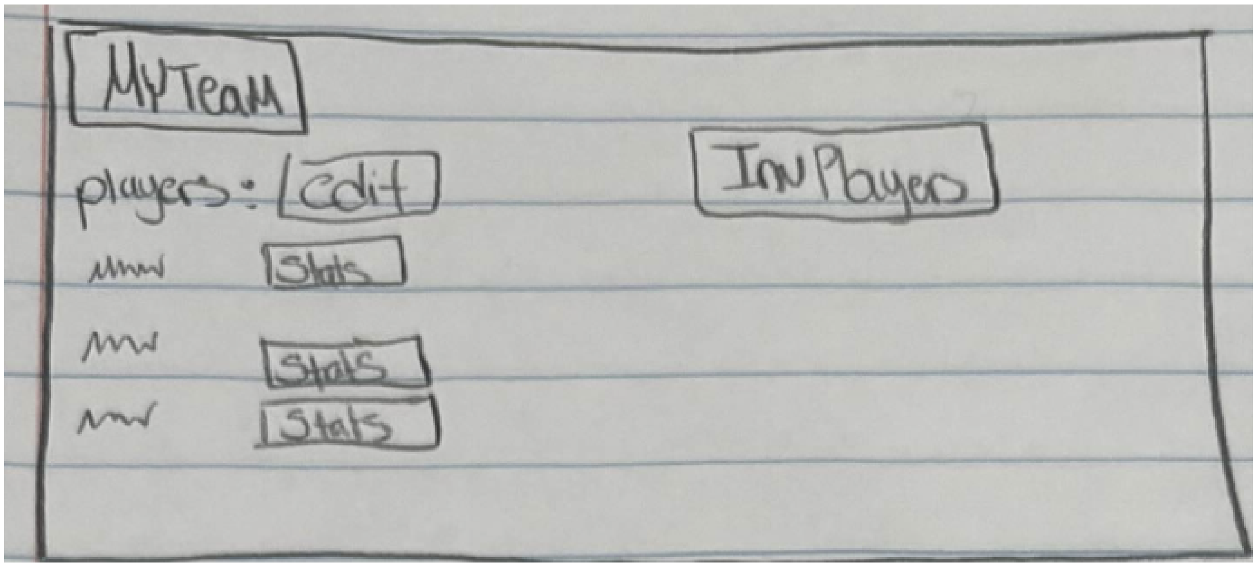


Figure 9: MOCKUITEAMPAGE-Coachs View Of Roster

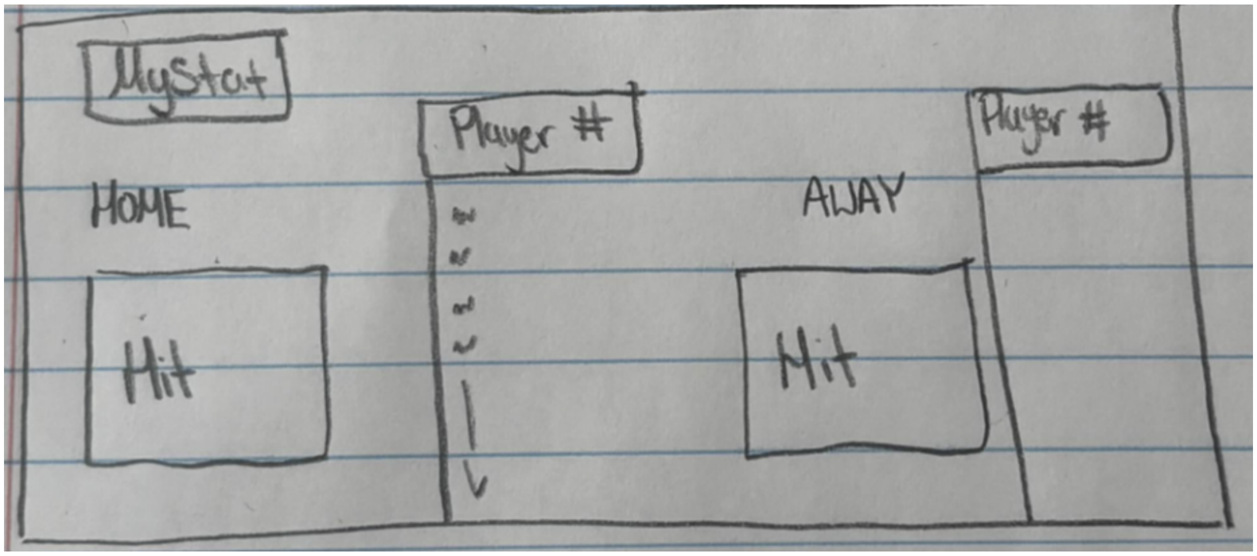


Figure 10: MOCKUISTATINPUT- Player/User View to Input Data

5 Iterations

5.1 Iteration Planning

[In CS496, you plan all iterations beforehand. In CS482, you update the planning here at each iteration.]

Table 3 shows the iteration planning.

			Points	
It.	**Dates**	**Stories**	**Planned**	**Done**
1	02/05 - 02/10	DE1 Data Entry, C2 Create Game, C4 Edit/Delete Game, U2 Landing Page, U5 Contact Us	09	09
		Total:	09	09

Table 3: Iteration Planning for Incremental Deliveries

5.2 Iteration/Sprint 1

5.2.1 Planning

The stories that we planned for this iteration were DE1, C2, C4, U2, and U5. We decided to focus on some basic pages and the basic data operations to start with as our foundation. This totals out to 9 story points, and we completed all 9. We decided on these story points since we wanted to have a webpage to present as soon as possible, and we wanted to get our back end database working as soon as possible with a basic ui for testing.

5.2.2 Work Done

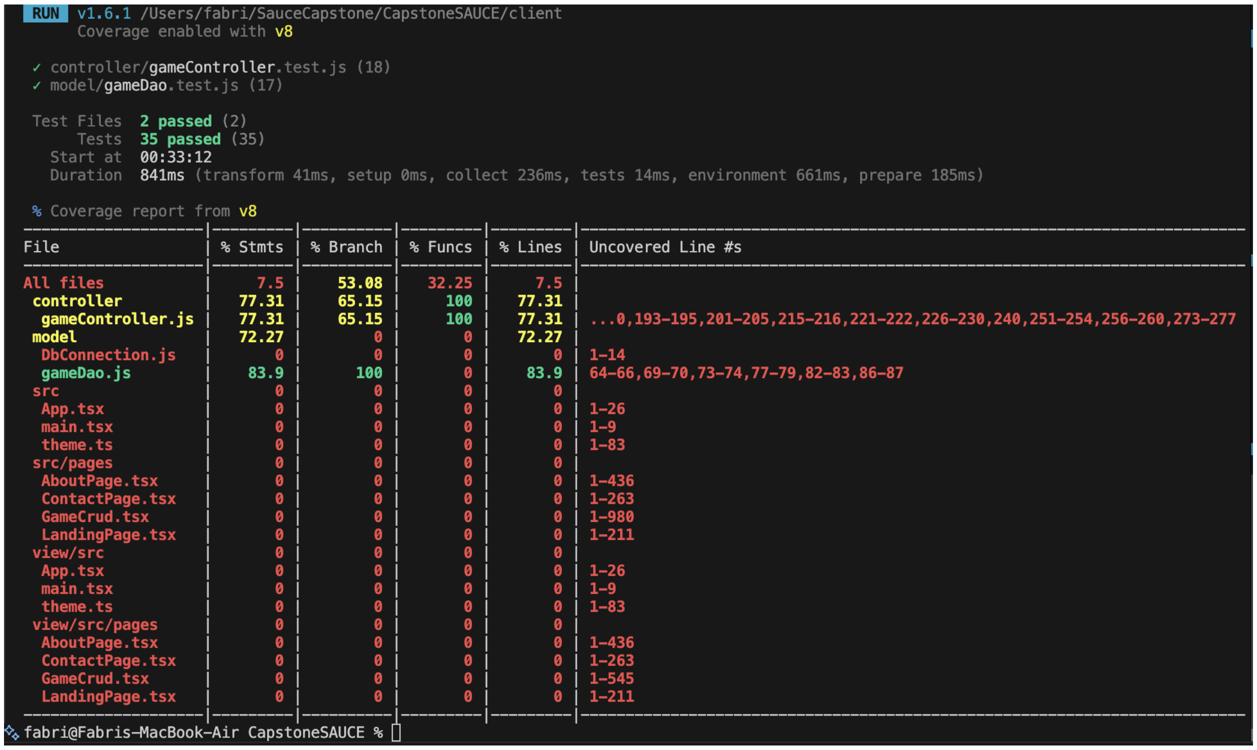
For the first iteration, we successfully established our MongoDB database and implemented React into our front-end architecture. We created both client and server components to better handle user interactions and feedback. In previous projects, running only a client side application was sufficient; however, for this project, we decided to implement feedback using a dedicated server instance that saves to a .txt file instead of relying on email based communication. This allowed us to structure the application more professionally using a true full stack approach.

We also established a concrete foundation for our models and controllers, following the MVC design pattern to ensure scalability and maintainability. Our GitHub repository was properly organized, and our Kanban board was updated to reflect progress throughout the sprint. All planning goals for this iteration were accomplished, including the development of the Landing Page, About Us Page, the Create Game user story, and the Edit/Delete Game user story. We also implemented the initial data entry point within the Create Game page (which will be refined in the next iteration). Finally, we completed backend testing for the Create Game CRUD functionality to ensure proper database integration and endpoint behavior.

5.2.3 Testing Coverage

For this this first iteration our main focus was creating a solid foundation to help future iteration development. Therefore our testing was less than ideal in terms of full coverage, but that being said we plan increasing coverage throughout our next iteration and devoting more time to testing to ensure a better report.

Figure 11 shows the test coverage



```
 RUN  v1.6.1 /Users/fabri/SauceCapstone/CapstoneSAUCE/client
      Coverage enabled with v8

 ✓ controller/gameController.test.js (18)
 ✓ model/gameDao.test.js (17)

 Test Files  2 passed (2)
      Tests  35 passed (35)
   Start at  00:33:12
   Duration  841ms (transform 41ms, setup 0ms, collect 236ms, tests 14ms, environment 661ms, prepare 185ms)

 % Coverage report from v8
---------------------|---------|----------|---------|---------|-------------------------------------------------------------------------
File                 | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
---------------------|---------|----------|---------|---------|-------------------------------------------------------------------------
All files            |     7.5 |    53.08 |   32.25 |     7.5 |
 controller          |   77.31 |    65.15 |     100 |   77.31 |
  gameController.js  |   77.31 |    65.15 |     100 |   77.31 | ...0,193-195,201-205,215-216,221-222,226-230,240,251-254,256-260,273-277
 model               |   72.27 |        0 |       0 |   72.27 |
  DbConnection.js    |       0 |        0 |       0 |       0 | 1-14
  gameDao.js         |    83.9 |      100 |       0 |    83.9 | 64-66,69-70,73-74,77-79,82-83,86-87
 src                 |       0 |        0 |       0 |       0 |
  App.tsx            |       0 |        0 |       0 |       0 | 1-26
  main.tsx           |       0 |        0 |       0 |       0 | 1-9
  theme.ts           |       0 |        0 |       0 |       0 | 1-83
 src/pages           |       0 |        0 |       0 |       0 |
  AboutPage.tsx      |       0 |        0 |       0 |       0 | 1-436
  ContactPage.tsx    |       0 |        0 |       0 |       0 | 1-263
  GameCrud.tsx       |       0 |        0 |       0 |       0 | 1-980
  LandingPage.tsx    |       0 |        0 |       0 |       0 | 1-211
 view/src            |       0 |        0 |       0 |       0 |
  App.tsx            |       0 |        0 |       0 |       0 | 1-26
  main.tsx           |       0 |        0 |       0 |       0 | 1-9
  theme.ts           |       0 |        0 |       0 |       0 | 1-83
 view/src/pages      |       0 |        0 |       0 |       0 |
  AboutPage.tsx      |       0 |        0 |       0 |       0 | 1-436
  ContactPage.tsx    |       0 |        0 |       0 |       0 | 1-263
  GameCrud.tsx       |       0 |        0 |       0 |       0 | 1-545
  LandingPage.tsx    |       0 |        0 |       0 |       0 | 1-211
---------------------|---------|----------|---------|---------|-------------------------------------------------------------------------
fabri@Fabris-MacBook-Air CapstoneSAUCE %
```

Figure 11: Iteration 1 test coverage report

5.2.4 Retroespective & Reflection

The biggest pitfall we had during this iteration was resolving unstaged changes in git. Since we were working independently over video call, we would often end up with git conflicts and issues pulling or pushing because of unstaged changes, and when we were able to push and pull, we had to resolve merge conflicts. I think we will improve this process in the next iteration by pair programming more often, coordinating our work so we aren't doing similar tasks, and doing git stash before git pull.

5.3 Iteration/Sprint 2

5.3.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.3.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.3.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.3.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.4 Iteration/Sprint 3

5.4.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.4.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.4.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.4.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.5 Iteration/Sprint 4

[CS496 has 5 sprints. CS482 only has only 3 sprints (remove Iterations 4 and 5 from this doc if you are writing a doc for 482]

5.5.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.5.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.5.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.5.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.6 Iteration/Sprint 5

5.6.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.6.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.6.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.6.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

6 Final Remarks

6.1 Overall Progress

[Have you completed everything? If so, present evidence on how you brought value to your client, and the overall client satisfaction. Otherwise, estimate how much progress you done and how long it would take to finish this project. Be concrete about your progress, you know how many story points your software is, how many points you completed (this shows your progress). You also how many points your team delivers at each iteration, therefore you can estimate how many more iterations it would take to finish the leftover points (show the math).]

6.2 Project Reflection

[Your personal reflection on the project. What lessons did you learned. What would you have done differently? How can you do better work in future projects? You may write this as a team or per person (or both — if all your iterations were team reflections, then it would be better to write individual reflections here)]

Appendix

[Appendix section if needed]