Group 11
Live Sports Score Tracker
Software Engineering Project

Name (s): Alexander Diaz, Bila Fofung, Luis Gonzalez Gonzalez, Sakib Jamal, Vijay Wulfekuhle

Semester: Fall 2024

Group Number: 11

Coordinator: Alexander Diaz

Name of the Guide: Dr. Tushara Sadasivuni

Date: (10/16/2024)

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Software Engineers' Information

Below, each member's resume is linked, detailing their skill-set:

Alexander Diaz
Bila Fofung
Luis Gonzalez Gonzalez
Sakib Jamal
Vijay Wulfekuhle

1.2 Planning and Scheduling

Assignee Name	Email	Task	Duration (hours)	Dependency	Due date	Note
Alexander Diaz	adiaz23@student.gsu.edu	Class diagram functions, database tables, behavoiral model			Sunday 11:59 PM	none
Bila Fofung	bfofung1@student.gsu.edu	Class diagram			Sunday 11:59 PM	none
Luis Gonzalez Gonzalez	lgonzalezgonzalez1@student.gsu.edu	Class diagram attributes/updates, updated problem statement, document format/page numbers	1	Layout of class diagram complete	Sunday 11:59 PM	none
Sakib Jamal	sjamal6@student.gsu.edu				Sunday 11:59 PM	none
Vijay Wulfekuhle	vwulfekuhle1@student.gsu.edu	Behavioral modeling			Sunday 11:59 PM	none

1.3 Teamwork Basics

Ground Rules

1. **Work Norms:** Teams should decide how to divide tasks, set deadlines, and handle missed deadlines.

2. **Facilitator Norms:** The facilitator's responsibilities and whether the role will rotate.
3. **Communication Norms:** Agree on when and how to communicate, considering preferences for methods like email or phone.
4. **Meeting Norms:** Coordinate meeting times and locations. Decide what to do about lateness, absences, and missed meetings.
5. **Consideration Norms:** Address what can/cannot be done during meetings, such as eating or smoking. How can norms be changed if necessary?

Hints for Handling Difficult Behavior

- **Overly Talkative:** Someone might dominate the conversation for various reasons. If subtle redirection fails, a private discussion might be needed.
- **Too Quiet:** A quiet team member could be shy or disengaged. Efforts should be made to draw them out, like asking their opinion or making them feel valued.
- **Argues:** If someone is critical of ideas, use it as constructive feedback. If they are criticizing others, tell them how this affects the team and ask them to be more mindful.
- **Complains:** Address the complaint by listening and set time aside for the group to resolve it and involve the person in finding a solution.

Hints for Handling Group Problems

- **Floundering:** Early team struggles can be managed by listing tasks and focusing on moving forward with clear objectives.
- **Going Off on Tangents:** A little side conversation is fine, but if it distracts from work then steer the group back to the project.
- **Making a Decision Too Quickly:** Someone may rush decisions. Ensure everyone agrees before moving forward.
- **Not Making a Decision:** Consensus is ideal. Tools like multivoting or distributing points can help if the group struggles to decide.
- **Feuding Between Members:** Conflicts need to be addressed before the team can progress. There must be discussion to resolve issues.
- **Ignoring or Ridiculing Others:** Exclusion or ridicule can damage group dynamics. Encourage cooperation and respect among all members.
- **Not Doing Their Share:** If a member isn't contributing, talk to them directly, explaining how it affects the group.

1.4 Problem Statement

- **What is your product, on a high level?**

This is a mobile application that tracks and analyzes live data from football games, with AI analysis and user interaction.

- **Whom is it for?**

This is for sports enthusiasts who want to utilize this data for whether to assist for sports betting or general game data purposes.

- **What problem does it solve?**

The app provides a live football data and match analysis app that shows the most important match information in an easy-to-understand format.

- **What alternatives are available?**

Alternatives to our apps are usual data tracking sports apps or information from the sport organizers' sites. (PlayerData EDGE and SkillCorner)

- **Why is this project compelling and worth developing?**

This project is ambitious to us because we are leveraging modern technology for common folk like us to enjoy football in a more leisurely environment.

- **Describe the top-level objectives, differentiators, target customers, and scope of your product.**

Top Level Objectives:

- Present matchup probabilities to users through regression and binary classifier models
- Provide score updates to users live through push notifications
- Create a chat-bot in place of a real-life live commentator, offering updates to match-time, scores and player stats as a match goes on
- Allow users to interact in the forms of real-time text chat about ongoing matches as well as live predictions for match events.

Target Customers:

- Football fans

Scope:

- Allow users to opt in to updates for any NFL game of their choosing

- **What are the competitors and what is novel in your approach?**

The competitors are other live football or sport data tracking applications, such as the NFL or ESPN applications.

Our approach is different because it combines data tracking with the live social aspects of chat and predictions, giving users more of their needs in one place.

- **Make it clear that the system can be built, making good use of the available resources and technology.**

1. SQL
2. Key-value data set
3. Unstructured data set
4. APIs for sports data
5. Locally-run AI models for predictions

- **What is interesting about this project from a technical point of view?**

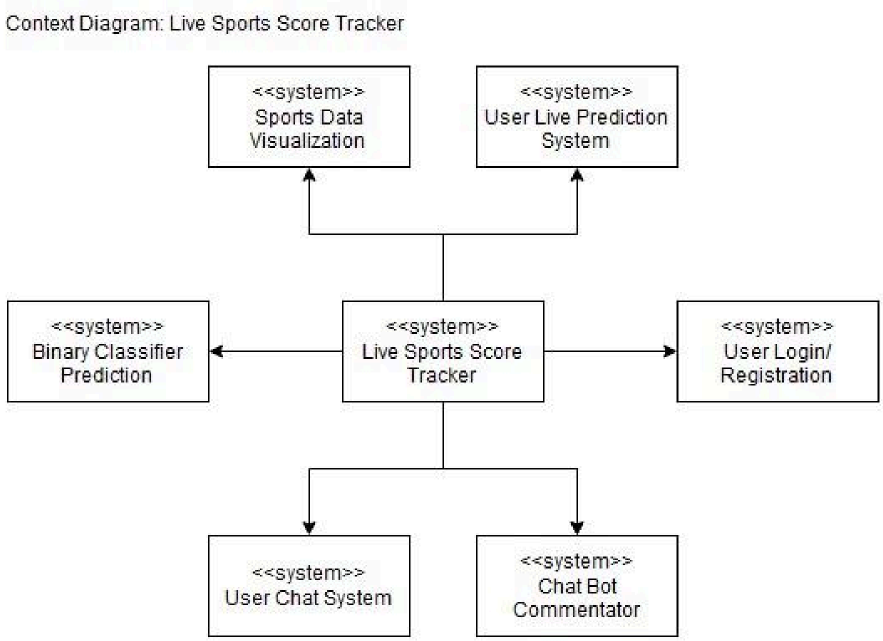
We will be using multiple technologies in combination, such as machine learning models, in order to provide a more comprehensive analysis of the live data.

- **Do you have a client login and an admin login?**

We plan on being able to implement a client and admin login into the app, to allow elevated privileges and to provide commentary.

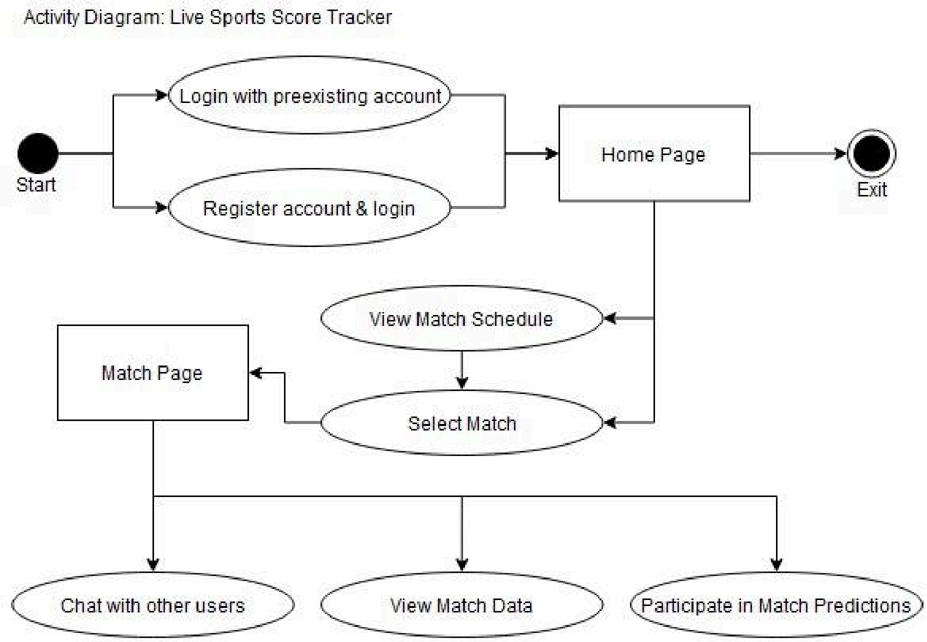
1.5 System Requirements

1.5.1 Context Diagram


Context Diagram: Live Sports Score Tracker
<<system>> Sports Data Visualization
<<system>> User Live Prediction System
<<system>> Binary Classifier Prediction
<<system>> Live Sports Score Tracker
<<system>> User Login/ Registration
<<system>> User Chat System
<<system>> Chat Bot Commentator

(See Ch5: Section 5.1 and Figures 5.1) (Chapter 1 Slide 45)

1.5.2 Activity Diagram



(Chapter 1 Slide 43)

2.0 **Requirements**

2.1 **Use Cases**

Use Case #: 1

Name of use case: data in live game

Actors: user, live game, database

Description:

- The user will be able to load the live game they want to have loaded
- There should have a display at the top of the teams playing and the current score
- A chat box/ live box should be loaded to show the current things going on
- And a live data table showing when some plays happened and who was involved

Exception path: if there is no current game going on, will prompt there is no game happening and will go back to previous page

Pre condition: User selects game to see a live game stats

Post condition: display a live game data page with all pre conditions been met

Use Case #: 2

Name of use case: player data

Actor: user

Description:

- The user click on player database
- Loads up an empty table with a search box on top
- Types in name of a player
- Loads player from database and gives the players with that name or values in their name
- Can add players to the following list so it will be on home screen

Alternate path: if player name isn't in the database then nothing comes up

Use Case #: 3

Name of use case: Filter

Actor: user, data table column

Description:

- The user clicks on filter button
- Will have a all filters possible to change and add onto a table
- Each one will have check boxes to show which ones user wants
- Will change data to show with the column names

Use Case #: 4

Name of use case: simple chat

Actor: users, player data, live match

Description:

- Fans join a chat room with other users dedicated to current game
- A chatbot will post live data or current events in the game when something happens
- Users can talk into chat room and communicate with other fans

Alternate path: If player doesn't want to be in a chat then it will just have the chatbot showing the live game that has no user input just live updates

Use Case #: 5

Name of use case: live point feature

Actor: user

Description:

- Fans can give their predictions of a future game
- A changeable score will have the player ask to change what they think will be the outcome of the match
- Predictions can be stored in a cache and be displayed under the live game match
- Users can add who scores points in the match itself

Exception path: users don't need to add a predictions and their wont be an added number of predictions on the score display

Use Case #: 6

Use Case Name: Live Match Weather Conditions

Actors: User, Match Database, Weather API

Description:

- The user views a match's page.
- The match's location is found in the match database.
- The location is sent as part of a request to the weather API and current weather conditions for the location are displayed on the user's device.

Alternate Path: If the API did not return the information within 10 seconds, another request is made to the API and the received information is then displayed.

Exception Path: If the match has no location in the database or the information is not received after a retry from the API, display to the user that the application failed to get weather data.

Pre-Condition: The user is viewing a match page.

Post-Condition: Weather information for the match's location is displayed to the user.

Use Case #: 7

Use Case Name: User Login

Actors: User, Account Database

Description:

- The user presses the button to log in to an account.
- The user inputs their account username and password into a text form.
- The account database system checks that the password is correct for the input username.
- The user's device is logged in and given an account token.

Alternate Path: If the user cancels their login, return to the main page.

Exception Path: If the username and password do not match the account database's records, the login will fail and prompt the user that the password was incorrect for that username.

Pre-Condition: The user has registered an account.

Post-Condition: The user's device is logged in and has saved an account token. The user is taken to the main page of the app.

Use Case #: 8

Use Case Name: User Registration
Actors: User, Account Database
Description:

- The user presses the register account button.
- The user inputs their desired email, username, and password into a text form.
- If the inputs are valid, the information is saved to the account database.
- The user's device is taken to the login page.

Alternate Path: If the user cancels their account registration, return to the main page.
Exception Path: If the inputs are not valid or match specifications, the registration will fail and prompt the user to try again with information on why it failed.
Pre-Condition: The user has not registered an account.
Post-Condition: The account database has the user's account details saved. The user is taken to the login page.

Use Case #: 9
Use Case Name: Tailored Predictions
Actors: User, Match Database, Account Database
Description:

The data pages most recently viewed by the user are saved to the user's history in the Account Database.

The user's view history is analyzed, and upcoming matches related to the ones viewed by the user are found in the Match Database.

The found matches are displayed by order of relevance on the main page of the app.

Alternate Path: If the user doesn't have more than 5 matches viewed, predictions for upcoming games will be displayed.
Exception Path: If the user is not logged in to an account, the Tailored Predictions section will not be displayed.
Pre-Condition: The user is logged in to an account.
Post-Condition: The Tailored Predictions section is displayed on the main page of the app on the user's device.

Use Case #: 10
Use Case Name:User Makes Prediction against the machine
Actors: User, ML Classifier
Description: User selects a team to win/lose, the model will output the maximum likelihood probability that it thinks the outcome will be. If there is a difference between the computer's and user's prediction the computer can "accept" the bet.
Exception Path: The user tries to make a prediction that the model's maximum likelihood calculation agrees with. User cannot bet "against" the computer.
Pre Condition: Cleaned DB w/ scheduled web scraping script, Supervised trained model
Post Condition: User picks bet against model's bet. It gets added to user's bet

Use Case #: 11

Use Case Name: User makes over/under prediction
Actors: User, Database, ML regression model
Description: The section of the prediction component engages with a regression model that is trained w/ supervision upon rolling averages on the cleaned database. The model is trained on selected features to find the optimal function for accurate predictions. The user can choose one of these features with the ML model's prediction being the over/under metric.
Exception Path: There isn't enough data to make an accurate prediction, null over/under will be presented
Pre Condition: user account initialized, cleaned DB w/ scheduled web scraping script + training script for ML model
Post Condition: the select prediction gets added to their prediction.

Use Case #: 12
Use Case Name: trending
Actors: user, DB. K clustering (unsupervised model)
Description: The model will be trained on relevant features to find K clusters of categories of players/teams. The user scrolls the trending tab where certain groups of players/teams are displayed for found behavior of trends.
Exception Path: No relevant players are found, null display will be presented.
Pre Condition: Cleaned Database, trained clustering model,
Post Condition: A displayed section of the prediction component

Use Case #: 13
Use Case Name: Betting Options
Actors:User, database, betting system
Description: user will be able to place monetary bets on specific events in game such as what player would score next or how much a team would score during a specific quarter of a match.

- These options would be made available based on real time game data
- Users can set/confirm their bet during a live game.

Exception Path: If the user tries to bet on a game that is unavailable or if the window to bet is closed, a notification would pop up to tell them it is currently unavailable and then option will gray out.
Pre Condition:The user is watching a live game, and the betting system has access to real-time game data.
Post Condition:User bet is registered to the system with a confirmation notification and payout will be distributed after the event of said bet.

Use Case #: 14
Use Case Name: Post Game Analysis
Actors: User, Database
Description: After the match ends, users will be able to view a detailed breakdown of the game which includes player performance stats and key moments. Users can select specific events or players for a more critical analysis like average speed and scoring efficiency.
Exception Path: If the analysis is unavailable, a message will inform the user of the delay and that the analysis of the match will be accessible later.
Pre Condition:The match has ended, and the system has gathered all required game data to begin analysis.
Post Condition: The user can explore detailed post-match stats and insights.

Use Case #: 15
Use Case Name: Social Media Share
Actors: User, Social Media Platform APIs
Description: User will be able to share moment from the match that they are live betting on to social media platforms such as twitter and facebook.

- The system will create shareable screenshots and images, prediction outcomes, or stat highlights.
- User can add comments to said social media sharing post.

Exception Path: If there is an issue connecting to the social media API, the system will notify the user to let them know it is unavailable.
Pre Condition:The user has chosen what content they want to share, and connect their social media accounts.
Post Condition:The content is shared on the selected social media platform, with a confirmation pop up shown to the user.

2.2 **Requirements**

Requirement #: 1
Use case #: 1
Intro: Live game data
Input: user click when they load up the live game
Description: The user will be able to load up a data table and would keep getting more data from the live game that is happening and store it in the current data going on. It would then move the data after the game into the full big data.
Output: The table itself with new data being added from the game itself.

Requirement #: 2
Use case #: 2
Intro: Search player data
Input: keyboard inputs
Description: The user will be able to look up any player name or keywords and will load up a table of the names of the players.
Output: The table of players that would display from the search as a table

Requirement #: 3
Use case #: 3
Intro: Filter functionality
Input: user click a button and drop menu of all column names to see what to see in a table
Description: The user will be able to load up a data table with a small box with an arrow going down that when clicked on will drop down with checkboxes of each column name in the data tables. This will cause a new datatable to be shown with only those filters visible
Output: The table itself with new data being added from the game itself.

Requirement #: 4
Use case #: 4
Intro: Simple Chat feature
Input: user keyboard input and clicking onto the live game chat.

Description: The fans can use a real time chat function to discuss the match happening with live updates of the game going on with a chat bot. They can add themselves to chat by clicking a button from a live update feed.
Output: The chat will display the joining room and will load the chat with the other users talking and have the chatbot have live updates show up in the chat room instead.

Requirement #: 5
Use case #: 5
Intro: live pointing or predictions
Input: user click inputs and number score for the final of the match
Description: The users can have a prediction of the final score of the game by having a pop up window come in the middle and ask basic predictions from the score and then who would make points.
Output: The user after opening it would show the predictions window of scores that can be changed and a plus sign to add players who score points, And will show on the main hub that their predictions are saved with the current score and time near it to compare.

Requirement #: 6
Use Case #: 6
Introduction: Real-time weather updates for locations of the matches.
Rationale: Provides relevant information to the user about weather for ongoing matches.
Input: User's currently selected match, data from weather API.
Requirement Description: The application will display weather data for the selected match's location.
1. If the user has selected a match, the system will send a request to a weather API for information on the weather in that match's location.
2. If information has been received from the API within 10 seconds, display relevant weather conditions. If this fails, retry once.
3. If the request does not succeed within 15 seconds on the retry, display to the user that the application failed to get weather data.
4. Display current weather conditions and temperature to the user.
Output: Current weather data is shown to the user for a match when they select it.

Requirement #: 7
Use Case #: 7
Introduction: A 2 input text form for a user to login to unlock more functionality for the app.
Rationale: Login process for differentiation of users will allow personalized content delivery and let the user find what they care about more quickly.
Input: Username and password
Requirement Description: The user will be able to login to a previously created account if all requirements are met.

1. All fields must be filled.
2. Fields have valid characters.
3. Fields are within their length requirements.
4. Login details are verified with the accounts database.
5. An account token is generated that will be valid for one week.

Output: User is logged in and directed back to the main page. An account token is stored on the user's device.

Requirement #: 8
Use Case #: 8
Introduction: A 3 input text form for a user to register an account.
Rationale: Allows users to create an account themselves to be automatically added to the database.
Input: Email, Username, Password
Requirement Description: The user will be able to register a new account if all requirements are met.

1. All fields must be filled
2. Fields have valid characters. Fields can only contain alphanumeric characters and basic symbols. Passwords must include an uppercase character and a number.
3. Fields are within their length requirements. Email and username must be 5 to 22 characters long, the password field must be 7 to 22 characters long.

Output: User is directed to the login page to login to their new account.

Requirement #: 9
Use Case #: 9
Introduction: Personalized match predictions on the user's home screen.
Rationale: Allows the application to prioritize displaying content relevant and meaningful to each user's interests and interactions.
Input: User's activity history (last viewed items, followed teams)
Requirement Description: The user will be able to view a curated list of predictions on their home screen based on their preferences and recent activities.

1. The system will track and analyze the user's last viewed match pages.
2. The system will keep an updated list of teams the user has chosen to follow.
3. The personalized section will prioritize upcoming games involving followed teams and select games based on games viewed in the past.
4. Match predictions to be shown should be refreshed once per day to ensure relevance.
5. If the user has less than 5 match pages viewed, predictions for upcoming games will be displayed as a fallback.

Output: A list of tailored predictions is displayed on the user's home screen, sorted by relevance based on the user's activity and followed teams.

Requirement #: 10
Use Case #: 10
Introduction: the prediction component will have a classifier model for win/loss team prediction.
Rationale: A classifier model will be trained for w/l predictions of teams in DB, the model will be trained w/ supervision. The user has the capability to bet against the W/l prediction of the model. The user cannot make a bet that the computer agrees with.
Input: Combination of next week's match up
Requirements Description: A binary classifier model trained on past two seasons. Logic for proper available predictions ("real matches")

Requirement #: 11
Use Case #: 11

Introduction: The prediction component will have a regression model trained for over/under predictions
Rationale: For each selected feature/stat of the available prediction categories on expected value will be outputted based on the model's prediction for user's over/under consideration
Input: Feature(stat) + player/team
Requirements Description: A cleaned DB for ML training w/ scripting. Custom model output functions for relevant combinations, initialized user account
Requirement #: 12
Use Case #: 12
Introduction: The prediction component will have a trending tab. This is a k clustering model train to find meta trends on players/teams, the model is trained on relevant group features (statistics).
Rationale: The model will take in the player/team on cleaned_data w/ batch training. The model will output k clusters of trending categories with n players/teams. These clusters will be displayed to the user.

Requirement #: 13
Use Case #: 13
Introduction**:** In-Game Betting Functionality
Rationale**:** Enables users to place live bets on real-time game events.
Input**:** User selects a specific betting option (e.g., who will score next, next quarter score).
Description**:** The system will offer a dynamic list of in-game betting options based on current game events with each betting option possessing associated odds and a clear betting window for when the bet can be placed. The output will be a **c**onfirmation of the user's bet, potential payout, and an update in the user's account history.

Process:

1. Bets will be locked once the real-time game data passes the betting window (e.g., after a point is scored).
2. The betting system must confirm each placed bet with the user, including the potential payout, and log the bet in the user's account history.
3. A notification will inform the user if the betting window has closed or if the game data is unavailable.

Requirement #:14

Use Case #: 14
Introduction: Post-Match Analysis and Insights
Rationale: Provides users with detailed game breakdowns and performance insights after the game has concluded.
Input: User selects post-game analysis option after the match has ended.
Requirement Description: After a match concludes, the system will automatically generate detailed player and team performance stats, such as scoring efficiency, turnovers, and play-by-play breakdowns then, the analysis engine must provide visualizations, such as graphs, heat maps, and performance charts, for better insight into the game afterwards, the user should be able to click on specific events or players for in-depth details (e.g., comparison between player stats). Finally, the system will notify the user if post-match analysis is delayed or unavailable.

Output: Post-match insights, player performance stats, visual breakdowns, and key game highlights are displayed to the user.

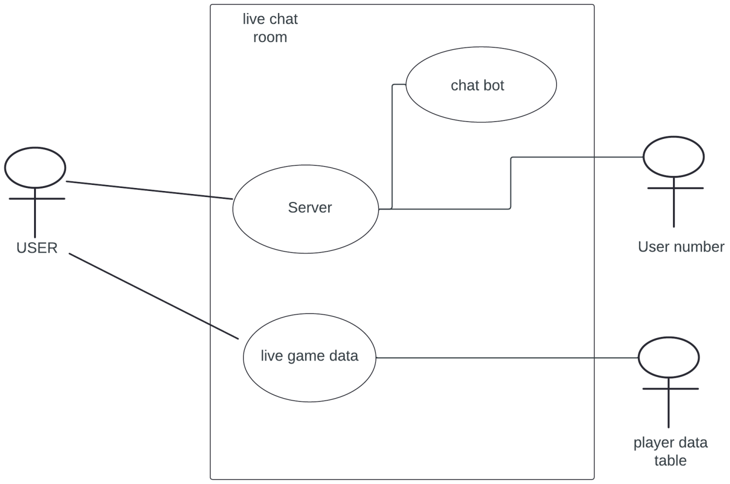
Requirement #: 15

Use Case #: 15
Introduction: Social Media Sharing Integration
Rationale: Allows users to share game highlights, predictions, or analysis directly to social media platforms.
Input: User selects a specific game moment, analysis, or prediction to share.
Requirement Description:

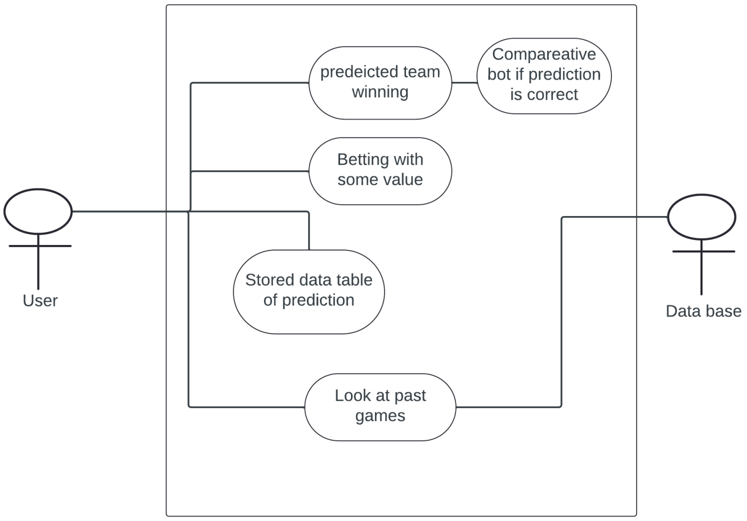
1. The system will provide a share option on key moments, predictions, and analysis, generating a shareable image or text snippet.
2. The system must integrate with the social media APIs (e.g., Twitter, Facebook) to enable direct sharing from the app.
3. If a social media account is not linked, the system will prompt the user to connect an account before sharing.
4. If sharing fails, the user should be notified, and the system will allow them to retry or save the content locally for manual sharing later.

Output: A confirmation of the shared content on the user's selected social media platform, or a prompt if sharing was unsuccessful.

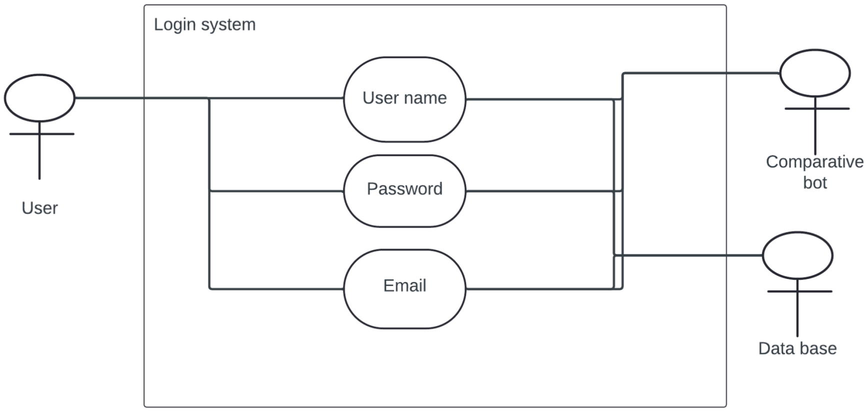
2.3 **Use case Diagrams**
Use case diagram #: 1
Use case #: 4



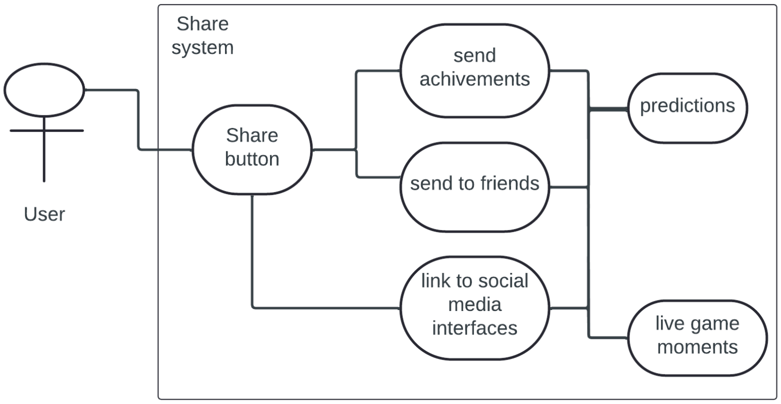
Use case diagram #:2
Use case #: 9



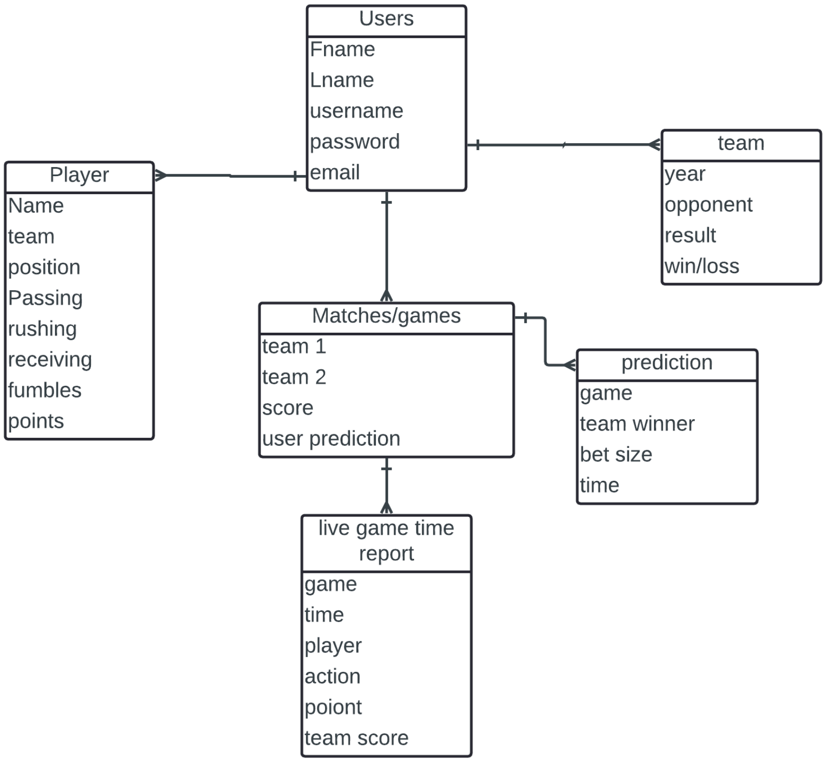
Use case diagram #: 3
Use case #: 7



Use case diagram #: 4
Use case #: 15



3.0 Database Tables



Users
Fname
Lname
username
password
email

team
year
opponent
result
win/loss

Player
Name
team
position
Passing
rushing
receiving
fumbles
points

Matches/games
team 1
team 2
score
user prediction

prediction
game
team winner
bet size
time

live game time report
game
time
player
action
poiont
team score

4.0 Class diagram

https://lucid.app/lucidchart/8c38011e-8823-4d40-a3df-7dfb500b5f28/edit?viewport_loc=-81%2C-181%2C2208%2C1088%2C0_0&invitationId=inv_6a436616-df87-4149-93ea-42fdb7012d16

classes

- live game
  - live game data
  - live game chat
  - live game play by play
- searching
  - search for players
  - search teams
- filter button
  - change and create data tables for user personal
  - add or sub columns of data table
- personal data
  - in search function
    - players/teams has option to be added to personal database for user ease of access
  - is like a check mark at end of it all
  - in home screen user can click and unfollow the player
- login
  - 2 input for user and password
  - 3rd input to ask for register
  - 4th input if forgotten password
- password forget
  - asks for email and send password used to email given
- register
  - 3 input for user, pass, and email
- live game match
  - hosts chat
    - has on/off function
  - hosts bet
  - hosts prediction
  - hosts play by play
  - hosts weather
  - hosts data of current game
- betting
  - user option of who wins
  - additional option by how many points for each team vs machine vs average user number

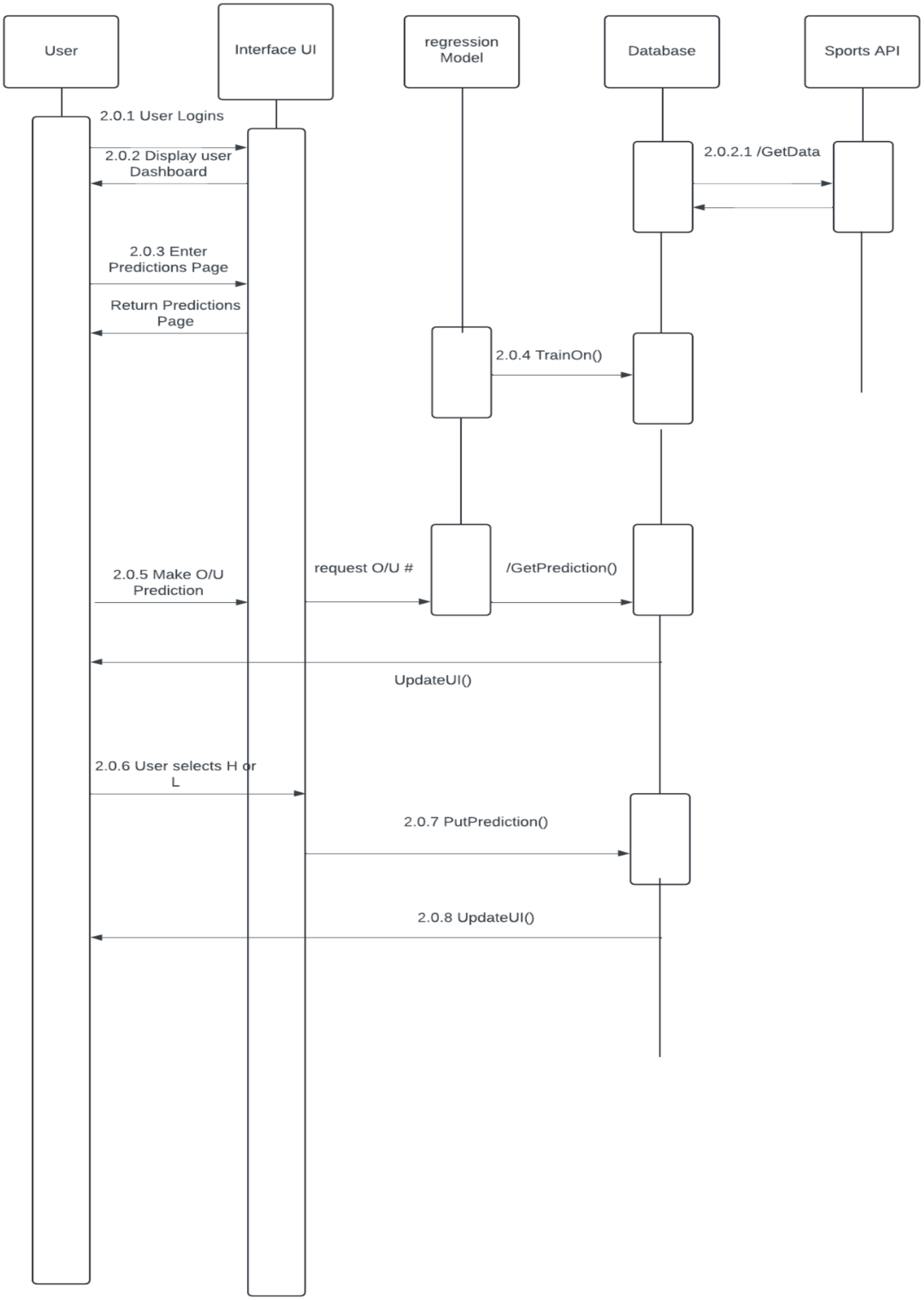
- prediction
  - user option on prediction who wins
  - additional option who makes play/ biggest play/ time/ yard amount
  - machine tells difference from actual game vs user
- weather live game
  - tells the weather of live game
  - stores it in data
- search feature
  - search using player as base
  - search using team as base
  - gives table results of using name and anything with letters in a row as results from table making new table
- filter feature
  - each table has a filter button
  - displays column options to add for user to only want too see
  - displays new table with new filter
  - new options redo this new table
- menu/home
  - displays trending
  - current live game
  - recent game if none has happened yet or going on

users personal added player/ teams to database for them too see

social media share button
idk where to put

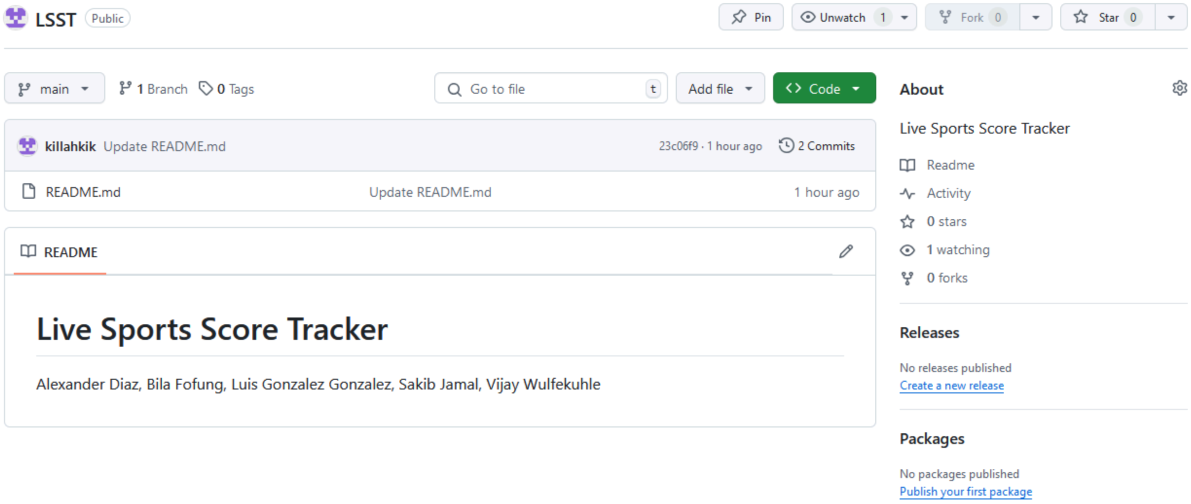
5.0 BEHAVIORAL MODELING

Sequence diagram for Use Case #2: User predicts Over/Under on select feature



9.0 GITHUB

https://github.com/killahkik/LSST



killahkik Update README.md 23c06f9 · 1 hour ago 2 Commits

README.md Update README.md 1 hour ago

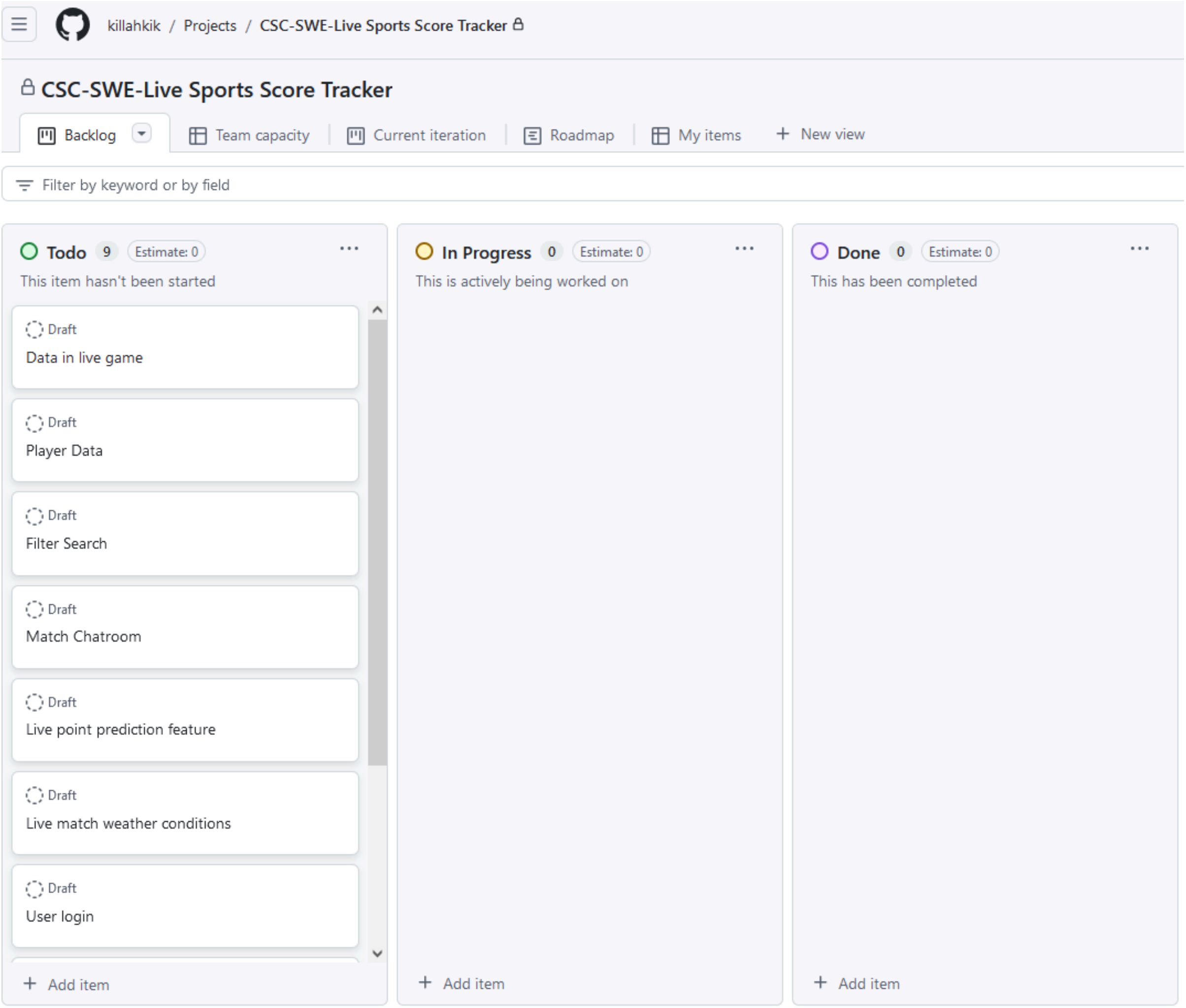
Live Sports Score Tracker

Alexander Diaz, Bila Fofung, Luis Gonzalez Gonzalez, Sakib Jamal, Vijay Wulfekuhle

About: Live Sports Score Tracker

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package


killahkik / Projects / CSC-SWE-Live Sports Score Tracker
Backlog
Team capacity
Current iteration
Roadmap
My items
New view
Filter by keyword or by field
Add item
Add item
Add item

CSC-SWE-Live Sports Score Tracker

Todo 9 Estimate: 0 — This item hasn't been started

- Draft — Data in live game
- Draft — Player Data
- Draft — Filter Search
- Draft — Match Chatroom
- Draft — Live point prediction feature
- Draft — Live match weather conditions
- Draft — User login

In Progress 0 Estimate: 0 — This is actively being worked on

Done 0 Estimate: 0 — This has been completed