**AI for Auburn University Baseball - Database**

Cycle 3 Report

Authors:

Teddy Cooke

Laura Cromwell

Frankie Donze

Caden Garrett

Micah Key

Braden Mosley

Jacob Munroe

Annie Murphy

COMP 4710: Senior Design Team #9 Orange and Blue

Department of Computer Science and Software Engineering

Samuel Ginn College of Engineering, Auburn University

Table of Contents

[**System Metaphor - AM . 2**](#_k5jp0x4tuuyj)

[**Cycle Intent - TC . 2**](#_iejjrgxmasnc)

[**Design Documentation 2**](#_t7u5avqwjuju)

[Requirements - LC 2](#_3cm13w2vq6pk)

[Design Decisions - CG, BM, MK, JM 3](#_bjee153ar6yi)

[User Stories - All 11](#_qnxskp5ehnnm)

[**Management Plan - FD 17**](#_id988kb516bu)

[**Test Results - JM 18**](#_tzyckvbadi0n)

[**Lessons Learned / Challenges - LC, AM 20**](#_a3oy3mng0k41)

[**Memoranda - All 21**](#_qjl0l8y7985d)

[**Source Code - All 21**](#_h7k54des74tj)

[**Presentation Slides - All 22**](#_589ssvn0xspo)

[**Previous Cycle Documents - All 22**](#_mtiud8x4q5n4)

[**Sponsor Approval 23**](#_8lxafouwwh9z)

# 

# System Metaphor

Contributor: Annie Murphy

The baseball database system can be compared to that of a scoreboard. Similar to how a scoreboard at a baseball stadium keeps track of every play and outcome during a game, our database and UI serve as the ultimate source of information for coaches and players. It records every aspect of the game, from player statistics and team performance to the differing scores between the two teams playing. Just as fans rely on the scoreboard to stay updated on the game’s progress, coaches and players can rely on our database to access all relevant information. It provides a comprehensive view of the baseball game, comparing pitchers versus opposing batters, allowing coaches and players to make data-driven decisions and improve overall game performance.

# Cycle Intent

Contributor: Teddy Cooke

Our intent for this cycle was focused around integrating the models into our database and website. We realized that creating an API was unnecessary and decided to directly integrate the models into the database instead. We planned most of this cycle around getting the models into the database and pulling these from the database into the website. Additionally, as we are reaching the end of the semester and our time with the project, we are identifying our remaining goals and setting goals for the summer and fall semester teams. With this, we are ensuring that all of our work is well-documented and ready to pass on to a new group.

# Design Documentation

## Requirements

Contributor: Laura Cromwell

1. **Centralized database that is compatible with AI Modeling Senior Design Team**: Our sponsor has requested a database that houses all current game data, practice data, pitching data, etc. This data will then be used to display current statistical information, and will also be used by the AI Modeling Senior Design Team. This means we will need to make sure our database and our code is compatible with the models being made by other teams.
2. **System should be easy to update**: In order to keep the statistics and models correct and relevant, coaches/administrators should have the ability to easily submit a csv file after each game. From there the database and any subsequent information will be updated to reflect the information inputted.
3. **Database is easily accessible**: Our sponsor wants a platform that is available anywhere, at all times, especially during games. In order to achieve this, our database will be hosted with a web-based interface to allow login anywhere at any time.
4. **Ease of usability**: Our platform should be easy to use and understand, especially for coaches and players who are not “tech-savvy”.
5. **Login/authentication for users**: Due to the nature of the data being stored on the database, it is imperative that it is not available to just anyone. The data should stay secure and only accessible by the Baseball Team.
6. **Different system access for administrators, coaches, players, etc.**: Our sponsor has requested that there are different levels of information access available based on a person’s role within the team. Coaches should have full access to the system, and players should have a limited view of the information available. Exact permissions will be clarified at a later date.
7. **Responsive web design**: To ensure the accessibility of our system, it should be able to be accessed on any device, whether that is a laptop, iPad, or other device. For this reason the system should have a responsive web design to ensure readability across devices.
8. **Ability to filter data**: Our sponsor has requested the ability to filter the data available. This will help them to single out players, teams, pitches, etc and aid in their decision making when using the platform. This will also aid the readability of the data available.

## Design Decisions

Contributors: Caden Garrett, Braden Mosley, Micah Key

With the large scope of this project, and since our group is the first to take on this project, we have focused a lot this cycle on making smart design decisions that will ensure this project’s stability among future groups in this project’s lifetime. Our meeting with Brian Plexico, who works on data analysis for the Tampa Bay Rays, was very influential in our design decisions. He advised us to spend time thoroughly planning out our system and make intentional design decisions, as these decisions will impact all of our future work.

Architecture

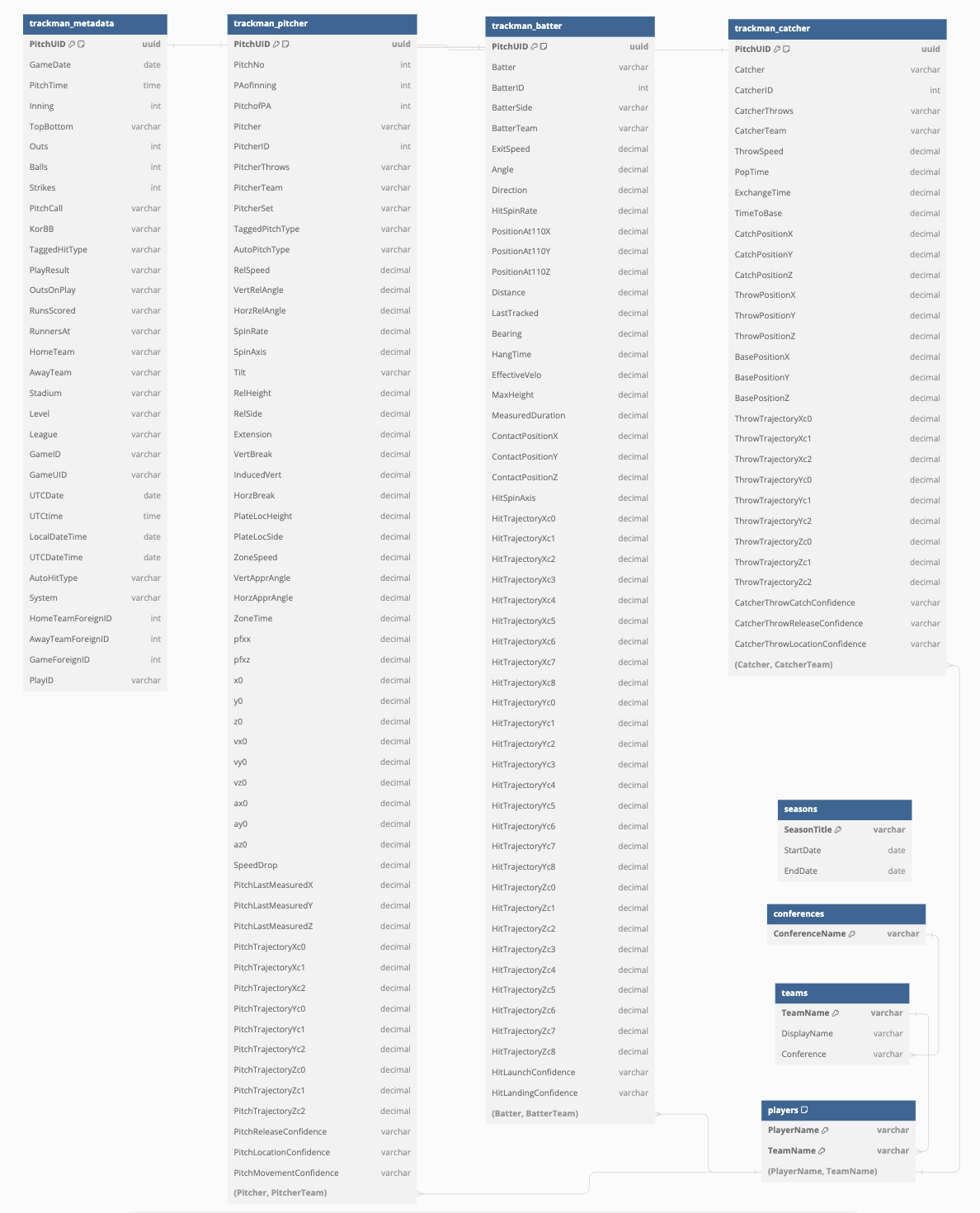
A high-level view of how our project will work follows: the user will visit the domain datagetta.app and will be presented with the landing page. On the landing page, the user will navigate to sign into our application The Eye and will be redirected to Auburn’s authentication system. Once logged in, the user will be on the dashboard of the app where they have a variety of options on how to use the app. One way the user can navigate the app is through the search bar. The user may search a team or a player and then be directed to their corresponding page. Another way a user could use the app is to navigate to the opponents page, click on the team they want to scout and be directed to the corresponding team page, and then click on the player from said team and be directed to the corresponding player page. The data will automatically be uploaded into the database from FileZilla, which holds all the Trackman CSVs, at a planned time of Monday mornings.

Back-end overview:

When it comes to the back-end portion of the project, we made the decision to use Postgresql as our database. The main reasons for choosing this is because Postgresql is a very popular database option among production databases and has many capabilities and performance optimizations that will be very beneficial to us. Another big reason is that Postgresql is a SQL database and will allow us to add indexes to the database to further speed up queries and calculations on our data. For our parsing options to get the data in the database, we have decided to use Python. This is mainly because of its popularity with most developers. It also has built in libraries for Postgres that will allow for interactions with it much easier now and in the future.

We now have a finalized schema for our database. Our schema includes the following tables: players, teams, seasons, trackman\_metadata, trackman\_pitcher, trackman\_batter, and trackman\_catcher. The players, teams, and seasons tables describe and connect player and team information to specific seasons. The trackman tables describe the trackman data collected from games, and that data is broken down into tables that focus specifically on metadata, pitchers, catchers, and batters. The calculations and statistics (described below) are described by views in SQL. These views are calculated from data described in tables in the schema. Our schema should not change from this point forward.

The schema is described on the following page:

Figure 1.0 Schema for the Database

Back-end: calculating statistics

In addition to a player’s name, team, and position, Coach Teaford has requested that we include the following statistics (below) as well. Our original plan was to scrape this data from TruMedia, but we encountered an obstacle trying to obtain access. After a discussion with Coach Teaford and Adam Nebel, we found that we will be able to calculate these numbers with Trackman data. Teams who participate in Trackman have access to all the other team’s data who also participate, which will enable us to store and calculate all necessary data. Statistics and definitions are defined by the [Fangraphs](https://library.fangraphs.com/) and [MLB](https://www.mlb.com/glossary) glossary web pages.

Statistics:

1. At Bats (AB)
   1. Number of trips to the plate in which the batter does not walk, get hit by a pitch, sacrifice (fly or bunt), or reach on interference.
2. Batting Average (AVG)
   1. Rate of hits per at bat, calculated as H/AB
   2. H: Number of hits
   3. AB: see above
3. On Base Percentage (OBP)
   1. Rate at which the batter reaches base, calculated as (H+BB+HBP)/(AB+BB+HBP+SF)
   2. H: Number of hits
   3. BB: Total number of walks (includes IBB)
   4. HBP: Number of times the batter reached after being hit by a pitch
   5. AB: see above
   6. SF: Number of times a batter’s fly out allowed a runner to tag up and score.
4. Slugging Percentage (SLG)
   1. Average number of total bases per at bat, calculated as Total Bases / AVG
   2. Total Bases: Total bases refer to the number of bases gained by a batter through his hits. A batter records one total base for a single, two total bases for a double, three total bases for a triple and four total bases for a home run.
   3. AVG: see above
5. On Base Plus Slugging Percentage (OPS)
   1. Combination of OBP and SLG, calculated as OBP+SLG
   2. OBP, SLG: see above
6. Isolated Power (ISO)
   1. Average number of extra bases per at bat, calculated several ways such as SLG minus AVG
   2. SLG - AVG (see above)
7. K Percentage (K)
   1. Frequency with which a pitcher strikes out hitters, calculated as Total Strikeouts / Batters Faced
   2. Total Strikeouts: total number of strikeouts for a pitcher
   3. Total Batters Faced: the total number of batters faced by a pitcher
8. Base On Ball Percentage (BB%)
   1. Calculated as Total Walks / Total Batters Faced
   2. Total Walks: total number of walks allowed by a pitcher
   3. Total Batters Faced: the total number of batters faced by a pitcher
9. In Zone Whiff Percentage
   1. Calculated as Misses in Zone / Swings in Zone
10. Chase Percentage (Chase %)
    1. Calculated as Total Number of Chases / Total Out of Zone Pitches
    2. Total Number of Chases: Number of times a batter swung at a pitch out of the zone and missed
    3. Total Out of Zone Pitches: Number of pitches thrown out of the zone by a pitcher
11. Total Pitches (NP)
    1. The total number of pitches thrown by a pitcher
    2. We are further breaking down this total into the total number of each pitch for the following types of pitches: curve balls, four seams, sinkers, sliders, two seams, and change-ups.
12. Games (G)
    1. The total number of games where a player makes an appearance at any point
13. Games Started (GS)
    1. The total number of times a pitcher starts a game. A game start for a pitcher is when he is the first pitcher to throw a pitch for his team.
14. Inning Pitched (IP)
    1. The number of innings a pitcher remains in a game. Each out in an inning represents one third of an inning
15. Hits (H)
    1. The total number of times a batter achieved one of the following: a single, double, triple, or home run.
16. Walks (BB)
    1. The total number of walks achieved by a batter
17. Strikeouts (SO)
    1. The total number of times a batter strikes out at the plate
18. Home Runs (HR)
    1. The total number of home runs achieved by a batter
19. Extra Base Hits (XBH)
    1. The total number of hits from a batter that is not a single. This includes doubles, triples, and home runs obtained by a pitcher.
20. Hit By Pitch (HBP)
    1. The total number of times a batter is struck by a pitch and did not swing at the ball
21. Plate Appearance (PA)
    1. The total number of completed turns batting from a player. A plate appearance takes into account every single time a batter appears at the plate and a result between the pitcher and batter is obtained.

The statistics are described as views in the database. The statistics are calculated using SQL and the data described in the tables in the schema. The first view is the pitch\_sums\_view, which has the total count of each pitch for each pitcher. The player\_stats\_view describes common statistics like At Bats, Batting Average, On Base Percentage, etc. The pitcher\_stats\_view describes common statistics related to pitchers, such as Total Strikeouts and Total Walks.

Back-end: inputting data from Trackman using a Parser

All of the data in the database comes from Trackman data. The data comes in the form of a CSV file and describes information like the batter, the pitcher, the play result, the pitch speed, etc. This information is made available to each team who uses Trackman. In addition, teams who use Trackman have access to all the game data from every team that uses Trackman. Because of the vast data collected, and the ability to access this data for every team, Trackman is our single source of data. Inputting data follows the following workflow: first connect to the FTP server which downloads every CSV file from a directory, then parse the file, and last upload the parsed data into the database. The parsing part of that workflow is written using Python, the rest of the workflow code is written in C++ and calls the Python parser within the program. The parser will automatically pull new data from FileZilla where all the Trackman CSVs are stored. This is planned to automatically happen on Monday mornings, after discussion with our sponsor. If needed this time will change, but at this moment we do not anticipate that. We will store data from last season forward.

Back-end: connecting the AI models

Our team solely developed the database and UI for this project, and other senior design teams have developed AI models based on the data in the database. For cycle 3, the next step is to integrate their AI models into our server and database so that they display on the UI.

Initially we planned to implement an API for the modeling teams to connect to the database. After much discussion, especially with the other teams, we decided against an API. An API would be a slow solution, would take considerable work to maintain, and is not the most reliable solution as it needs an internet connection. Instead of an API, our team developed the following solution. The defensive shift modeling team and the Industrial and Systems Engineering team have been provided an SQL view. These models require calculations, so the views perform the necessary calculations on our end and then the models pull data from those SQL views. The heat map team has a direct connection into the database instead of an SQL view. This team needs raw pitch data instead of aggregated data, which is why we provided the direct connection.

The modeling teams’ code has been hosted locally on our server in Docker containers. The code runs on the server, and the output data from their models are stored in a model specific table in the database. Our team has added a table for each modeling team in the database, resulting in 3 new tables. Every time the code runs the data in the table is replaced as updated data gets inputted. To view the model on the UI, our team queries the data from the modeling teams’ tables and outputs an image on the website. Our team wrote code on the front end that takes the data and produces the image.

Front-end

To develop the front-end of our project, the decision was made to use the React framework Next.js along with the React component library Material UI. The decision to use both of these tools came from a priority to choose popular and well-known technologies that are respected in their field and will allow further continuation of this project for semesters to come. Additional reasons we chose Next.js is it uses the React library and extends the latest of React’s features, the ability to develop full-stack applications allowing ease and efficient connections to our back-end, and the ability to do server-side rendering which will improve load and speed times especially when data is queried from the database. Additional reasons we chose Material UI is it implements Google’s Material Design, an adaptable system of guidelines, components, and tools that support the best practices of user interface design, and the inclusion of premade base components and styling ensuring consistency across the interface with multiple developers working on the project over a long duration of time.

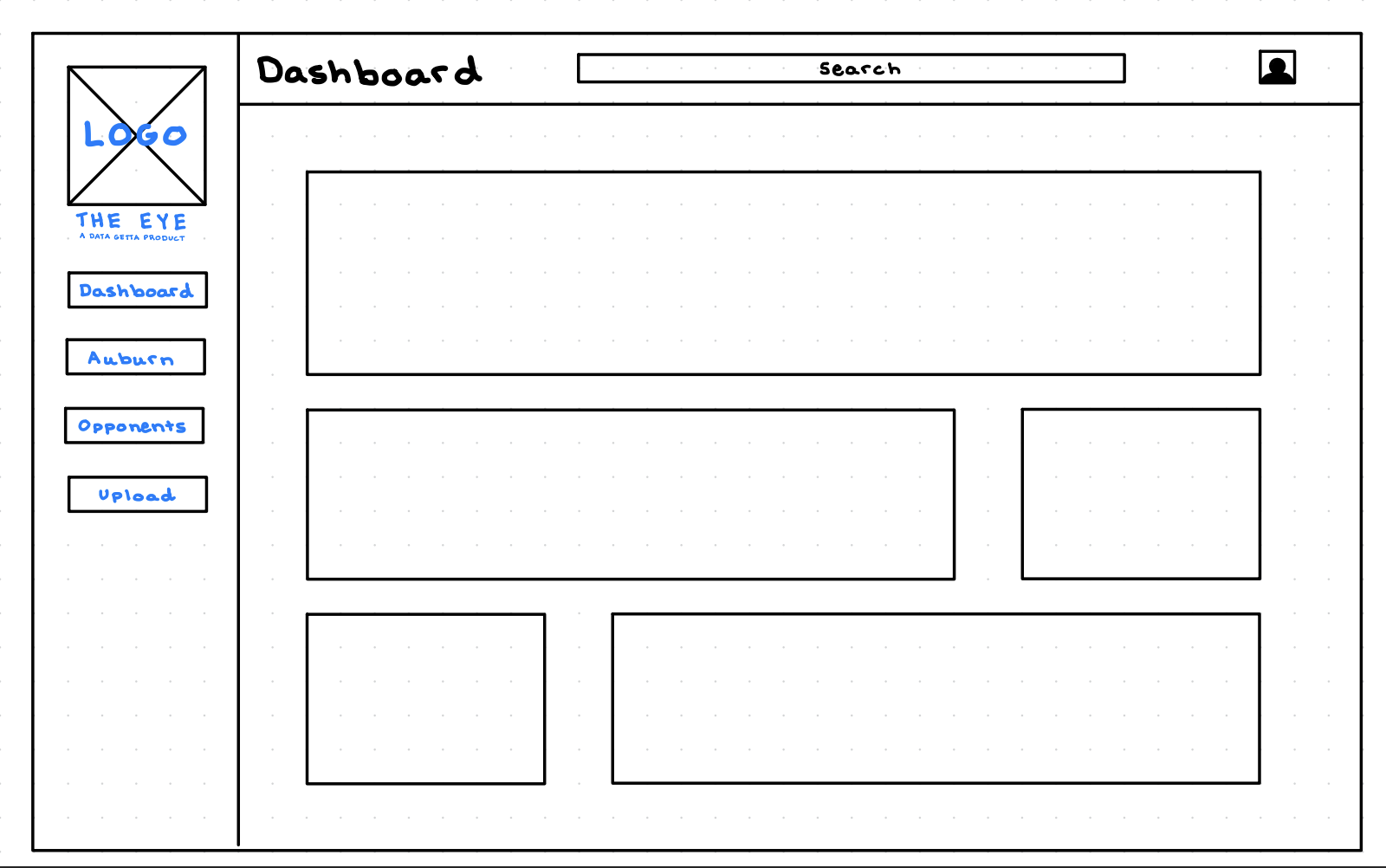


Figure 2.1 Wireframe of the overall layout of the interface

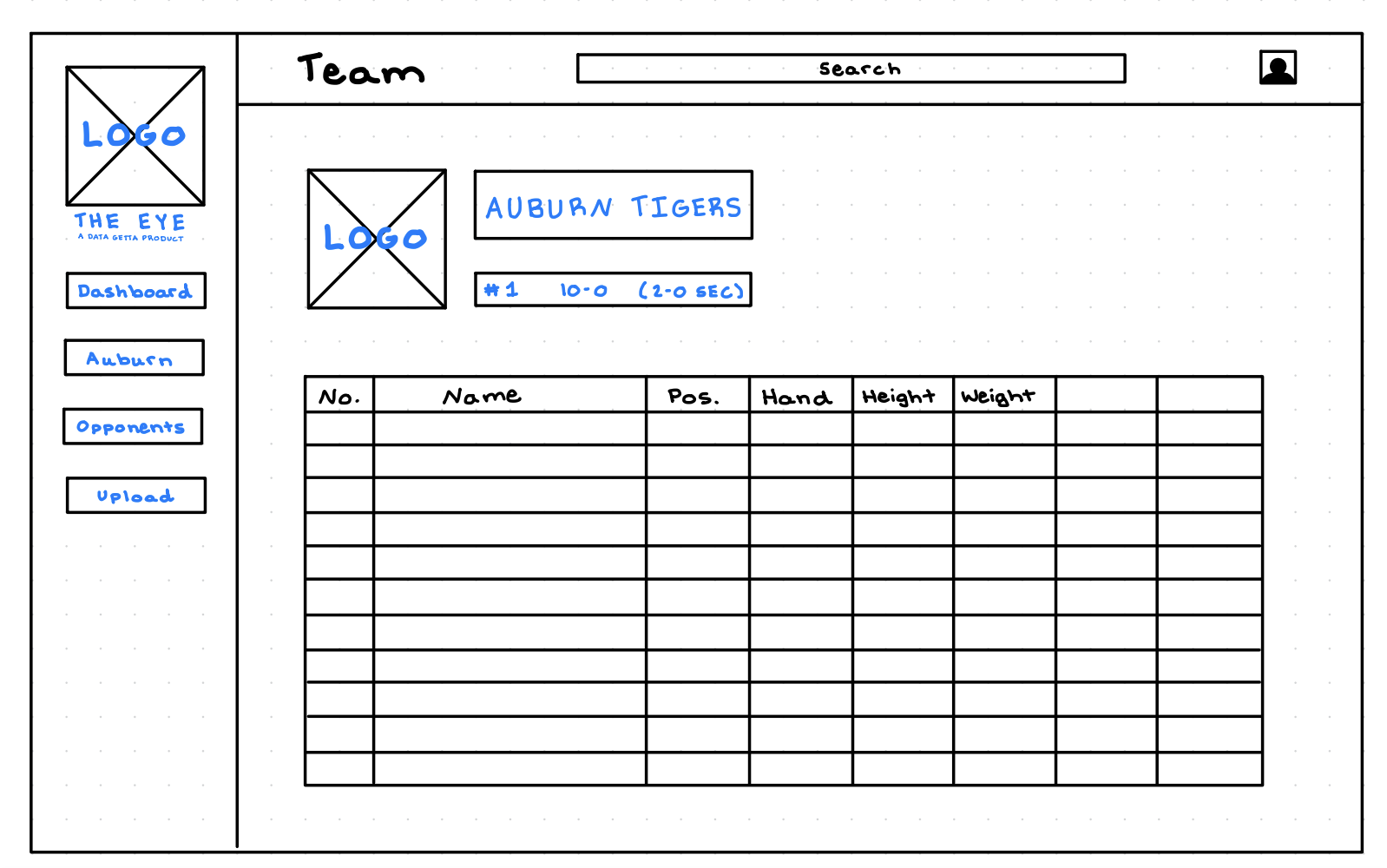


Figure 2.2 Wireframe of a team page

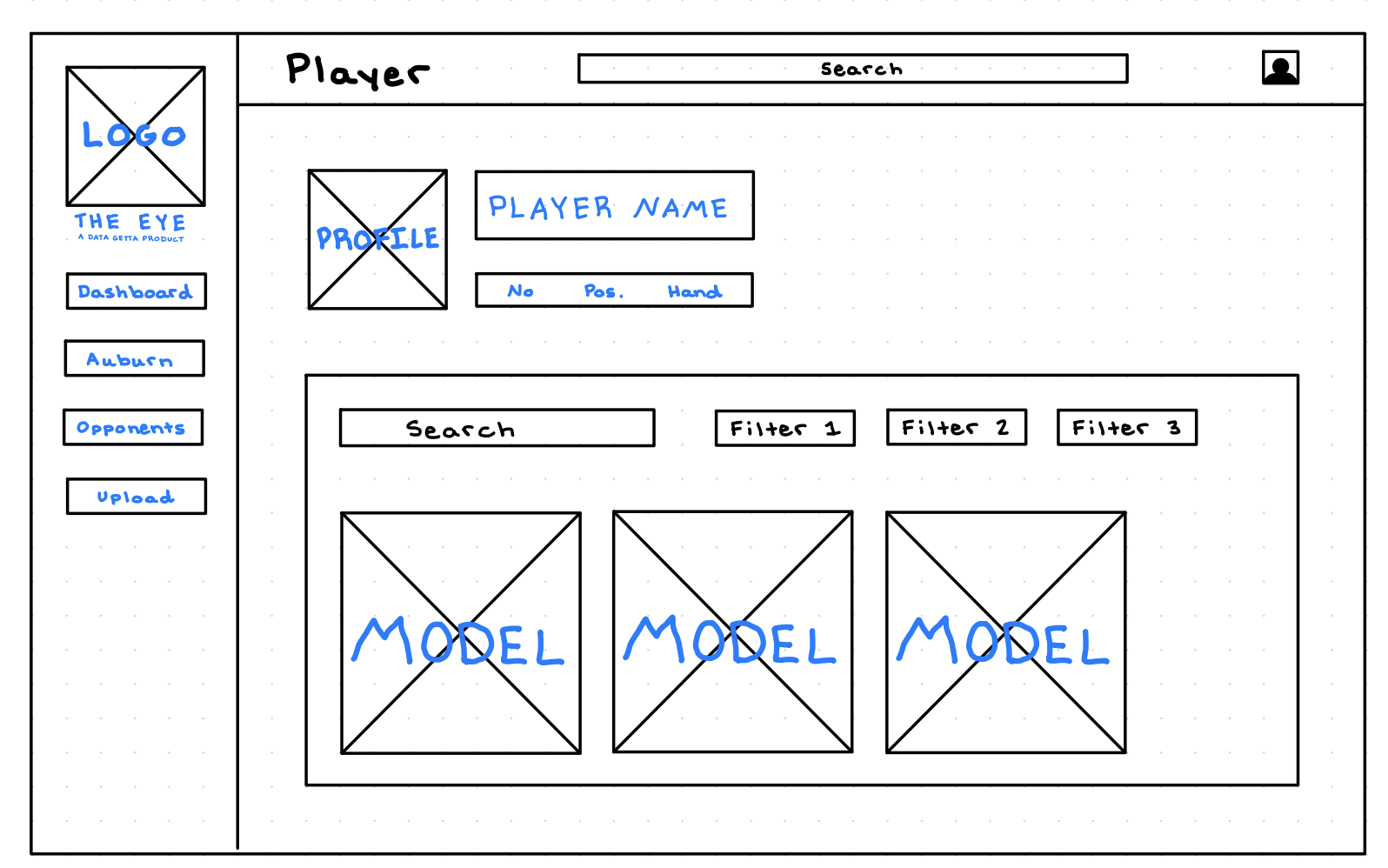


Figure 2.3 Wireframe of a player page

Figures 2.1 - 2.3 show sketches created to give the group members working on the interface an initial idea of how to create their high-fidelity prototypes that will be shown to the sponsor and baseball team staff to receive their inputs and opinions on what features they do and do not like.

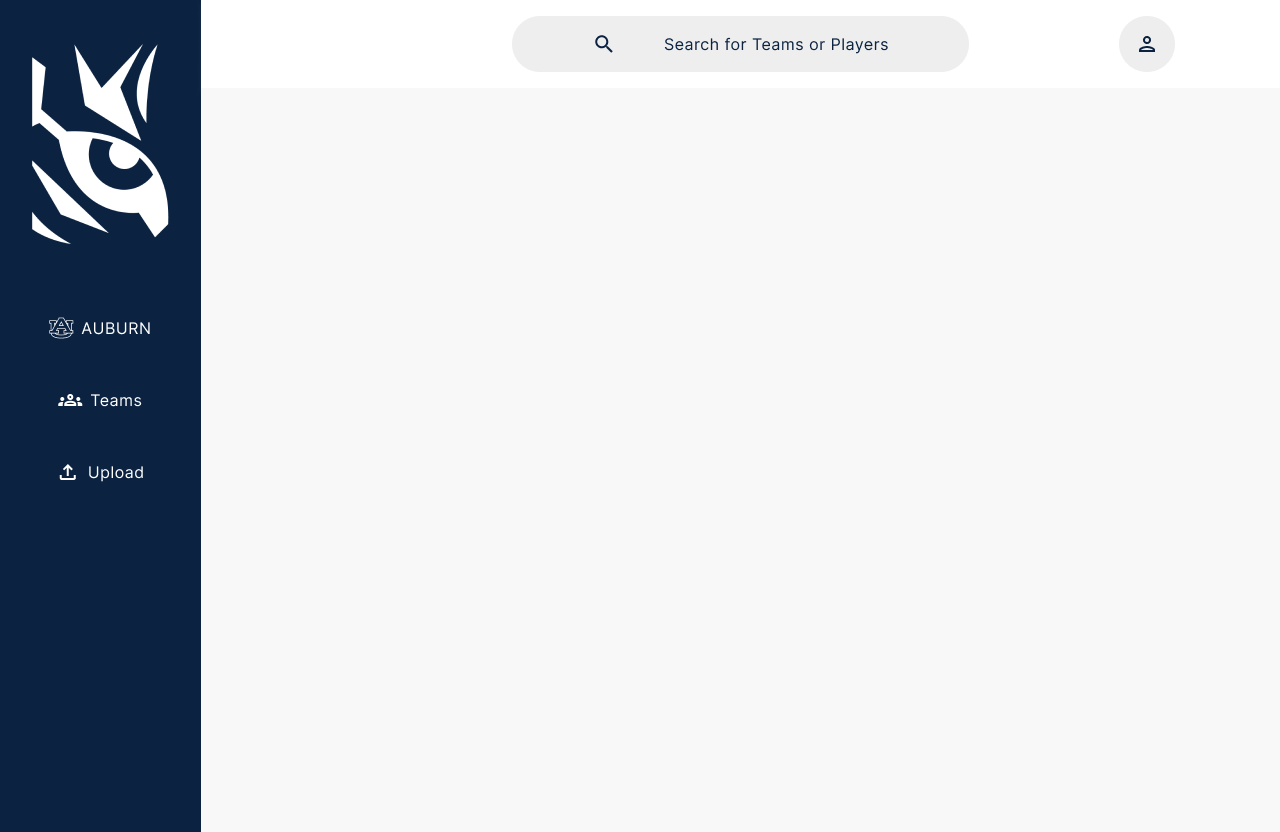


Figure 3.1 High-Fidelity Prototype of the overall layout of the interface

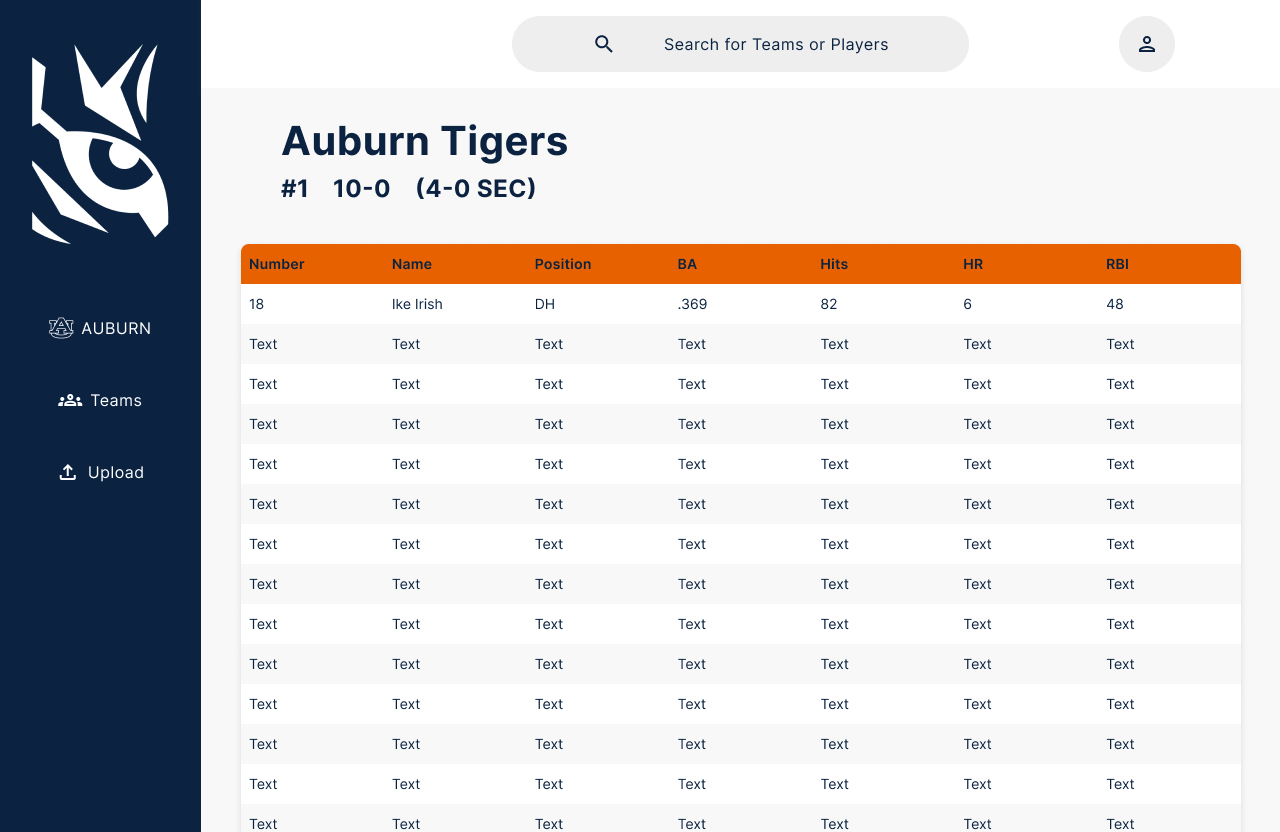


Figure 3.2 High-Fidelity Prototype of a team page

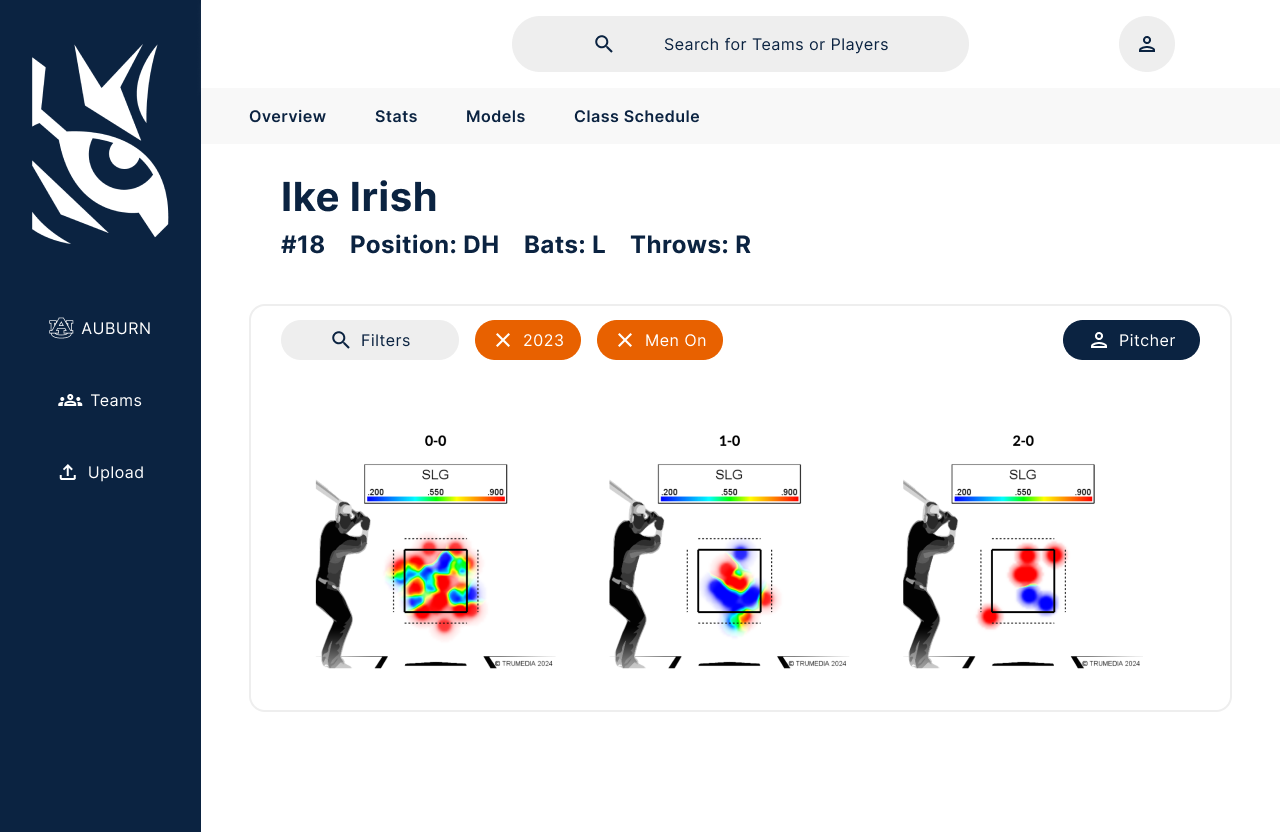


Figure 3.3 High-Fidelity Prototype of a player page

Figures 3.1 - 3.3 show one of the sets of high-fidelity prototypes shown to the sponsor and baseball staff to to receive their inputs and opinions on what features they do and do not like.

## User Stories

Contributors: All

| 1.0 Administrator Log In |
| --- |
| Summary: As a coach, I want login authentication, so that only Auburn University coaches and players have access to this data. |
| Description: This system stores a lot of information that could be harmful if leaked to non-affiliated parties. Therefore it is necessary to design a login feature that emphasizes security and usability. We should include access to the login page, the ability to enter valid credentials, “forgot password” feature, and a log-out feature. |
| Planned Hours:   * Total: 8 * This Cycle: 8 |
| Actual Hours:   * Total: 2 * This Cycle: 2 |
| Coder Names: Caden Garrett |
| Test Names: Caden Garrett, Braden Mosley |
| Reviewer Names: Caden Garrett, Braden Mosley |
| Story Status: Completed |

| 2.0 Schema Design |
| --- |
| Summary: As a user I want fast lookup speeds so that it does not take minutes or hours to sort and view the data. |
| Description: This system has a lot of data to retain and then display as needed. Therefore it is necessary to design a schema for this data that supports efficient loading and lookup. Our team needs to ensure tables are properly set up, relationships between tables are clearly defined, and that the data is indexed properly. |
| Planned Hours:   * Total: 4 * This Cycle: 10 |
| Actual Hours:   * Total: 20 * This Cycle: 8 |
| Coder Names: Laura Cromwell, Caden Garrett, Annie Murphy, Jacob Munroe |
| Test Names: Caden Garrett |
| Reviewer Names: All |
| Story Status: Completed |

| 3.0 CSV Parser |
| --- |
| Summary: As a user, I want to input data from a game and upload it to the database. |
| Description: A simple python script that reads the data from a given csv, separates the data, and uploads the data to the database. |
| Planned Hours:   * Total: 2 * This Cycle: 2 |
| Actual Hours:   * Total: 6 * This Cycle: 4 |
| Coder Names: Micah Key |
| Test Names: Micah Key, Braden Mosley, Caden Garrett |
| Reviewer Names: Micah Key, Caden Garrett |
| Story Status: Completed |

| 4.0 Roster Web Scraper |
| --- |
| Summary: As a user, I want to have access to all team rosters so I can view any player I want to. |
| Description: There are a lot of teams with a lot of players on them and manually inputting their names and teams would be very time-consuming, especially since they need to be refreshed every year. So, a web scraper is to be made to make this process super easy and mostly automatic to save time. |
| Planned Hours:   * Total: 2 * This Cycle: 2 |
| Actual Hours:   * Total: 2 * This Cycle: 2 |
| Coder Names: Caden Garrett |
| Test Names: Caden Garrett |
| Reviewer Names: Caden Garrett |
| Story Status: Discarded |

| 5.0 Hosting the Database and Interface |
| --- |
| Summary: As a user, I want to be able to see the website so that I can access all the data and models provided by it. |
| Description: We need to make sure our users can access all the data and the models. In order to achieve this we have to host the database somewhere. Our team needs to consider the costs and benefits between hosting the system through Auburn University OIT or through another cloud service like Microsoft Azure or Amazon Web Services. |
| Planned Hours:   * Total: 6 * This Cycle: 2 |
| Actual Hours:   * Total: 6 * This Cycle: 2 |
| Coder Names: Laura Cromwell, Caden Garrett, Annie Murphy, Jacob Munroe |
| Test Names: Caden Garrett |
| Reviewer Names: Caden Garrett |
| Story Status: Completed |

| 6.0 - Sort and Display Data |
| --- |
| Summary: As a coach, I want to sort and display to view the data more effectively. |
| Description: The data we gather needs to be sorted so that coaches can easily navigate important stats and look at the worst and best players in each category. The data also needs to be displayed in an intuitive way so that less tech-savvy coaches have an easier time navigating the web application. |
| Planned Hours:   * Total: 16 * This Cycle: 8 |
| Actual Hours:   * Total: 16 * This Cycle: 8 |
| Coder Names: Micah Key |
| Test Names: Micah Key |
| Reviewer Names: Micah Key |
| Story Status: Completed |

| 7.0 Filter Data |
| --- |
| Summary: As a user, I want to be able to filter data and save custom filters to query data and create custom models. |
| Description: With the large amount of data at the user’s disposal, certain queries and generated models can be very specific depending on what a player/coach might want to see and/or compare, so being able to filter out certain results and outliers that may skew results or could be time-consuming to filter down, giving a user a more efficient experience. |
| Planned Hours:   * Total: 16 * This Cycle: 16 |
| Actual Hours:   * Total: 14 * This Cycle: 14 |
| Coder Names: Jacob Munroe |
| Test Names: Jacob Munroe |
| Reviewer Names: Jacob Munroe |
| Story Status: Completed |

| 8.0 View Team Rosters |
| --- |
| Summary: As a coach or player I want to be able to view rosters for my team as well as opposing teams across Division 1 baseball. |
| Description: A lot of coaching decisions are made through the roster decisions of other teams, so it is important for coaches to be able to view other team rosters as they update throughout the season. Players must also be able to scout and analyze upcoming competition in order to adequately prepare for their performance. |
| Planned Hours:   * Total: 16 * This Cycle: 8 |
| Actual Hours:   * Total: 16 * This Cycle: 8 |
| Coder Names: Jacob Munroe |
| Test Names: Jacob Munroe |
| Reviewer Names: Jacob Munroe |
| Story Status: Completed |

| 9.0 View Player Page/Stats |
| --- |
| Summary: As a coach I want to be able to analyze the performance of specific players on my team so that I can identify areas that need improvement for that specific player. |
| Description: This will be a page of the user interface for analyzing specific players’ stats. I also want to be able to analyze specific players on opposing teams to find exploitable weaknesses for that player. |
| Planned Hours:   * Total: 16 * This Cycle: 8 |
| Actual Hours:   * Total: 16 * This Cycle: 12 |
| Coder Names: Braden Mosley, Frankie Donze, Teddy Cooke |
| Test Names: Frankie Donze |
| Reviewer Names: Frankie Donze |
| Story Status: Completed |

| 10.0 AI Model Integration |
| --- |
| Summary: Connection into database for Modeling and Analysis teams to pull data from. |
| Description: As the modeling and analysis team I want to pull accurate data from the database to implement into my models for them to learn and for me to perform analysis. |
| Planned Hours:   * Total: 16 * This Cycle: 7.5 |
| Actual Hours:   * Total: 10 * This Cycle: 10 |
| Coder Names: Jacob Munroe, Teddy Cooke, Caden Garrett, Braden Mosley |
| Test Names: Jacob Munroe, Teddy Cooke, Caden Garrett, Braden Mosley |
| Reviewer Names: Jacob Munroe, Teddy Cooke, Caden Garrett, Braden Mosley |
| Story Status: Completed |

# Management Plan

Contributors: Frankie Donze

We will continue with keeping the Orange Team and the Blue Team in close communication and working together on this project. We have decided to split our groups into two groups, with two members from each team in the groups. The first team is the Database Team, who will handle the schema, setup, and information organization of the database. Members of the Database Team include Annie, Laura, Caden, and Jacob. The next team is the Interface Team, who will develop a web-based user interface to display the data for the Baseball Team to interact with. Members of the Interface Team include Braden, Frankie, Micah, andTeddy.

Because we have our Database team, the AI Modeling team, and a senior design team from the Department of Industrial and Systems Engineering, Coach Teaford has emphasized the importance and need for clear communication between all involved. We have made this a priority within our project, and have implemented team group chats and a project-wide Discord server to ensure that all involved are on the same page. As well as online communication throughout the week with all teams and coach Teaford, we also have a weekly in-person meeting at 8 am every Monday to talk about progress, as well as discuss the next steps and delegate tasks for the week.

We have officially set up our version control tool GitHub. We will use GitHub to ensure all team members are working on the most recent version of the project. GitHub will help us stay organized and reduce the chance of any redundant work being done.

During Cycle 2, we finalized the database schema and began importing data. Our CSV parser has been completed and tested. Through collaboration within our team and with the coaches, we have identified and implemented the formulas to calculate statistics within our database. The user interface is almost completed but will need adjustments to accommodate the calculated statistics as well as the modeling teams’ outputs. The team has continued to have weekly meetings with all AU baseball senior design teams, Coach Teaford, and Adam Nebel as well as just our database teams.

During Cycle 3, we integrated the AI models into the database and UI. We engaged in lengthy discussions with the AI modeling teams to understand what information they utilize in their models. The outcome of these discussions led us to grant direct access to the database, as well as develop SQL views for any calculations needed in the AI model. We have also made updates and adjustments to the UI as needed as we received feedback from Coach Teaford. We also have been in contact with OIT to set up authentication through Auburn University, so that users login with their Auburn credentials. Lastly, we are preparing our resources and starting to develop documentation so that future groups can successfully partake in this project. Throughout the cycle we have continued weekly meetings with all AU baseball senior design teams, Coach Teaford, and Adam Nebel as well as just our database teams.

Future work for future groups:

* Adding a search bar that enables users to search for a specific player or team
* Any additional UI features or tweaks that Coach Teaford requests
* Finalizing authentication with OIT and SSO login

# Test Results

Contributors: Jacob Munroe

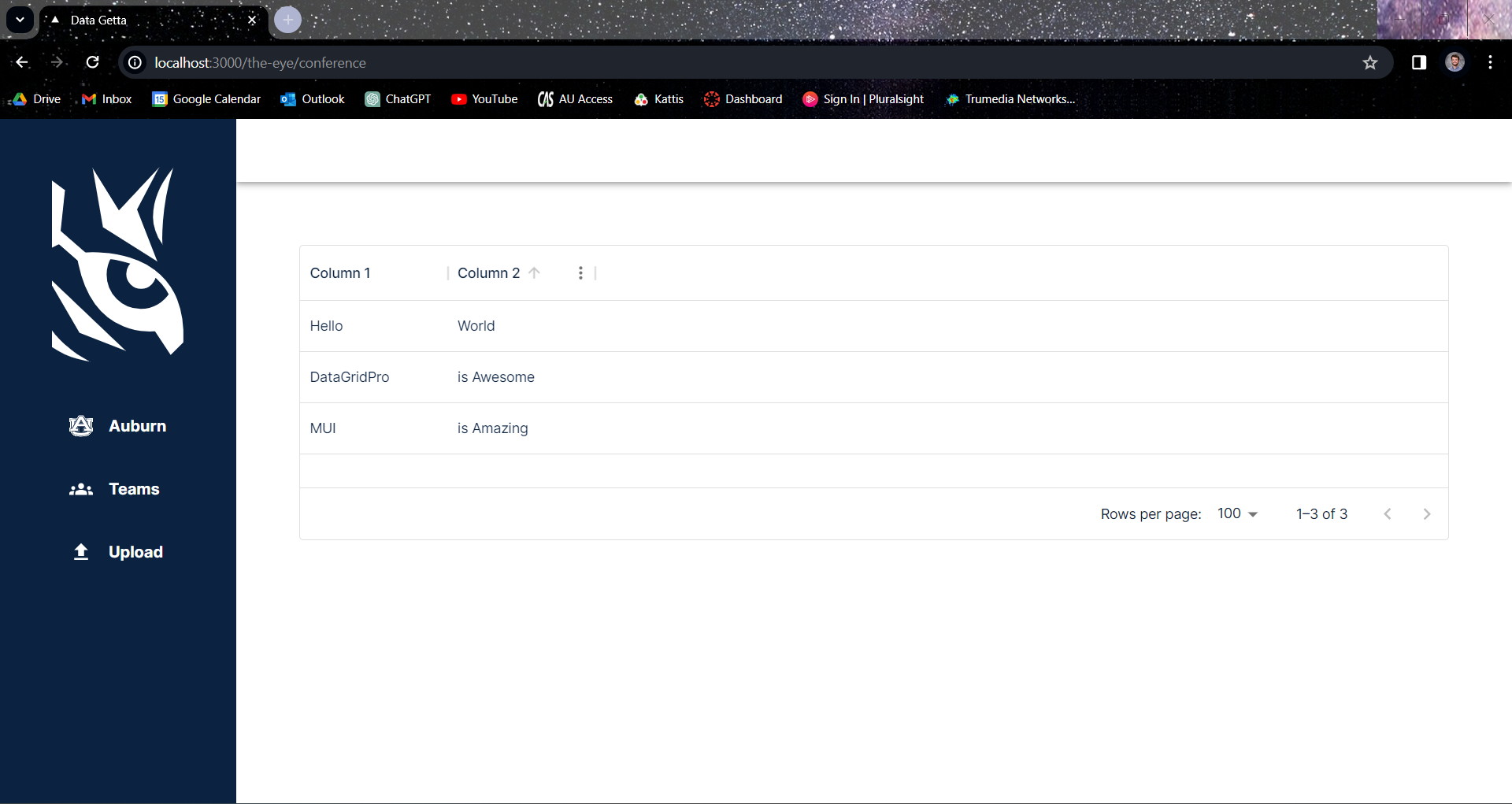
When it comes to testing our system, we are following a 3-step process:

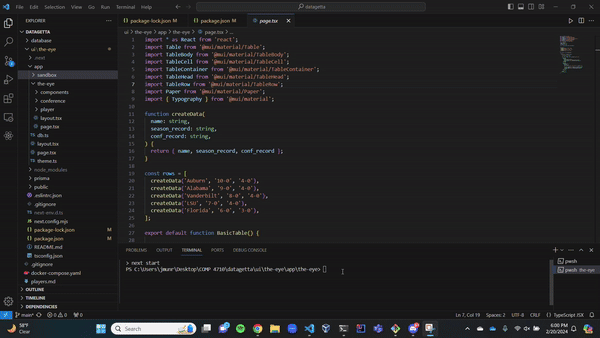
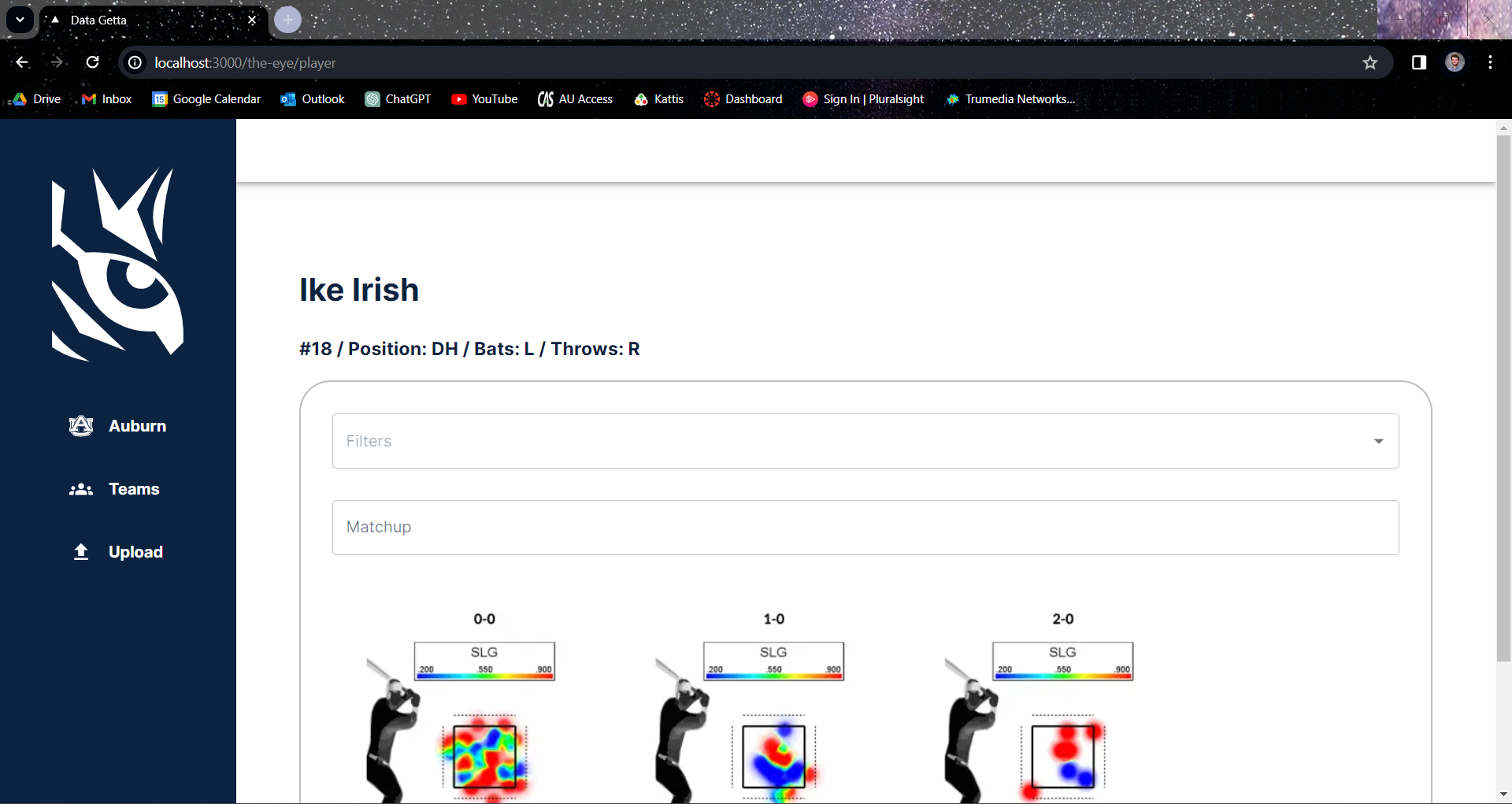
1. Integration Testing
2. Performance Testing
3. Usability Testing

This process allows us to introduce features into our product in an iterative design process that also includes the Auburn Baseball team and their feedback. Integration Testing allows for us to test how the system will react as we introduce new features to ensure that we will not have any ill-fated reactions to new features. Performance Testing will give us an idea of how the new feature will perform when scaled to the system’s size to ensure that feature performs to the standard that we expect as well as the system as a whole. Usability Testing is where we will see how the features perform for our clients and verify that the new feature introduced has been integrated in a helpful and useful way. With this 3-step process, we are able to take feedback that we receive at each stage and make small changes to fix the issues along the way rather than making large changes when trying to push the feature in a crunch.

Cycle 1 Testing (Integration focused)

In Cycle 1 we were working to release our 1.0 version with a target date around the beginning of SEC play on March 15th. Currently, we have our application hosted through the University’s OIT with the below features currently developed. We look forward to moving into the Performance Testing process with our initial features including team retrieval, player stat retrieval, and trackman data retrieval. Below are screenshots from our Integration Testing of developed pages:





Cycle 2 Testing (Performance focused)

In Cycle 2 we now have added testing to ensure that the data entered into the database is stored correctly. This follows a two step testing process of testing the schema, then testing the upload of data into the schema for our parser.

Testing the schema was an iterative process of inputting data and viewing the results in the database. To do this, a local installation of Postgres and access to a few CSVs were needed. Once these were obtained, Postgres is installed locally and the current schema being tested is inserted into it. Then, insertions of the CSVs into the database were done and made easier using Dbeaver, an easy to use database manipulation tool, by using the built in tasks feature. This allowed for mapping of CSV columns to the schema tables and can be done repeatedly once set up. Finally, after inserting this data, the schema was inspected for any misconfigurations, data out of place, improper relationships, and any other errors related to the schema.This allowed for us to come up with the idea to move calculations from a regular table to views as this will prove to be a much better solution to the problem.

Testing the parser also followed an iterative process of expanding test cases. The first test case only included practice data from Auburn’s practice games. The next test case included one game day's worth of data for each team in Trackman. The final test case includes game data on each team in Division 1 from this past month. Criteria to consider a test case passed include: verification that variables in each column line up correctly, and that the data inside each column looks correct (no abnormally large numbers, for example). Eventually we will include data from this past year. Testing revealed that Trackman was inputting data from teams in Divisions 1, 2 and 3. From this we have adjusted our inputting process to only include teams from Division 1, since Auburn only plays teams in Division 1.

Cycle 3 Testing (Usability focused and Integration focused)

In Cycle 3 we have now connected the AI models into the UI and have successfully uploaded data related to common pitching and batting statistics. To test usability, Coach Teaford has been interacting with the system and providing feedback as we add more features to the UI. To respect Coach Teaford’s time and busy schedule, we decided to receive feedback conversationally at meetings, instead of through an official feedback form, that would require extra time and effort from him. We have been receiving feedback in week iterations: every Monday when our team meets Coach Teaford has been providing us with changes he wants to the UI. Teaford has continued to give positive feedback throughout our cycle, and has given us overall approval of the UI. We have received some feedback, and have made those adjustments as requested. The first edit was changing some common pitching statistics, like innings pitched, to be decimals instead of integers. The second edit requested was our individual player statistics page, that initially included tables of information and was changed to bar graphs that aid visual understanding.

In addition to UI testing we also conducted Integration testing to integrate the AI models into our server and database. Our team collaborated with the AI model teams to successfully create a docker container for their model, connect to the database, and write to the database. As our teams encountered errors and bugs we communicated both in person and over Discord to resolve the issues. Integration testing has been an important part of this cycle to ensure that all models are connected to the database correctly, pulling the right data, and writing accurate data to their tables for us to then generate visualizations of their models.

# Lessons Learned / Challenges

Contributors: Annie Murphy, Laura Cromwell

1. **Time Required for Data Imports**: Coach Teaford has expressed the need to include our game data, practice data, other teams’ game data, and all data from past years in our system. Due to the sheer amount of data we will be using, we have determined it will take a while to get caught up from past years, and a smaller but still significant amount to upload each week.
2. **Clarification of Other Teams’ Requirements:** During this cycle, we have continued the discussions of what the other AU baseball senior design teams need. Having an idea of their goals have impacted our UI design plans and have pushed us to create some “standard” way for them to retrieve data. We planned to implement an API for these modeling teams to pull their data from, but we instead decided to give them direct access into the database. Additionally, we used database views to calculate the averages they needed for their models.
3. **Integration of Other Teams’ Models**: As we’ve finalized our major design implementation for the database and user interface, we have moved into integrating the other teams’ AI models. With the integration of new code, we have run into a number of errors over the weekend. After an extension conversation over the Discord and our Monday morning meeting, we were able to successfully get the models integrated.

# Memoranda

Contributors: All

[Link to all meeting notes](https://docs.google.com/document/u/0/d/1zTX4V2OhcKLaDK_ggbvqOMCj8uptEkAQWR-p0RsFn_o/edit)

Cycle 3 Meeting Notes:

4/2/24 @3:30pm

Length: 1.5 hours

Present: AU Baseball Database teams (Orange and Blue)

* Discussing w other senior design teams of how to integrate their models
  + No API
  + Just give them access
* Senior design showcase on Thursday
  + Poster has been finalized, will print

4/8/24 @8am

Length: 1 hour

Present: AU Baseball Database teams (Orange and Blue), Senior Design Modeling teams, ISE Senior Design team, Coach Teaford, Adam Nebel

* View in database: instead of a table it’s just a formula
  + Statistics calculated using views
* How to differentiate in the middle of the game when the innings switch
  + Use a trigger after inputting a game?
  + Set variable equal to first team of a game?
* Starting lineup
  + Give capability to coach to input starting lineup
* List that it is an approximate value in the UI
  + Whiff and chase percentages

4/15/24 @8am

Length: 1 hour

Present: AU Baseball Database teams (Orange and Blue), Senior Design Modeling teams, ISE Senior Design team, Coach Teaford, Adam Nebel

* Continuing to integrate AI models into our system
* Beginning to finalize our deliverables and will begin outlining goals for summer and fall design teams
  + Completing reports and documentation for this

# Source Code

Contributors: All

Cycle 3 Highlights:

1. **CSV parser:** The csv parser continues to parse and load data into our database, and we have added automatic, scheduled parsing functionality.
2. **SQL views for the AI models**: The Defensive Shift AI model and the Heatmap AI model utilize certain calculations of the data in the database. We have written SQL views that complete the calculations on our end, and the AI models can then utilize the data from those calculations in their models.
3. **SQL tables in the database for the AI models:** Once the AI models are finished running, they need to store their output to be manipulated. To accommodate this we have updated the schema to include tables for each AI model that stores the output data. These tables are stand alone tables and do not interact with the Trackman data tables. Each time the AI models run these tables update the data stored, replacing old data with new data.
4. **Integrating the AI models into the server:** Part of integrating the AI models into the system is integrating their code into our server. To achieve this we have updated our docker-compose file to include containers for the AI models being developed.
5. **Developing AI model images:** With the AI model output being stored in tables in the database, we need to take that data and transform it into an image to display on the website. We have developed code within our UI that does just that, it takes the data from that AI model’s table and develops the appropriate image to visualize that model on the website.

[Link to our GitHub repository](https://github.com/bradenmosley/datagetta)

# Presentation Slides

Contributors: All

| [AU Baseball DB C1 Presentation](https://docs.google.com/presentation/u/0/d/1DlhH75N35pwBMDOkHg7csyF2SXcnS1bI_Te_5GSINc0/edit) |
| --- |
| [AU Baseball DB C2 Presentation](https://docs.google.com/presentation/d/1Z_bxJiWxVZvGInJJxdJCF3Bydd-s3i_lKpX9PfNldzU/edit?usp=drive_link) |
| [AU Baseball DB C3 Presentation](https://docs.google.com/presentation/u/0/d/1_fdibqGWujtTjjHSvRptWDas0x2pNHCYBEVskbhI2kE/edit) |

# Previous Cycle Documents

Contributors: All

| **Link** | **Grade** |
| --- | --- |
| [AU Baseball DB ArchSpike Report](https://docs.google.com/document/u/0/d/1lwp5IrcnW67RzPcHakxiheKahmz9HEjI0yyh3J2-5Wo/edit) | 4/4 |
| [AU Baseball DB ArchSpike Presentation](https://docs.google.com/presentation/d/1D6bvxMDfrJ5ddSs0TZ2ja0rgIXxUICSs/edit?usp=sharing&ouid=108740258898027032018&rtpof=true&sd=true) | 4/4 |
| [AUDB Meeting Notes](https://docs.google.com/document/u/0/d/1zTX4V2OhcKLaDK_ggbvqOMCj8uptEkAQWR-p0RsFn_o/edit) | - |
| [Status Report](https://docs.google.com/spreadsheets/u/0/d/1fk1K5ryEUwZ1KLwSItuJoFcL_24sdYUM/edit) | Completed |
| [Status Report 2](https://docs.google.com/spreadsheets/u/0/d/1frGen6AR2vhlhUT1ZjGAMaC9sLzE3MtQ/edit) | Completed |
| [AU Baseball DB C1 Report](https://docs.google.com/document/d/1WNHabGe8BBkWjGAfcrc9z5dh_R7MHzP_E9HN2fSnwPI/edit?usp=drive_link) | 4.92/5 |
| [AU Baseball DB C1 Presentation](https://docs.google.com/presentation/u/0/d/1DlhH75N35pwBMDOkHg7csyF2SXcnS1bI_Te_5GSINc0/edit) | 3/3 |
| [Status Report 3](https://docs.google.com/spreadsheets/d/11arYYXz1l2TL_K0BBI6n5_6D2QuAnqEv/edit?usp=drive_link&ouid=108740258898027032018&rtpof=true&sd=true) | Completed |
| [Status Report 4](https://docs.google.com/spreadsheets/d/1suN7frGhwSuPzV1xSzLN_AzoqNfWnvKv/edit?usp=drive_link&ouid=108740258898027032018&rtpof=true&sd=true) | Completed |
| [AU Baseball DB C2 Report](https://docs.google.com/document/u/0/d/1cbcb6_f1k3A3HKpMkAaaKZWfLo3xJ3e7sgIRZRC5e0c/edit) | 5/5 |
| [AU Baseball DB C2 Presentation](https://docs.google.com/presentation/u/0/d/1Z_bxJiWxVZvGInJJxdJCF3Bydd-s3i_lKpX9PfNldzU/edit) | 2.91/3 |
| [Status Report 5](https://docs.google.com/spreadsheets/u/0/d/1U40FXwCNY6mEdz-YtK6eckVQDOj-NJmH/edit) | Completed |
| [Status Report 6](https://docs.google.com/spreadsheets/u/0/d/1Bbo5XVDGTn51b_ZNl2-Om-WaSTV-6aSR/edit) | Completed |

# Sponsor Approval

The written report and presentation for Cycle 3 were sent to Coach Teaford and Adam Nebel and approved via email on April 18, 2024.