

Effectiveness of Our Team Contract

Course: 431W CMPSC

Team Members: Quinn Butcher Viswa Rathnakumar

Team Leader: Quinn Butcher

Under each section of the Team Contract, there is an **EFFECTIVENESS** header that details the effectiveness of that section of the team contract. These are the terms of group conduct and cooperation that we **AGREE** on as a team.

Group Members' Behavioral Expectations

Name	Behavioral Expectation	Example
Viswa Rathnakumar	Punctuality	Communicate meetings and project deadlines within reasonable expectations. Reminders and check-ins on member's progress are expected.
Quinn Butcher	Communicative	I want to ensure that I communicate with my team in a manner that ensures no conflicts while still being responsive and clarifying any doubts. An example of this would be to update my team regularly with decisions I made and WHY I chose to go with that option!

EFFECTIVENESS:

We met frequently and discussed project setup and guidelines. We clarified any tech stack decisions and made sure the other partner understood when code changes were occurring. We made sure that the other partner was comfortable with changes to the code as well. We also ensured that there was a QA done on most of the more complex pulls. Another feature that we used heavily was git branches, as these allowed for a preview of the code without having to merge it directly into the main branch. The behavioral expectations of both punctuality and communication through these methods were very effective, and we adhered to them precisely.

Skill Strength Identification

Name	Skill Strength	Specify Activity
Viswa Rathnakumar	Workflow planning	Planning out the times and workflow charts to get systems running together within reasonable amounts of time
Quinn Butcher	User Interface Design	I have built many React apps with a variety of libraries. I am an expert in TypeScript, as seen through my internship experiences, and am very confident in creating a very sophisticated UI in a short period.

EFFECTIVENESS:

Quinn laid out the groundwork for the REST API design, including the **CREATE** table statements in the backend. He also worked on the front-end. Viswa worked on all the routes in the backend, including the data validation. Viswa worked on the manual and team contract as

well. This split was incredibly effective as it made development much faster. This allowed us to do more with our time, while also producing a project that had satisfied our high standards.

Participation: We agree to....

Share ideas and expectations for the design documents. Work on the project within the deadlines and over time, avoiding procrastination.

Communication: We agree to...

Often share updates on the project, with detailed summaries. Check back in with the contract to make sure that the contracted duties are fulfilled. We also agree to check in with the professor for any questions immediately.

Meetings: We agree to....

Hold regular meetings that are both productive, as well as concise in an effort to ensure that there is no wasted time. We agree to actively ensure that both group members are available to attend these meetings. If one is not able to attend meetings, we will reschedule the meeting to a time that works for both members.

EFFECTIVENESS:

We consistently held timely and well-organized meetings, ensuring that discussions were focused and productive. Throughout the project, we made a point to regularly check in on each other's progress, sharing updates and offering support as needed. We maintained clear and open communication, promptly informing one another of any developments or milestones achieved. Both of us contributed meaningfully to the design of both the front-end and back-end, ensuring that our collective input shaped what the final outcome was.

This collaborative approach helped us stay on track, meet project goals, and ensure that the final project was a perfect blend of the final project that we each envisioned separately.

Conduct: We agree to...

Follow in accordance with Penn State AI rules. Follow with the agreed contract, fulfilling duties. Any difficulties in following the contract will be discussed. Members will continue meeting, conflict resolutions, communicating, and participating to ensure that progress on the project moves successfully.

Conflict: We agree to...

We commit to openly discussing our differences and mediating any conversations that we have in the chance of encountering any disagreements. These discussions will be held with respect and understanding of both parties, guaranteeing that the point of view of both members is held in high regard. We plan to find the best solution and will maintain a continued constructive dialogue on continued disagreements.

Deadlines: We agree to...

Hold ourselves accountable to these strict deadlines in order to make sure that we don't create backlogs for our other teammates. We also want to adhere to these deadlines in order to create open and continued dialogues, such that we minimize any conflicts. These deadlines will be strictly enforced unless otherwise agreed upon by both members.

EFFECTIVENESS:

After each meeting, we developed a clear, cohesive plan to complete the work before the deadline. We were effective and worked together. Any conflicts we had in deciding the code we resolved very quickly. We have followed all code of conduct rules and ensured that all the

code that we have developed was written and created solely by ourselves, and only referenced publicly available documentation for many of these tools and libraries that were utilized to make an easier development cycle.

The timeline was the only aspect that experienced adjustment, as some initial setup issues led to a shift in the tasks to be completed. However, these issues were resolved promptly, and the deadlines were only extended by one week. Overall, this delay was a minor setback and did not significantly impact the project's progress.

Progress and Assessment

Project Tasks / Requirements	Who Is Completing This Activity / Task	Date to be Completed
Data Gathering	Viswa and Quinn	9/27
Data Trimming	Viswa and Quinn	9/27
Database design documents	Viswa and Quinn	10/11
Front end design documents	Viswa and Quinn	10/11
Database setup	Viswa and Quinn	10/18
SQL endpoint set up	Viswa	10/25
React css	Quinn	10/18
React javascript	Quinn	10/25
React endpoint set up	Quinn	10/25
React endpoint testing	Quinn	10/28
SQL Endpoint testing	Viswa	10/28
Any additional features planning	Viswa and Quinn	11/01

EFFECTIVENESS:

Quinn and Viswa both gathered data. Quinn and Viswa both did the design documents. Quinn did all the React frontend, and boilerplate code for the backend. Viswa did the SQL endpoints and tested them.

Both Viswa and Quinn planned the trades feature as well as the advanced stats feature. These deadlines were followed loosely with some of the frontend and backend setup taking slightly longer. This was discussed with both partners and agreed upon, thus shifting our schedule back slightly.

Reflection Overview

When we first started this project, it felt overwhelming. With so many moving parts and the use of technologies we weren't entirely familiar with, the challenge became even more daunting.

Initial Steps

At the outset, we leaned heavily on a reference document to break down the full-stack application into manageable pieces, which helped us gain a clearer understanding of what needed to be done. The first step was setting up an API and backend. We followed a guide on Medium, which enabled us to create a solution that worked well from the start.

Once the initial backend was in place and confirmed to be working using Postman/Insomnia (our API development and testing platforms), we moved on to the frontend. This part was more familiar to Quinn, and he was able to get much of it up and running quickly. As he worked, he focused on making key components as abstract as possible, knowing that many of the tables would be reused in different parts of the app. This is where we created `DefaultTable.tsx`, which became a versatile component used throughout the frontend. Getting this working was a major milestone, as it marked the integration of both the frontend and backend.

With everything set up, our final initial task consisted of importing our CSV data into the table. We found the necessary files in Artifact II, which contained the data needed for our database to host the information for the NBA 2023 season. A lot of the data required cleaning, particularly the `team_id`, which was inconsistently formatted across different CSV files (e.g., `PHO` or `PHX` for the Phoenix Suns). Once cleaned, we used Python's built-in CSV parser to make multiple `INSERT` statements, adding the data to the database.

Secondary Steps

The next phase involved adding more complex routes—anything beyond the basic routes like "get player by name," "get all players," and simple mutations. These were part of the initial setup, along with the search functionality. Viswa took the lead on this part, focusing on the multi-table join routes and any other routes supporting our app's features. This is also where we created a different 5 table join (our previous one did not work as expected for our frontend).

Meanwhile, Quinn continued working on the frontend, building out new features as each route was completed. Most of the frontend work was finished during this phase, with the exception of mutation-related features, which would be tackled in the final steps.

Final Steps

In the final phase, Viswa added the mutation routes, including the trade table, which we had not initially anticipated but ultimately proved to be a fascinating addition. The trade table introduced a new layer to our database and sparked ideas for future features. For instance, we plan to eventually support more advanced player stats, such as those related to player movement and strength of schedule (SOS), which are not yet part of our current system.

During this stage, Quinn also worked on data entry, leveraging the new routes with `useMutation` in Tanstack to integrate them into the frontend. This was also when we

conducted quality assurance (QA) for the frontend, ensuring that data validation worked correctly for all the new routes.

Final Thoughts

Building a full-stack application from scratch is a challenging process, made even more complex when multiple opinions are involved in solving each problem. One thing that worked particularly well for us was bringing up relevant documentation during meetings to justify our choices and ensure we were on the same page about the best approach. This experience helped us navigate proper time management with a complex project. Furthermore, we learned more about cooperation, as both front-end and back-end developers are dependent on each other's progress. Lastly, there are security issues that need to be addressed on both ends, and this project helped us gain a bit more insight on the nature of security and project development.

GitHub Link

<https://github.com/psu-db/cmppsc431w-project-quinn-and-viswa15983cdcc871f8ee3fd74492c7f448cdc684e10f>