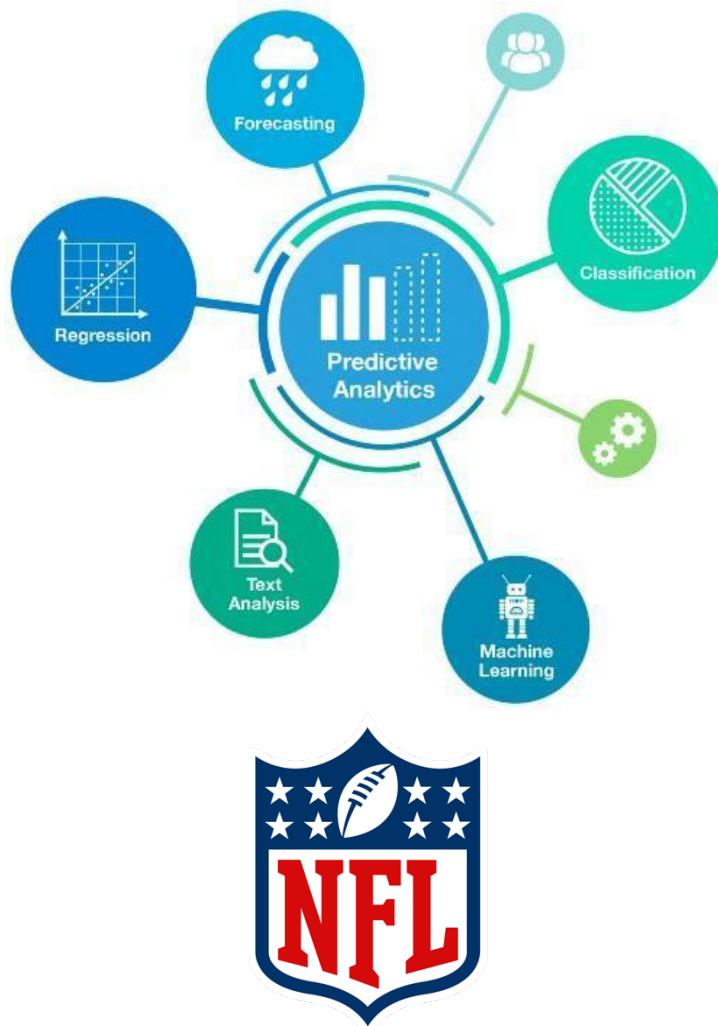


Sport Data Xchange

Group Project – Team Sport Data Xchange CIS 9590 – Section PTRA



Dohun Kim (dohun.kim@baruchmail.cuny.edu)
Hyuk Kim (hyuk.kim2@baruchmail.cuny.edu)
Shobhit Ratan (shobhit.ratan@baruchmail.cuny.edu)
Jason Subhani (Jason.subhani@baruchmail.cuny.edu)
Luke Weeks (Luke.weeks@baruchmail.cuny.edu)

Table of Contents

Phase 1: Problem definition object: Define your project	3
Project Summary	3
The System Setting	3
Describe the Project Environment	3
Systems history	3
The Systems Description	4
Principal Inputs and their forms	4
What's the process to handle data?	4
What's the Output?	4
Goals for the System	4
Data Gathering Plan	4
Phase 2: System Representation – Design Object: Design the Logical Model	6
Summary	6
Data Flow Diagrams	9
Menu Hierarchies	10
Report Design	11
Phase 3: Project & Systems Analysis	13
Project Analysis	13
Systems Analysis	14
Major Processes, Decision Tables, Decision Trees	15
File Descriptions	17
Phase 4: Build your System / Prototype	19
Summary	19
Phase 5: System Implementation	33
Executive Summary	33
Hardware / Software	33
Testing / Training	34
Implementation Plan	35
Post Launch / Support Plan	35
Future Modification Report	36
Appendix	37
A. MS Project Plan	37
B. Project Charter	37
C. Meeting Logs	38
D. Change Logs	43

E. Issues Risks Logs	44
F. Variance Report	44

Phase 1: Problem definition object: Define your project

Project Summary

US sports betting industry is expected to grow to \$8 billion within five years. Our Sport Data Xchange application can be used as a tool for sports betting and Sport Data Xchange application will be a leading analytical website that can predict upcoming football games outcomes as well as outcomes for theoretical, unplayed games. In this deliverable, our project team will delve into how the system is built, how we gathered and managed business requirements, how to effectively manage the project as a team, and ultimately build an analytical website for our users that provide the intended business value.

The System Setting

Our Sports Data Xchange application will be a web application that users can easily use the “Sport Data Xchange” website where users are able to have future theoretical results of the game by simply selecting teams for the game. We intend to use historical data to help predict NFL football games. Within the statistics of wins & losses between 2015 to 2019, the Sport Data Xchange allows users to predict the outcome of the NFL games. Key factors in the dataset are utilized to analyze the game of football such as the result of win & loss, the home & away team, the total number of team rush yards etc, and also key statistical factors are identified to help determine the winners of the football games over time, allowing users to choose theoretical marches and future game prediction.

Describe the Project Environment

Systems history

There are plenty of web applications out on the market that use sports analytics tools, so this is a proven concept that has been done before. The distinction of SDX is that it will be fully implemented on the Ruby on Rails Framework.

System Users and Functional Responsibilities

1. For Sport Data Xchange Admins
 - Able to manage users' activity, request, approval, delect.
 - Able to input/update/delete data to utilize for statistic
2. For Sport Data Xchange Users
 - Profile setup
 - Open new account

- Add/edit personal information
- Update payment details
- Search team
 - Select own preferred team

The Systems Description

Principal Inputs and their forms

Users will create accounts to access our web services. Since the primary object of Sport Data Xchange is to predict outcomes of football games the user asks for, the user inputs teams competing, and the website will output the winner.

What's the process to handle data?

The user inputs teams competing, and the website will output the winner based on the calculated result into a data analytics tool of the website.

Database will primarily store information and sports information (team name, wins, etc.) for data analysis tools.

What's the Output?

The output would be a report detailing the winner team, along with relevant statistics.

Goals for the System

1. Provide the best user experience
 - Simple UX/UI ease to navigate
 - the most straightforward sports prediction experience
2. Become a standard for sports betting tools
 - Freemium platform to build a growing user base
 - Create an efficient web app that is responsive and delivers quick results
3. High accuracy in our predictions
 - Use up to date information on teams and players for game prediction throughout the season

Data Gathering Plan

A majority of the data handled by the prototype will be the football statistics that the user shall access. For the prototyping phase, the team will choose the most cost-effective gatherings

with data scraped from the NFL website as well as NFLSavant.com since the .csv files are already available. Generally, the scope of data is only NFL games which have occurred in the past five years. For larger scale implementation and more accurate results, we would need significantly more data that dates back to the earliest known archives of play by play data of the NFL. Generally, this type of data would need to be purchased from reputable third parties, as searching for the data could lead to inconsistency in data cleanliness and uniformity.

Freely acquired data tend to have missing fields and inconsistencies, but for the sake of prototyping and having a proof of concept, we can still make it work. The prototype is meant to be able to handle large volumes of sports data that the user can view. User data can provide a lot for usage analytics in the future, but right now the user data we gathered are made up data by the developers to test the site. When the website is fully launched, the user data gathering process will be minimal, mostly store name, addresses, and payment information. Passwords and credit card information will be encrypted in the Rails framework and can only be known by the user.

Phase 2: System Representation – Design Object: Design the Logical Model

Summary

Our system representation can be broken down into six areas.

1. Registration

1.1. Open new account

- 1.1.1. The system will allow the user to sign up for the website.
- 1.1.2. The system shall require that user create a user ID
- 1.1.3. The system shall require that the user id is in a valid email address format.
- 1.1.4. The system shall require that the password is at least 8 characters.
- 1.1.5. They system shall require that the password contain a number, alphanumeric such as \$, @, &, etc.
- 1.1.6. The system shall require the new applicant to select 3 security questions and provide an answer for them
- 1.1.7. The system shall verify and authenticate the new applicant's email.

1.2. User Logs In

- 1.2.1. The system will require that the User logs in with a valid User ID and Password.
- 1.2.2. The system shall display login error if user credentials do not match.
- 1.2.3. The system shall contain a "Forgot User ID" module.
- 1.2.4. The system shall contain a "Forgot Password" module.

1.3. Add Personal Details

- 1.3.1. The system shall require the new applicant to enter all necessary fields

1.4. Add Payment Details

- 1.4.1. The system shall require the new applicant to enter all necessary fields.

1.5. Notify User

- 1.5.1. They system shall notify new applicants that they have successfully completed the registration process and their approval as a registered user (member) is under review.
- 1.5.2. The system shall notify 'Super Admin' and 'Member Approver' when a new applicant applies to join Sport Data Xchange as a registered user (member).
- 1.5.3. The system shall notify new applicants when the payment for subscription fee is processed for premium members.
- 1.5.4. The system shall notify new applicants on approval as registered user of Sport Data Xchange.
- 1.5.5. The system shall notify new applicants and registered premium users whenever their payments for subscription fee could not be processed by Sport Data Xchange.

- 1.5.6. The system shall notify new applicants when their application to join Sport Data Xchange as a registered user is denied.
- 1.5.7. The system shall notify registered premium users two weeks before the monthly subscription expiration date.

1.6. Update Personal Details

- 1.6.1. The system will allow registered users to update personal details.
- 1.6.2. The system shall also allow the new applicants to update their personal details.

1.7. Update Payment Details

- 1.7.1. The system will allow registered users to update payment details.
- 1.7.2. The system shall allow the registered users to update payment details.

2. Payment

2.1. Process Transaction

- 2.1.1. The system shall send a request to validate the credit card.
- 2.1.2. The system shall process the transaction after validating the credit card by 1st date of each month.

2.2. Confirm Transaction

- 2.2.1. The system shall send confirmation to registered users once their monthly subscription payment is processed.

2.3. Cancel / Renew

- 2.3.1. The system shall allow the registered premium users to cancel their subscription anytime.
- 2.3.2. The system allows the registered premium users to renew the subscription – manually or automatically anytime. No refunds will be given for the subscription fee for the month in which the membership is cancelled.

3. Premium

3.1. User Signs up

- 3.1.1. The system will allow the user to signup for the Premium account.
- 3.1.2. The system will require that the User logs in with a valid User ID and Password.
- 3.1.3. The system shall take the user to a payment processing page.

3.2. [Payment]

- 3.2.1. The system shall follow the processes highlighted in Functional Area 2.0.

3.3. Notify User

- 3.3.1. The system shall notify new applicant that they have successfully completed the registration process for Premium membership and their approval as a registered Premium User (member) is under review.
- 3.3.2. The system shall notify ‘Super Admin’ and ‘Member Approver’ when a new applicant applies to join Sport Data Xchange as a registered Premium User.

- 3.3.3. The system shall notify new applicant whenever the payment for subscription fee is processed for premium members.
- 3.3.4. The system shall notify new applicant on approval as registered user of Sport Data Xchange
- 3.3.5. The system shall notify new applicants and registered premium users whenever their payments for subscription fee could not be processed by Sport Data Xchange.
- 3.3.6. The system shall notify new applicants when their application to join Sport Data Xchange as a registered user is denied.
- 3.3.7. The system shall notify registered premium users two weeks before the monthly subscription expiration date.

3.4. Cancel / Renew

- 3.4.1. The system shall allow the registered premium users to cancel their subscription anytime.
- 3.4.2. The system shall allow the registered premium users to renew the subscription – manually or automatically anytime. No refunds will be given for the subscription fee for the month in which the membership is cancelled.

4. Theoretical Game prediction

4.1. User inputs teams to compete

- 4.1.1. The system shall allow users to input teams that they want to compete against each other.
- 4.1.2. The system shall require users to enter all necessary fields.

4.2. System implements User Input into Theoretical Win System

- 4.2.1. The system shall process the User Input.
- 4.2.2. The system shall send confirmation to Users once their User Input is processed.

4.3. [Theoretical Win Data Analytics System]

- 4.3.1. The system shall implement the results in the Theoretical Win Data Analytics System.

4.4. System collects results from data analysis

- 4.4.1. The system shall collect the results from the Theoretical Win Data Analytics System.
- 4.4.2. The system shall output the results from the Theoretical Win Data Analytics System.

4.5. Generate Report

- 4.5.1. The system shall generate a report of the top indicators and key results from the data analysis.
- 4.5.2. The system shall output key data visualizations based on the key performance indicators.

5. Future Game Prediction

5.1. User inputs information of future games

5.1.1. The system shall allow users to input teams that they want to compete against each other.

5.1.2. The system shall require users to enter all necessary fields.

5.2. System implements User Input into Future Win Systems

5.2.1. The system shall process the User Input.

5.2.2. The system shall send confirmation to Users once their User Input is processed.

5.3. [Future Win Data Analytics System]

5.3.1. The system shall implement the results in the Future Win Data Analytics System.

5.4. System collects results from data analysis

5.4.1. The system shall collect the results from the Future Win Data Analytics System.

5.4.2. The system shall output the results from the Future Win Data Analytics System.

5.5. Generate Report

5.5.1. The system shall generate a report of the top indicators and key results from the data analysis.

5.5.2. The system shall output key data visualizations based on the key performance indicators.

6. Report

6.1. System inputs information into data analytics system

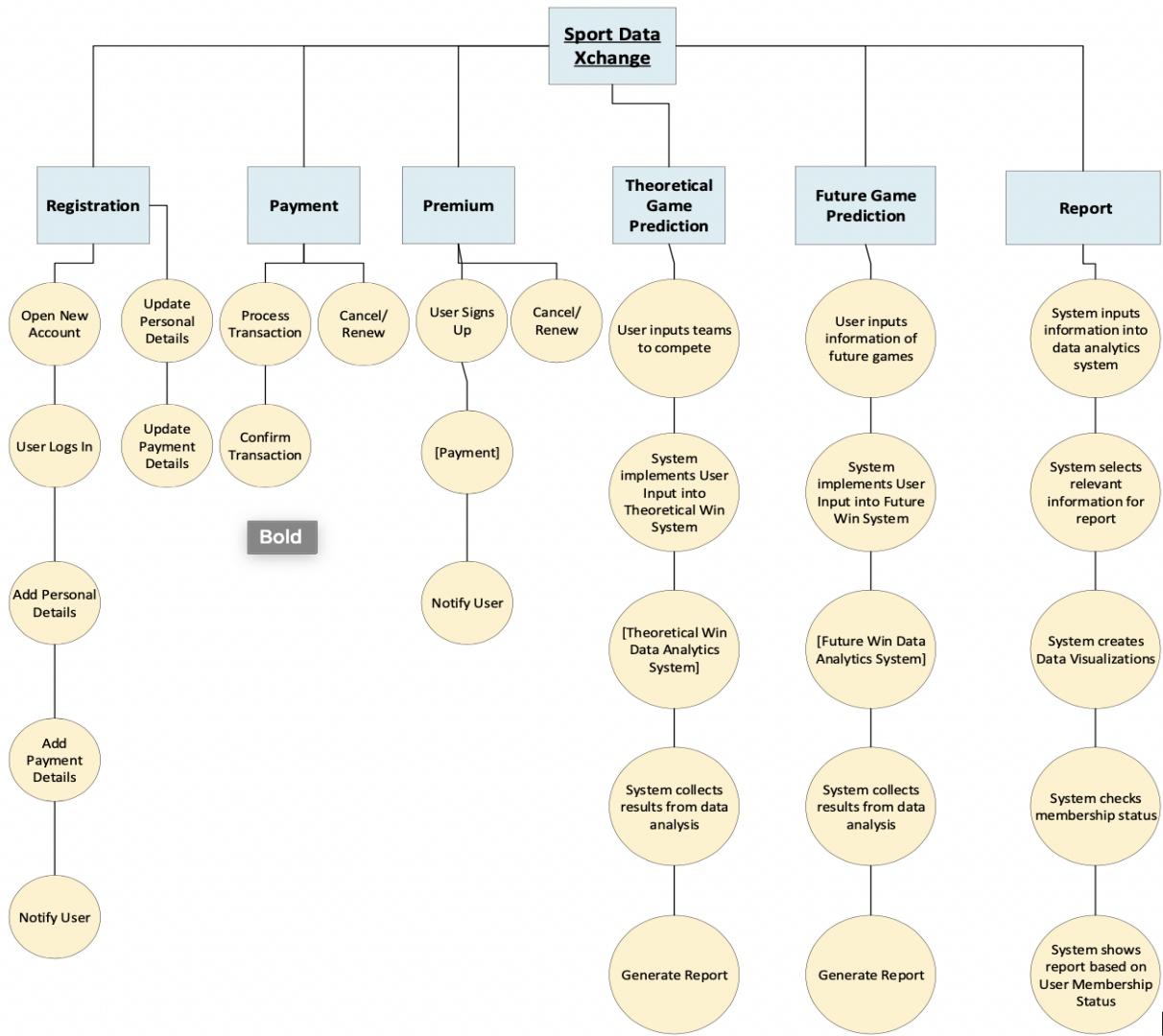
6.2. System selects relevant information for report

6.3. System creates Data Visualizations

6.4. System checks membership status

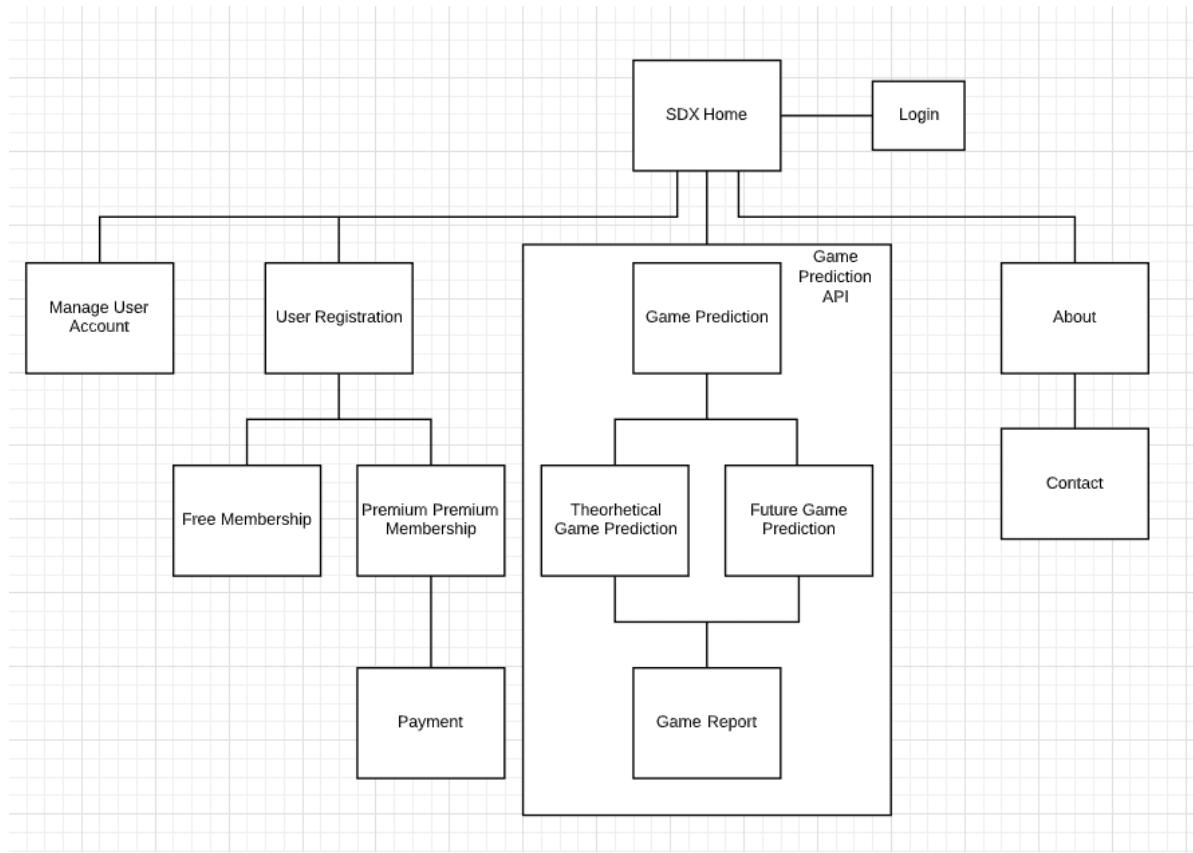
6.5. System shows report based on User Membership Status

Data Flow Diagrams



Menu Hierarchies

On the home page, users can immediately sign up or use the game predictor if they are signed in. Game report and other sports data stays within the Game Prediction API.



Report Design

- 1. Based on user's team selection, the report displays**
 - 1.1. Who will more likely to win
 - 1.2. Key determining metrics
- 2. Users can view detailed team statistics comparison such as time of possession of a ball, passing yards, rushing yards, penalties, etc.**

Buffalo Bills vs. Cincinnati Bengals

Winner: Buffalo Bills

Loser: Cincinnati Bengals

Game Time: 2019-09-22 00:00:00 UTC

Game stats are as follows

Team Name	Score	Time of Possession	Passing Yards	Rushing Yards	Total Turnovers	Penalties
Buffalo Bills	21	36:54	241.0	175.0	1	7
Cincinnati Bengals	17	23:06	240.0	66.0	2	2

Based on this game's analysis, we have seen that the winning teams on average have higher passing yards and higher rushing yards than the losing team.

Time of possession is a significant factor in completely one sided games but has a reduced impact in even contests.

[Back](#)

Copyright © SportsDataXchange. All rights reserved.

Phase 3: Project & Systems Analysis

Project Analysis

To estimate our resource analysis, we used a parametric estimate. The estimated cost of the initial building project is \$74,688. The cost of programmer resources (Python, Ruby, SQL) are determined by the average hourly rate and the estimated hours to engage in the initial building project (*please refer to Table 1*). Additionally, we plan to spend \$75,000 on marketing. This is because unlike a traditional marketing strategy, our goal is to build a groundswell of support for free marketing platforms such as social media through multiple platforms where track record matters.

To estimate the cost of ongoing engineering per month for 3 years, we applied the parametric estimate again. First, we multiplied each programmer's average hourly rate by the number of required hours per week. As our product grows, the increase in weekly working hours for each programming language will result in the total ongoing engineering costs per month (*please refer to Table 2*).

Budget Summary				
Initial Build Estimate	Hours Required	Cost Per Hour	Total Cost for initial Build	
Python	250	\$75.00	\$18,750.00	
Ruby	500	\$97.50	\$48,750.00	
SQL	200	\$35.84	\$7,168.00	
			\$74,668.00	
Total Initial Costs Medium				
Engineering	\$100,000		Ongoing Engineering Costs Per Month	
Marketing	\$75,000			
Ongoing Engineering Costs Per Month				
Year 1	\$56,668.00		Year 1 \$56,668.00	
Year 2	\$70,835.00		Year 2 \$70,835.00	
Year 3	\$77,918.50		Year 3 \$77,918.50	
Total Initial Costs Low				
Engineering	\$66,667			
Marketing	\$50,000			
Total Initial Costs Medium				
Engineering	\$100,000			
Marketing	\$75,000			
Total Initial Costs High				
Engineering	\$133,333			
Marketing	\$100,000			

(Table 1)

Budget Summary							
Ongoing Programming	Rate/Hour	Hours/Week Year 1	Cost Year 1	Hours/Week Year 2	Cost Year 2	Hours/Week Year 3	Cost Year 3
Python	\$75.00	100	\$7,500.00	125	\$9,375.00	137.5	\$10,312.50
Ruby	\$97.50	50	\$4,875.00	62.5	\$6,093.75	68.75	\$6,703.13
SQL	\$35.84	50	\$1,792.00	62.5	\$2,240.00	68.75	\$2,464.00
		Totals/Week	\$14,167.00		\$17,708.75		\$19,479.63
		Totals/Year	\$736,684.00		\$920,855.00		\$1,012,940.50
Ongoing Engineering Costs Per Month							
Year 1							\$56,668.00
Year 2							\$70,835.00
Year 3							\$77,918.50
Marketing	Rate/Month	Cost Year 1					
Employee - Social Digital	\$2,000.00		\$24,000.00				
Ad Costs - TBD			\$50,000.00				
Website Hosting	Rate/Month	Size					
Data Hosting	\$8.00						
Cloud Costs	\$23.71						
Outsource Support	Low \$8 hr 24/7	High \$15 hr 24/7					
Yearly Support Costs	\$70,000		\$130,000				

estimating cost of labor for UIUX program
 *All depends on the App's complexity. A basic UI/UX design of the apps may take a few working days. but Applications with complex UI like Uber or Instagram may take up to 400+ hours for UI/UX design.

(Table 2)

Systems Analysis

There are use case diagrams for three functional areas that display the interaction between the system and different users. All use case diagrams are traceable back to BRD.

1. Registration (BRD 1.0)

- 1.1. System handles user information and account creation
- 1.2. System allows users to log into their accounts

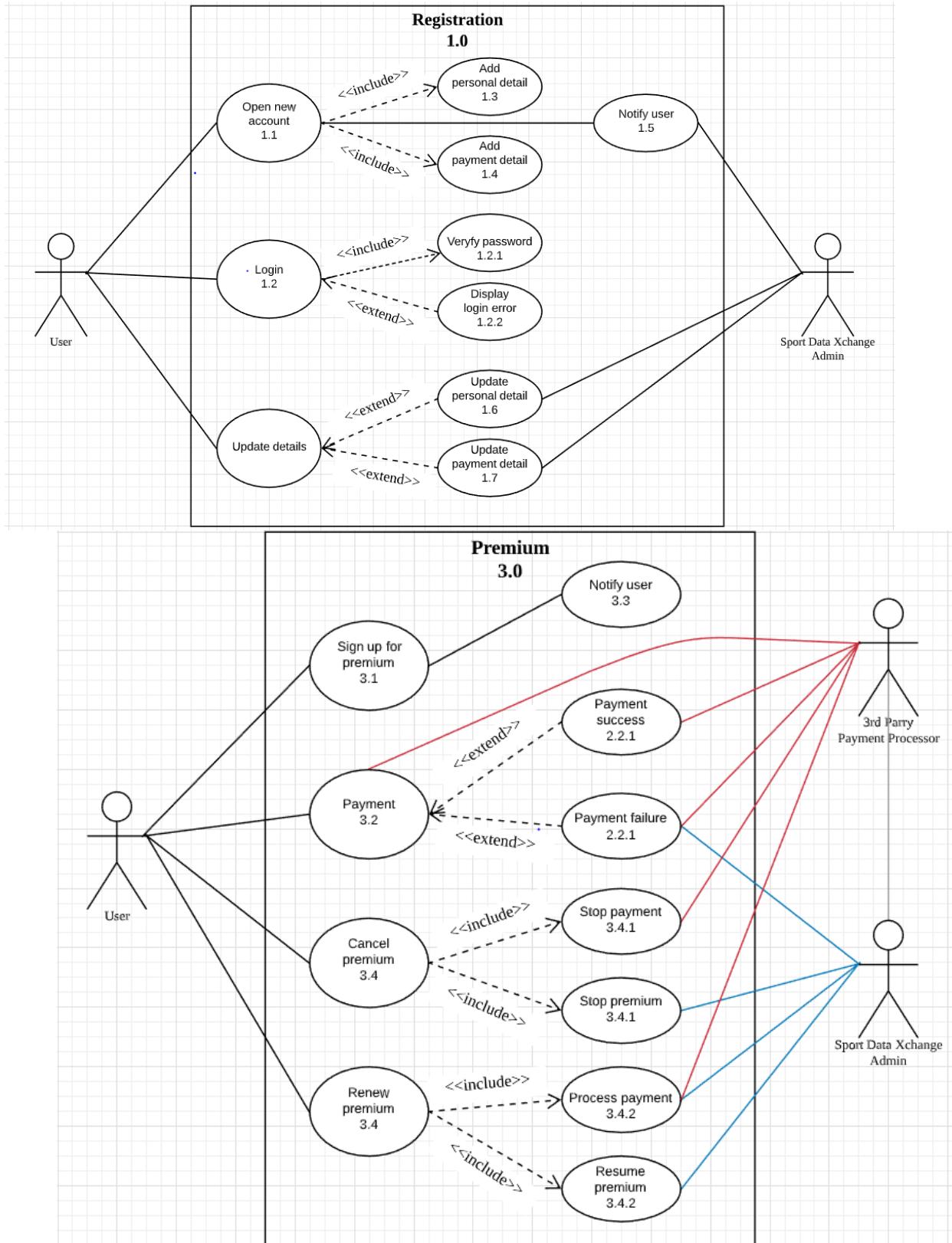
2. Theoretical game prediction (BRD 4.0)

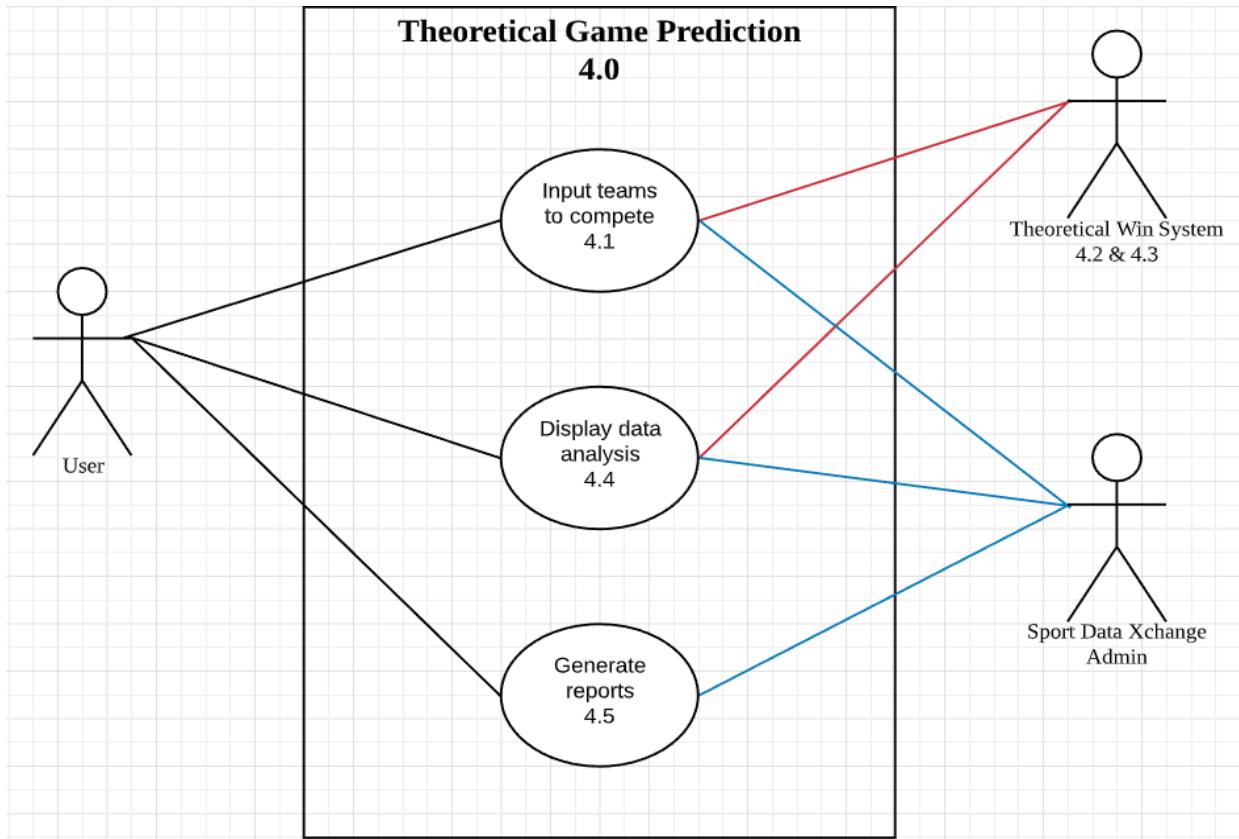
- 2.1. Database handles user info and sport statistics
- 2.2. Back-end calculates output based on user input and current statistics
- 2.3. Administrations oversee user operation and updates of database

3. Premium (BRD 3.0)

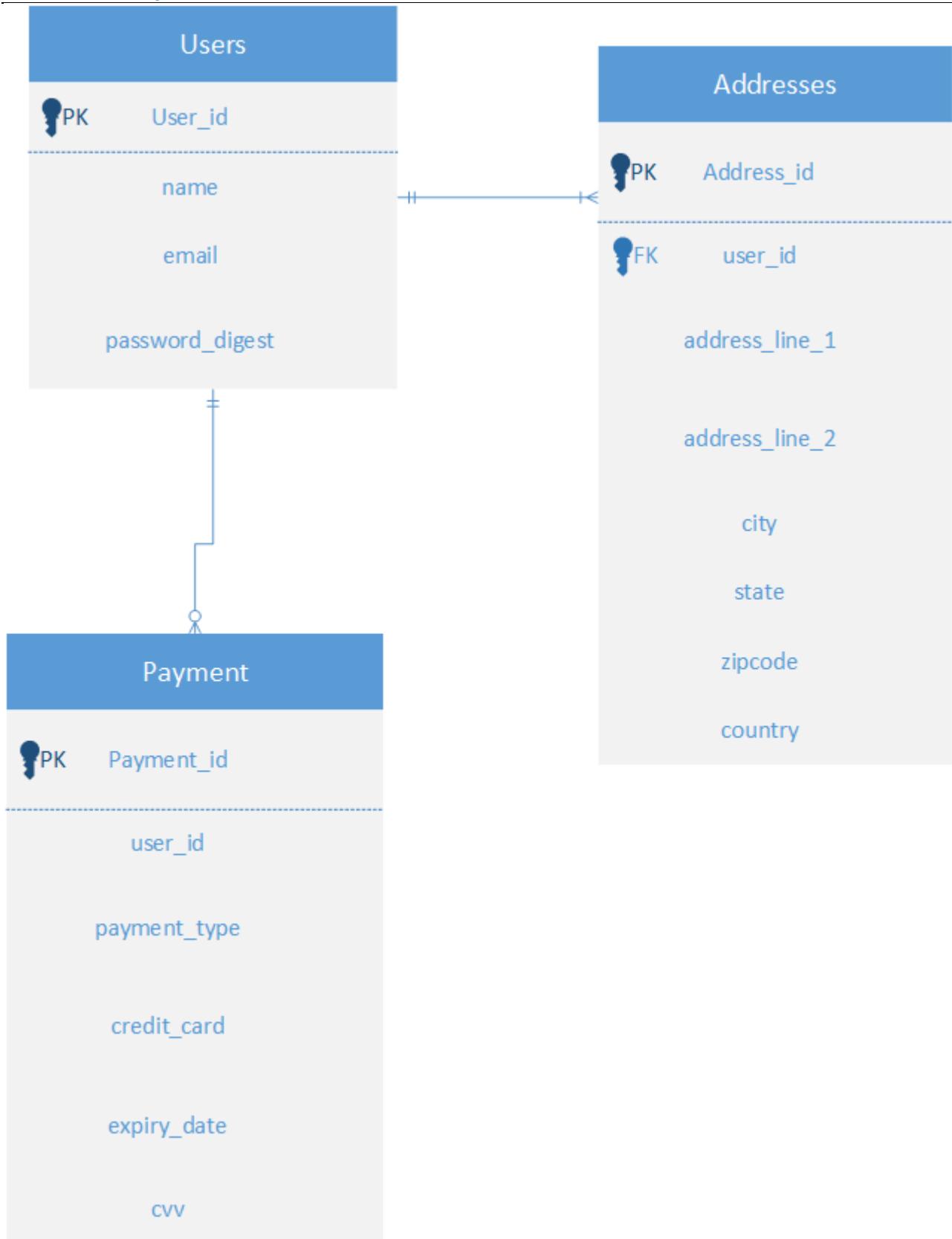
- 3.1. System works with third party payment processor to handle financial transactions between the user and the website
- 3.2. System allows users to pay, renew, and cancel their premium membership
- 3.3. System assigns administrators to keep an eye on the financial activity and check for discrepancies

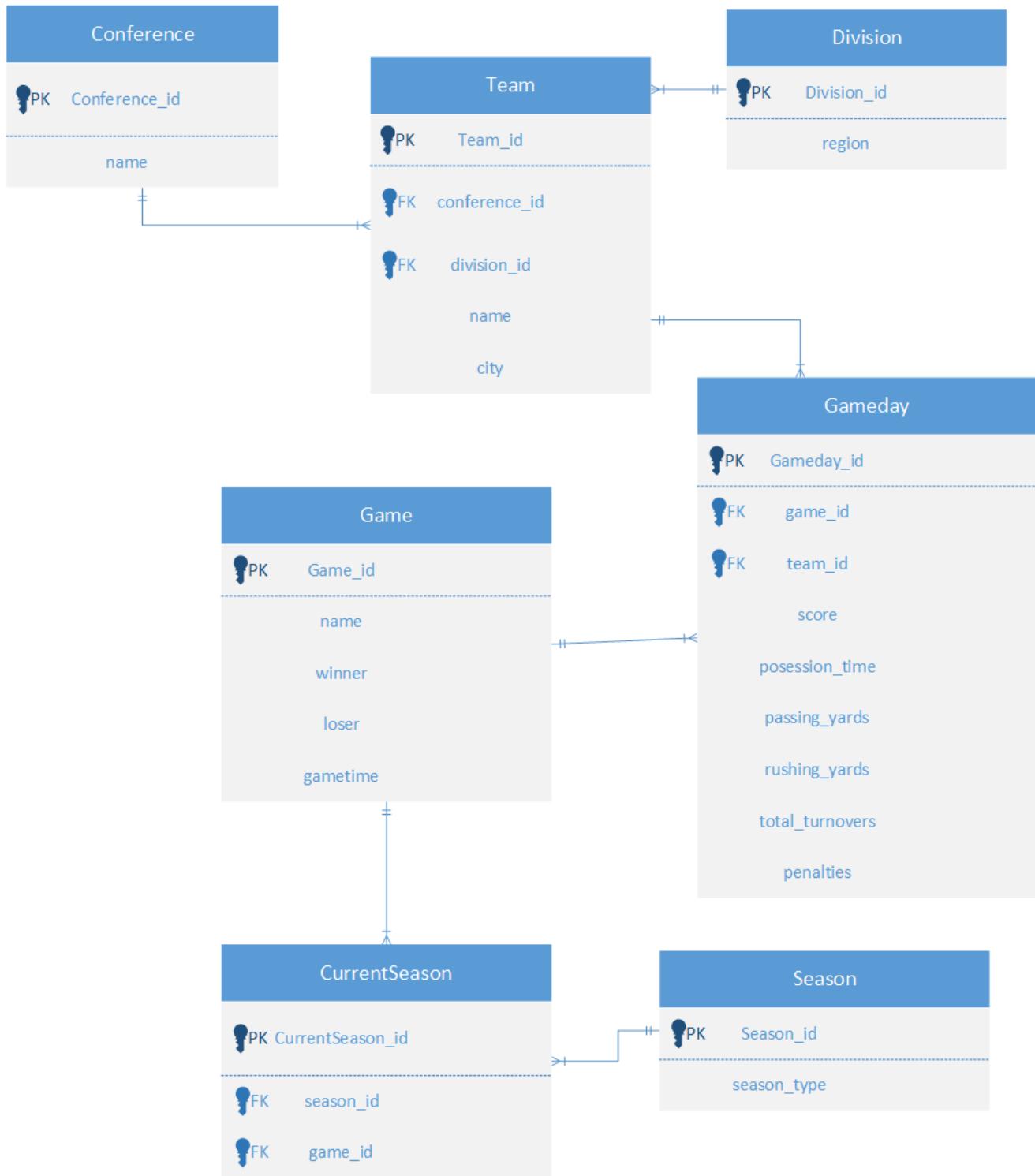
Major Processes, Decision Tables, Decision Trees





File Descriptions





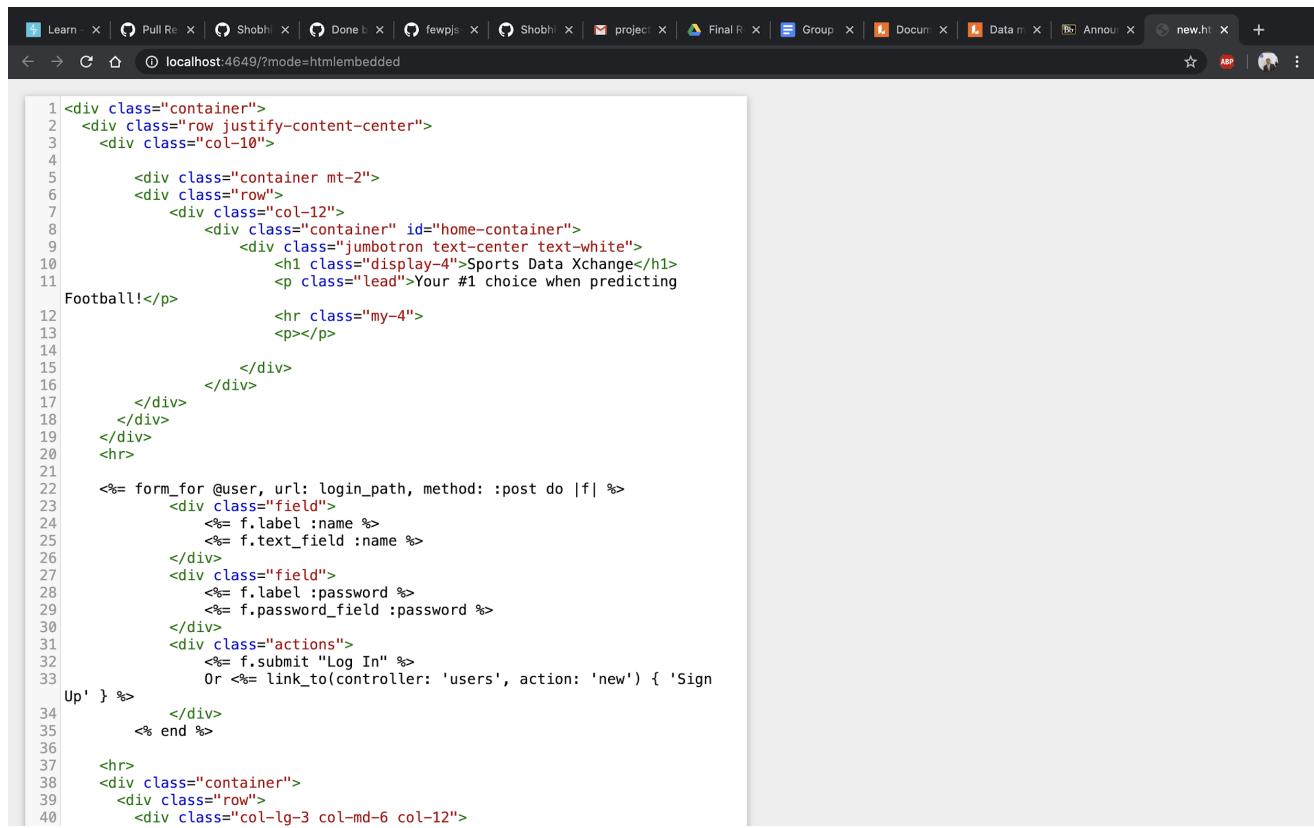
Phase 4: Build your System / Prototype

Summary

Our Github repository is as follows: <https://github.com/ShobhitRatan/sdx>.

In order to view the site in the browser please follow the following steps:

1. Fork and clone the repository.
2. Open the repository in an external code editor.
3. Run bundle install in the terminal to install the ruby gems.
4. Please run rails db:migrate and rails db:seed to load the database.
5. Please type rails s in the terminal to fire up the rails server and view our site in a browser of your choice.



The screenshot shows a browser window with the URL http://localhost:4649/?mode=html_embedded. The page displays the source code of a Ruby on Rails view file. The code is written in HTML and includes some Ruby syntax for form handling. The content of the page is a simple login form with fields for name and password, and a submit button labeled "Log In". Below the form, there is a link to "Sign Up". The code is numbered from 1 to 40 on the left side.

```
1 <div class="container">
2   <div class="row justify-content-center">
3     <div class="col-10">
4
5       <div class="container mt-2">
6         <div class="row">
7           <div class="col-12">
8             <div class="container" id="home-container">
9               <div class="jumbotron text-center text-white">
10                 <h1 class="display-4">Sports Data Xchange</h1>
11                 <p class="lead">Your #1 choice when predicting
12                   Football!</p>
13                 <hr class="my-4">
14                 <p></p>
15
16               </div>
17             </div>
18           </div>
19         </div>
20         <hr>
21
22         <%= form_for @user, url: login_path, method: :post do |f| %>
23           <div class="field">
24             <%= f.label :name %>
25             <%= f.text_field :name %>
26           </div>
27           <div class="field">
28             <%= f.label :password %>
29             <%= f.password_field :password %>
30           </div>
31           <div class="actions">
32             <%= f.submit "Log In" %>
33             Or <%= link_to(controller: 'users', action: 'new') { 'Sign
34             Up' } %>
35           </div>
36         <% end %>
37
38         <hr>
39         <div class="container">
40           <div class="row">
41             <div class="col-lg-3 col-md-6 col-12">
```

```
41      <div class="col-lg-3 col-md-6 col-12">
42        <h2> Step 1:</h2>
43        <h4> Choose what you want to calculate</h4>
44        <p>Choose to predict an upcoming game or feed your curiosuty with
45          a fantasy match</p>
46        </div>
47      <div class="col-lg-3 col-md-6 col-12">
48        <h2> Step 2:</h2>
49        <h4> Choose your teams</h4>
50        <p>By default, game predictor gives you dates to choose.</p>
51        </div>
52      <div class="col-lg-3 col-md-6 col-12">
53        <h2> Step 3:</h2>
54        <h4> Predict!</h4>
55        <p>Once you picked all your settings, click predict and see the
56          outcomes. Wait a bit while our tool calculates.</p>
57        </div>
58      <div class="col-lg-3 col-md-6 col-12">
59        <h2>Step 4:</h2>
60        <h4> See the results</h4>
61        <p>A full report of the game is provided, with visuals.</p>
62      </div>
63    <hr>
64  </div>
65  <div class="container">
66    <div class="row">
67      <div class="text-center col-md-6 col-12">
68        <h3>Join for Free</h3>
69        <p>Get access to using our "Fantasy Match" program, where you can
70          compare the performance of teams in hypothetical games.</p>
71        <p><%= link_to "Sign Up", new_user_path %></p>
72      </div>
73      <div class="text-center col-md-6 col-12">
74        <h3>Go Premium</h3>
75        <p>Get full access to our tool to predict this season's games. Be
76          the talk of your friends with the ability to predict next week's game.</p>
77        <p><%= link_to "$14.99/mo.", payments_path %></p>
78      </div>
79    </div>
80  </div>
81  <hr>
```

```
71    </div>
72  <div class="text-center col-md-6 col-12">
73    <h3>Go Premium</h3>
74    <p>Get full access to our tool to predict this season's games. Be
75      the talk of your friends with the ability to predict next week's game.</p>
76    <p><%= link_to "$14.99/mo.", payments_path %></p>
77  </div>
78  <hr>
79  <div class="container">
80    <div class="row">
81      <div class="col-lg-4 col-md-6 col-sm-6 mb-md-0 mb-2">
82        <h2>Contact Us</h2>
83        <address>
84          <strong>Sports Data Xchange.</strong><br>
85          Zicklin School of Business,<br>
86          New York, New York, US<br>
87          <abbr title="Phone">P:</abbr> (123) 456-7890
88        </address>
89      </div>
90    </div>
91    <div class="col-lg-8 col-12">
92      <h2>About Us</h2>
93      <p>Sports Data Xchange began as a project in January 2020 to answer
94        the age old question: who are you putting your money on?</p>
95      <p>Sports Data Exchange consists of ....</p>
96    </div>
97  </div>
98  <hr>
99
100 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
101 <script src="js/jquery-3.4.1.min.js"></script>
102 <!-- Include all compiled plugins (below), or include individual files as
103     needed -->
104 <script src="js/popper.min.js"></script>
105 <script src="js/bootstrap-4.4.1.js"></script>
106
107 </div>
108 </div>
109 </div>
```

HTML and Embedded Ruby Code behind the home/login page.

SDX

Home About Account Theoretical Game Prediction Statistics

Sports Data Xchange

Your #1 choice when predicting Football!

Name

Password

[Log In](#) Or [Sign Up](#)

Step 1:
Choose what you want to calculate

Choose to predict an upcoming game or feed your curiosity with a fantasy match

Step 2:
Choose your teams

By default, game predictor gives you dates to choose.

Step 3:
Predict!

Once you picked all your settings, click predict and see the outcomes. Wait a bit while our tool calculates.

Step 4:
See the results

A full report of the game is provided, with visuals.

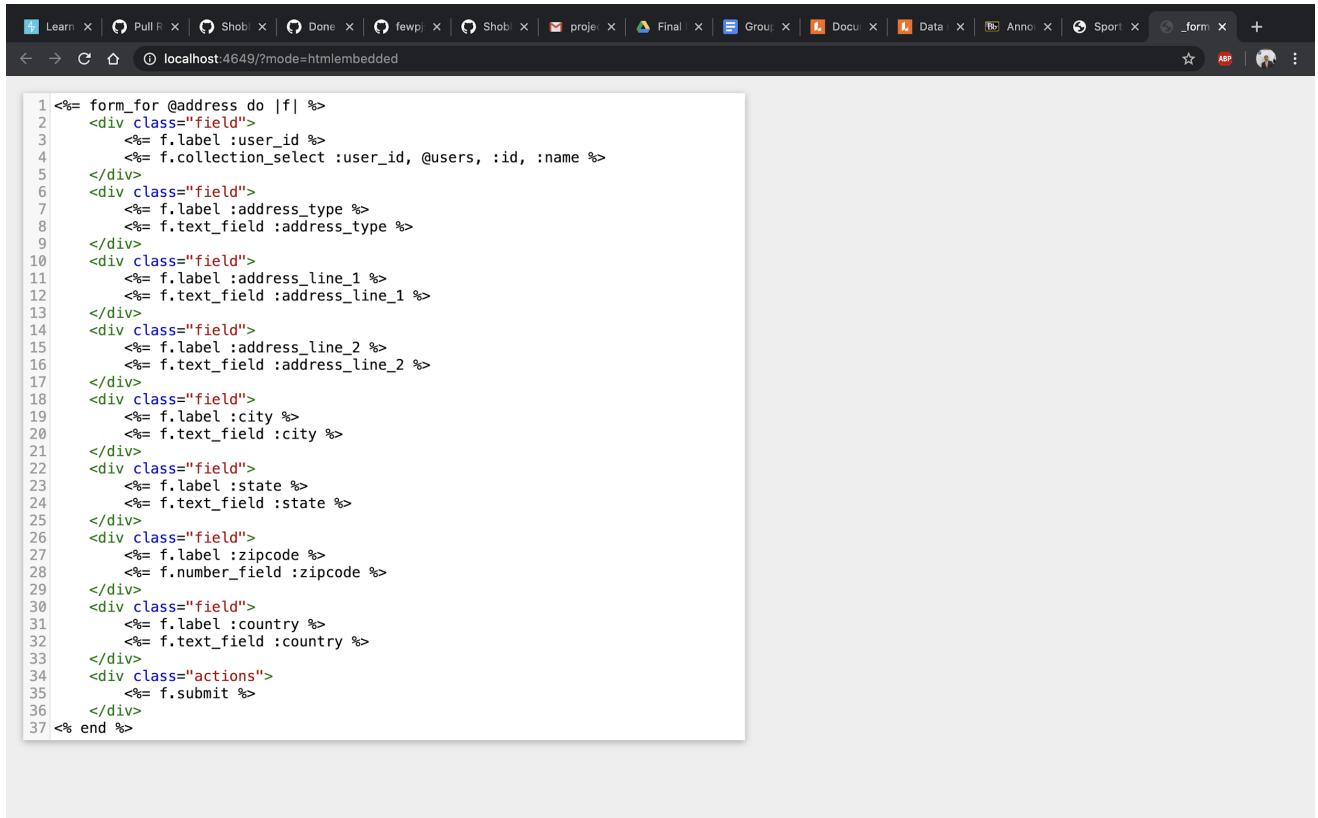
Login Page

```
DEBUG CONSOLE OUTPUT TERMINAL PROBLEMS
1: ruby

Processing by SessionsController#create as HTML
Parameters: {"authenticity_token"=>"PfkuyCZqEFFw2xuKPt6WUH2/jWme42P8SiBJYImAMx7lq1json0zj0ps9x7r/n15VBouJ4GAz6px5V6x0Nddw==", "user"=>{"name"=>"Barry Allen", "password"=>"[FILTERED]", "commit"=>"Log In"}
User Load (0.7ms)  SELECT "users".* FROM "users" WHERE "users"."id" IS NULL LIMIT ?  [["LIMIT", 1]]
↳ app/controllers/application_controller.rb:6:in `current_user'
User Load (0.4ms)  SELECT "users".* FROM "users" WHERE "users"."name" = ? LIMIT ?  [["name", "Barry Allen"], ["LIMIT", 1]]
↳ app/controllers/sessions_controller.rb:6:in `create'
Redirected to http://localhost:3000/
Completed 302 Found in 218ms (ActiveRecord: 1.1ms | Allocations: 1878)

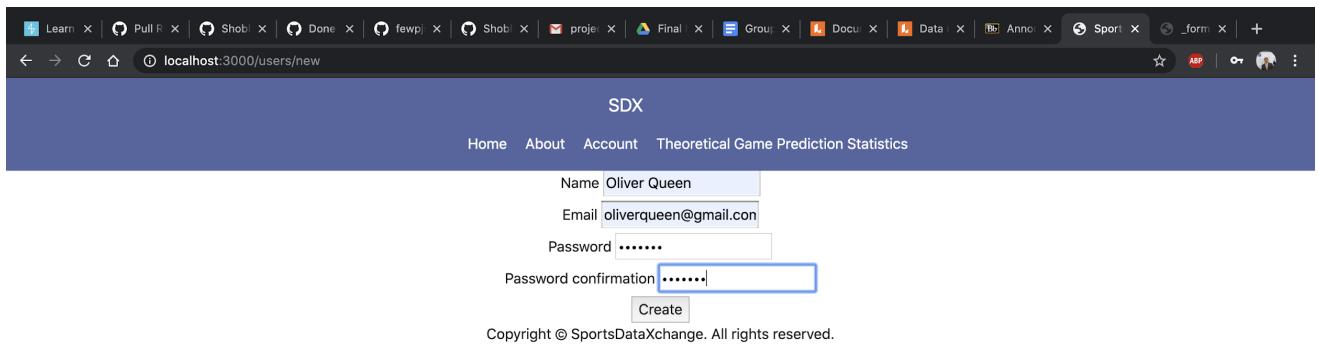
Started GET "/" for ::1 at 2020-05-14 04:32:57 -0400
Processing by WelcomeController#home as HTML
```

SQL Query when User logs in to the system.



```
1 <%= form_for @address do |f| %>
2   <div class="field">
3     <%= f.label :user_id %>
4     <%= f.collection_select :user_id, @users, :id, :name %>
5   </div>
6   <div class="field">
7     <%= f.label :address_type %>
8     <%= f.text_field :address_type %>
9   </div>
10  <div class="field">
11    <%= f.label :address_line_1 %>
12    <%= f.text_field :address_line_1 %>
13  </div>
14  <div class="field">
15    <%= f.label :address_line_2 %>
16    <%= f.text_field :address_line_2 %>
17  </div>
18  <div class="field">
19    <%= f.label :city %>
20    <%= f.text_field :city %>
21  </div>
22  <div class="field">
23    <%= f.label :state %>
24    <%= f.text_field :state %>
25  </div>
26  <div class="field">
27    <%= f.label :zipcode %>
28    <%= f.number_field :zipcode %>
29  </div>
30  <div class="field">
31    <%= f.label :country %>
32    <%= f.text_field :country %>
33  </div>
34  <div class="actions">
35    <%= f.submit %>
36  </div>
37 <% end %>
```

Embedded Ruby and HTML code behind New and Edit User forms.



The screenshot shows a user creation form titled "SDX". The URL in the address bar is "localhost:3000/users/new". The form fields are as follows:

- Name: Oliver Queen
- Email: oliverqueen@gmail.com
- Password: (redacted)
- Password confirmation: (redacted)

A "Create" button is located below the password fields. At the bottom of the page, there is a copyright notice: "Copyright © SportsDataExchange. All rights reserved."

New User creation in progress.

```

EXPLORER          DEBUG CONSOLE      OUTPUT      TERMINAL      PROBLEMS
OPEN EDITORS      seeds.rb db
Gemfile
# application.css app/assets/styles...
home.html.erb app/views/welcome
addresses.scss app/assets/style...
new.html.erb app/views/sessions
_form.html.erb app/views/addresses
games.scss app/assets/stylesheets
index.html.erb app/views/addresses

SDX
javascripts
channels
packs
JS application.js
jobs
mailers
models
views
addresses
_form.html.erb
edit.html.erb
index.html.erb
new.html.erb
show.html.erb
conferences
index.html.erb
show.html.erb
divisions
index.html.erb
show.html.erb
games
index.html.erb
show.html.erb
layouts
footer.html.erb

```

User Load (0.4ms) `SELECT "users".* FROM "users" WHERE "users"."id" IS NULL LIMIT ?` [["LIMIT", 1]]
 Parameters: {"authenticity_token": "47TLCyGe2L0KdcAEINX0fDZNfS51bSHJuTg0g0l4kgExylAT6w/Jk6oMjT4SB0dT8/M60Zy2xg0SiDRw==", "user": {"name": "Olive Queen", "email": "oliverqueen@gmail.com", "password": "FILTERED", "password_confirmation": "FILTERED"}, "commit": "Create"}
User Load (0.2ms) `SELECT "users".* FROM "users" WHERE "users"."id" IS NULL LIMIT ?` [["LIMIT", 1]]
Parameters: {"current_user": "oliverqueen@gmail.com"}
User Create (0.1ms) `INSERT INTO "users" ("name", "email", "password_digest", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?)` [["name", "Oliver Queen", "oliverqueen@gmail.com", "\$2a\$12\$pwUfrwGcIpbgf1007Lwo515DmQKn8ld9ltIUb.KCG0kFh0"], {"created_at": "2020-14 08:38:13.997875", "updated_at": "2020-05-14 08:38:13.997875"}]
app/controllers/application_controller.rb:6:in `current_user'
app/controllers/users_controller.rb:15:in `create'
(0.1ms) begin transaction
User Create (3.1ms) `INSERT INTO "users" ("name", "email", "password_digest", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?)` [["name", "Oliver Queen", "oliverqueen@gmail.com", "\$2a\$12\$pwUfrwGcIpbgf1007Lwo515DmQKn8ld9ltIUb.KCG0kFh0"], {"created_at": "2020-14 08:38:13.997875", "updated_at": "2020-05-14 08:38:13.997875"}]
app/controllers/users_controller.rb:15:in `create'
(0.1ms) commit transaction
app/controllers/users_controller.rb:15:in `create'
Redirected to http://localhost:3000/
Completed 302 Found in 224ms (ActiveRecord: 4.1ms | Allocations: 8593)

Started POST "/users" for ::1 at 2020-05-14 04:38:13 -0400
Processing by UsersController#create as HTML
Parameters: {"authenticity_token": "47TLCyGe2L0KdcAEINX0fDZNfS51bSHJuTg0g0l4kgExylAT6w/Jk6oMjT4SB0dT8/M60Zy2xg0SiDRw==", "user": {"name": "Olive Queen", "email": "oliverqueen@gmail.com", "password": "FILTERED", "password_confirmation": "FILTERED"}, "commit": "Create"}
User Load (0.2ms) `SELECT "users".* FROM "users" WHERE "users"."id" IS NULL LIMIT ?` [["LIMIT", 1]]
Parameters: {"current_user": "oliverqueen@gmail.com"}
User Create (0.1ms) `INSERT INTO "users" ("name", "email", "password_digest", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?)` [["name", "Oliver Queen", "oliverqueen@gmail.com", "\$2a\$12\$pwUfrwGcIpbgf1007Lwo515DmQKn8ld9ltIUb.KCG0kFh0"], {"created_at": "2020-14 08:38:13.997875", "updated_at": "2020-05-14 08:38:13.997875"}]
app/controllers/application_controller.rb:6:in `current_user'
app/controllers/users_controller.rb:15:in `create'
(0.1ms) begin transaction
User Create (3.1ms) `INSERT INTO "users" ("name", "email", "password_digest", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?)` [["name", "Oliver Queen", "oliverqueen@gmail.com", "\$2a\$12\$pwUfrwGcIpbgf1007Lwo515DmQKn8ld9ltIUb.KCG0kFh0"], {"created_at": "2020-14 08:38:13.997875", "updated_at": "2020-05-14 08:38:13.997875"}]
app/controllers/users_controller.rb:15:in `create'
(0.1ms) commit transaction
app/controllers/users_controller.rb:15:in `create'
Redirected to http://localhost:3000/
Completed 302 Found in 224ms (ActiveRecord: 4.1ms | Allocations: 8593)

Started GET "/" for ::1 at 2020-05-14 04:38:14 -0400
Processing by WelcomeController#home as HTML
User Load (0.1ms) `SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ?` [["id", 32], ["LIMIT", 1]]
CACHE User Load (0.1ms) `SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ?` [["id", 32], ["LIMIT", 1]]
Parameters: {"id": 32}
app/controllers/application_controller.rb:6:in `current_user'
app/controllers/welcome_controller.rb:6:in `current_user'
Rendered welcome/home.html.erb within layouts/application
Rendered welcome/home.html.erb within layouts/application (Duration: 1.3ms | Allocations: 239)
[Webpacker] Everything's up-to-date. Nothing to do
Rendered layouts/_navigation.html.erb (Duration: 0.2ms | Allocations: 217)
Rendered layouts/_messages.html.erb (Duration: 0.0ms | Allocations: 17)
Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)

SQL query creating the new user based on the create user form.

```

<h1>My Locations</h1>


<%= link_to 'Create Address', new_address_path %>


<br />
<table style="margin-left:auto;margin-right:auto;">
  <thead>
    <tr>
      <th>Address Type</th>
      <th>Address Line 1</th>
      <th>Address Line 2</th>
      <th>City</th>
      <th>State</th>
      <th>Zip Code</th>
      <th>Country</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <% @user.addresses.each do |address| %>
      <tr>
        <td><%= address.address_type %></td>
        <td><%= address.address_line_1 %></td>
        <td><%= address.address_line_2 %></td>
        <td><%= address.city %></td>
        <td><%= address.state %></td>
        <td><%= address.zipcode %></td>
        <td><%= address.country %></td>
        <td>
          <%= link_to 'View', address %> |<br/>
          <%= link_to 'Edit', edit_address_path(address) %> |<br/>
          <%= link_to 'Delete', address, method: :delete, data: { confirm: 'Are you sure you want to delete this address?' } %>
        </td>
      </tr>
    <% end %>
  </tbody>
</table>

```

Embedded Ruby and HTML code behind the Locations index page.

The screenshot shows a browser window with the URL localhost:3000/addresses?. The page title is "SDX". The main content area displays a table with one row, showing address details for a home location. The table has columns for Address Type, Address Line 1, Address Line 2, City, State, Zip Code, Country, and actions (View | Edit | Delete). Below the table, a copyright notice is visible.

Address Type	Address Line 1	Address Line 2	City	State	Zip Code	Country	
Home	120 14th Avenue	Apt. 2	Seattle	WA	98122	United States	View Edit Delete

Copyright © SportsDataXchange. All rights reserved.

User Locations Index Page

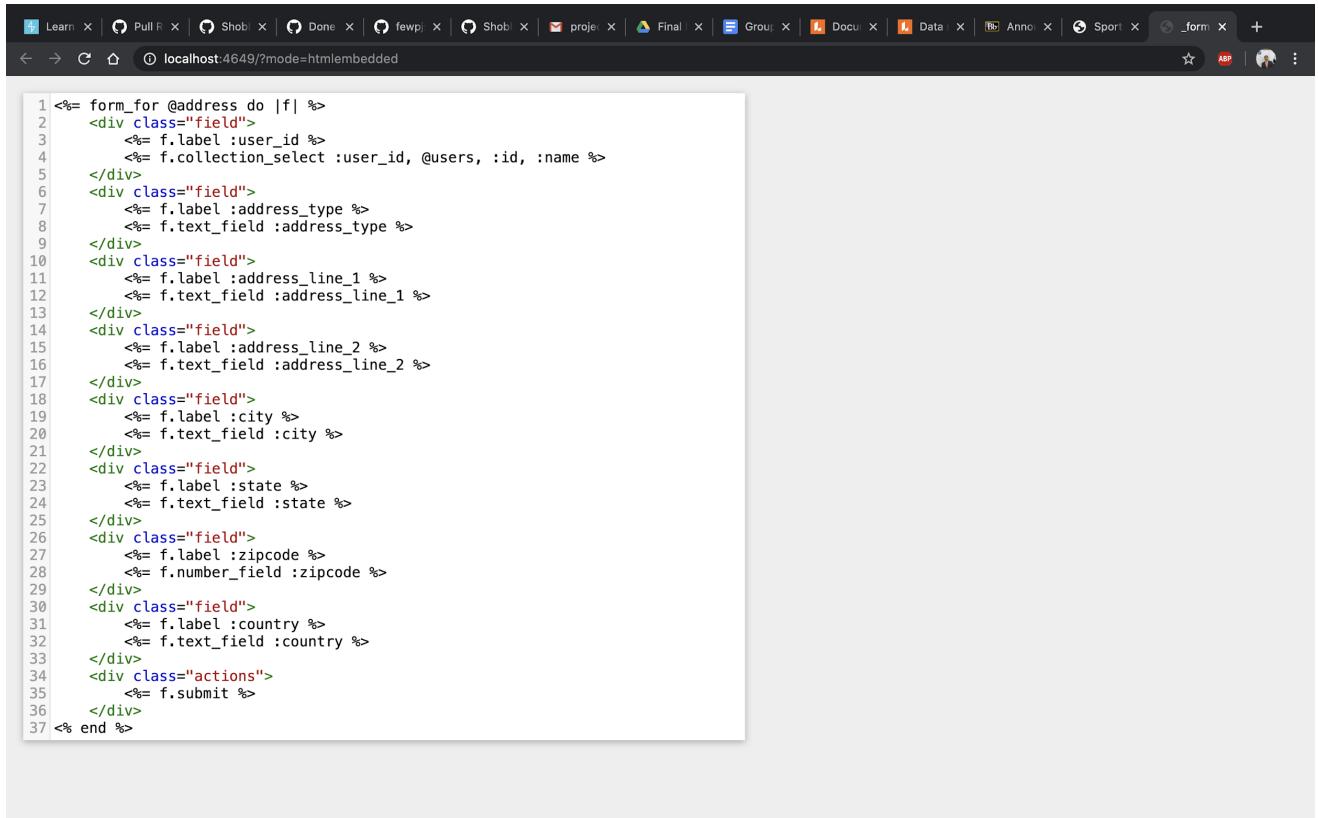
The screenshot shows a browser window displaying a Ruby on Rails application's logs. The logs are organized into sections corresponding to different routes and actions. Each section includes the log entry, processing time, and detailed parameters or results.

```
Started POST "/addresses" for ::1 at 2020-05-14 04:42:31 -0400
Processing by AddressesController#create as HTML
Parameters: {"authenticity_token"=>"DfPPFTIm5dQwNjNep3nukP2BjE6GE1LB1P189mgzrXT/YQkYPSStkd6EY7NDIN2+0y9hwzo0ak7M8gUNsYfcQ==", "address"=>{"user_id"=>"32", "address_type"=>"Home", "address_line_1"=>"120 14th Avenue", "address_line_2": "", "city"=>"Seattle", "state"=>"WA", "zipcode"=>"98122"}, "utf8"=>"✓", "commit"=>"Create Address"}
User Load (0.1ms)  SELECT "users".* FROM `users` WHERE `users`.`id` = ? LIMIT 1  [["id", 32], ["LIMIT", 1]]
CACHE User Load (0.0ms)  SELECT "users".* FROM `users` WHERE `users`.`id` = ? LIMIT 1  [["id", 32], ["LIMIT", 1]]
app/controllers/application_controller.rb:6:in `current_user'
app/controllers/addresses_controller.rb:19:in `create'
(0.1ms) begin transaction
app/controllers/addresses_controller.rb:19:in `create'
Address Create (0.8ms) INSERT INTO `addresses` ("user_id", "address_line_1", "address_type", "address_line_2", "city", "state", "zipcode", "country", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?, ?, ?, ?)  [["user_id", 32], ["address_line_1", "120 14th Avenue"], ["address_type", "Home"], ["address_line_2", "Apt. 2"], ["city", "Seattle"], ["state", "WA"], ["zipcode", 98122], ["country", "United States"], ["created_at", "2020-05-14 08:42:31.041467"], ["updated_at", "2020-05-14 08:42:31.041467"]]
app/controllers/addresses_controller.rb:19:in `create'
(2.2ms) commit transaction
app/controllers/addresses_controller.rb:19:in `create'
Redirected to http://localhost:3000/addresses/2
Completed 302 Found in 25ms (ActiveRecord: 3.3ms | Allocations: 7649)
```

```
Started GET "/addresses/2" for ::1 at 2020-05-14 04:42:31 -0400
Processing by AddressesController#show as HTML
Parameters: {"id"=>"2"}
User Load (0.2ms)  SELECT "users".* FROM `users` WHERE `users`.`id` = ? LIMIT 1  [["id", 32], ["LIMIT", 1]]
CACHE User Load (0.0ms)  SELECT "users".* FROM `users` WHERE `users`.`id` = ? LIMIT 1  [["id", 32], ["LIMIT", 1]]
app/controllers/application_controller.rb:6:in `current_user'
Address Load (0.2ms)  SELECT "addresses".* FROM `addresses` WHERE `addresses`.`user_id` = ? AND `addresses`.`id` = ? LIMIT 1  [["user_id", 32], ["id", 2], ["LIMIT", 1]]
app/controllers/addresses_controller.rb:9:in `show'
Rendering addresses/show.html.erb within layouts/application
Rendered addresses/show.html.erb within layouts/application (Duration: 2.7ms | Allocations: 599)
[Webpacker] Everything's up-to-date. Nothing to do
Rendered layouts/_navigation.html.erb (Duration: 0.3ms | Allocations: 219)
Rendered layouts/_messages.html.erb (Duration: 0.0ms | Allocations: 17)
Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)
Completed 200 OK in 16ms (Views: 10.0ms | ActiveRecord: 0.4ms | Allocations: 8669)
```

```
Started GET "/addresses" for ::1 at 2020-05-14 04:42:38 -0400
Processing by AddressesController#index as HTML
User Load (0.2ms)  SELECT "users".* FROM `users` WHERE `users`.`id` = ? LIMIT 1  [["id", 32], ["LIMIT", 1]]
app/controllers/application_controller.rb:6:in `current_user'
CACHE User Load (0.0ms)  SELECT "users".* FROM `users` WHERE `users`.`id` = ? LIMIT 1  [["id", 32], ["LIMIT", 1]]
app/controllers/addresses_controller.rb:6:in `current_user'
Rendered addresses/index.html.erb within layouts/application
Address Load (0.3ms)  SELECT "addresses".* FROM `addresses` WHERE `addresses`.`user_id` = ?  [["user_id", 32]]
app/views/addresses/index.html.erb:18
Rendered addresses/index.html.erb within layouts/application (Duration: 2.6ms | Allocations: 893)
[Webpacker] Everything's up-to-date. Nothing to do
Rendered layouts/_navigation.html.erb (Duration: 0.3ms | Allocations: 213)
Rendered layouts/_messages.html.erb (Duration: 0.1ms | Allocations: 17)
```

SQL Query creating an address in the database and displaying the Locations index page.

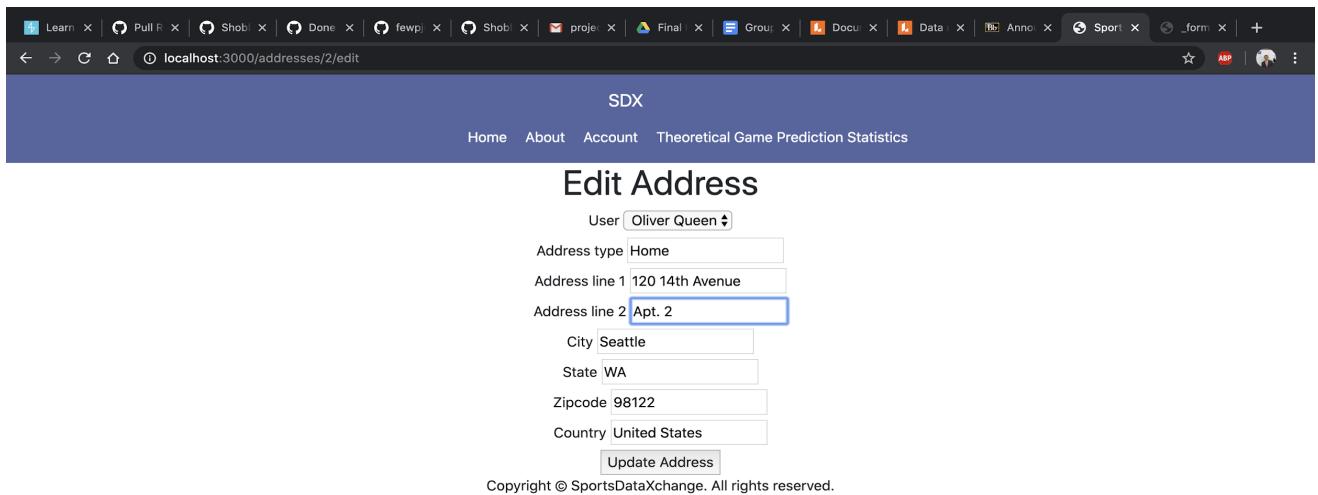


```

1 <%= form_for @address do |f| %>
2   <div class="field">
3     <%= f.label :user_id %>
4     <%= f.collection_select :user_id, @users, :id, :name %>
5   </div>
6   <div class="field">
7     <%= f.label :address_type %>
8     <%= f.text_field :address_type %>
9   </div>
10  <div class="field">
11    <%= f.label :address_line_1 %>
12    <%= f.text_field :address_line_1 %>
13  </div>
14  <div class="field">
15    <%= f.label :address_line_2 %>
16    <%= f.text_field :address_line_2 %>
17  </div>
18  <div class="field">
19    <%= f.label :city %>
20    <%= f.text_field :city %>
21  </div>
22  <div class="field">
23    <%= f.label :state %>
24    <%= f.text_field :state %>
25  </div>
26  <div class="field">
27    <%= f.label :zipcode %>
28    <%= f.number_field :zipcode %>
29  </div>
30  <div class="field">
31    <%= f.label :country %>
32    <%= f.text_field :country %>
33  </div>
34  <div class="actions">
35    <%= f.submit %>
36  </div>
37 <% end %>

```

Embedded Ruby and HTML code for creating and editing User's Address



The screenshot shows a web application interface for editing an address. The top navigation bar includes links for Home, About, Account, and Theoretical Game Prediction Statistics. The main title is "Edit Address". Below the title, there is a dropdown menu labeled "User" with "Oliver Queen" selected. The form fields are as follows:

- Address type: Home
- Address line 1: 120 14th Avenue
- Address line 2: Apt. 2
- City: Seattle
- State: WA
- Zipcode: 98122
- Country: United States

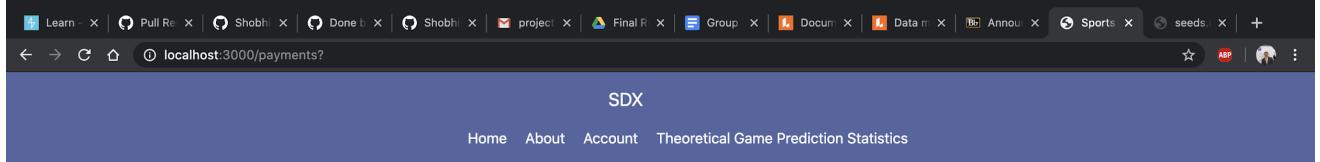
A "Update Address" button is located at the bottom of the form. A copyright notice at the very bottom states "Copyright © SportsDataXchange. All rights reserved."

Edit Address page

The screenshot shows a developer's workspace with several windows and tabs open:

- EXPLORER**: Shows the project structure with files like `seed.rb`, `Gemfile`, `application.css`, `home.html.erb`, `addresses.scss`, `new.html.erb`, `form.html.erb`, `games.scss`, `index.html.erb`, `index.html.erb`, `show.html.erb`, `users`, `edit.html.erb`, `new.html.erb`, `show.html.erb`, `welcome`, `home.html.erb`, `bin`, `config`, `db`, and `migrate`. Migrations include `20200510033550_create_users.rb`, `20200510033640_create_addresses.rb`, `20200510033747_create_payments.rb`, `20200510033841_create_teams.rb`, `20200510033856_create_conferences.rb`, `20200510033930_create_divisions.rb`, `20200510033948_create_games.rb`, `20200510034000_create_games.rb`, `20200510034014_create_seasons.rb`, and `20200510034032_create_currents.rb`.
- DEBUG CONSOLE**: Shows the history of database queries and processing times.
- OUTPUT**: Shows the command-line output, including the start of the application server (`rails s`) and log messages.
- TERMINAL**: Shows the current terminal session with the command `rails s`.
- PROBLEMS**: Shows any errors or warnings in the project.
- RIGHT SIDE**: Shows the current file in the editor, `addresses_controller.rb`, with a detailed view of the `edit` action. It includes the SQL query for selecting users, the update operation, and the final response.

SQL Query after user's address is updated.



Payment History

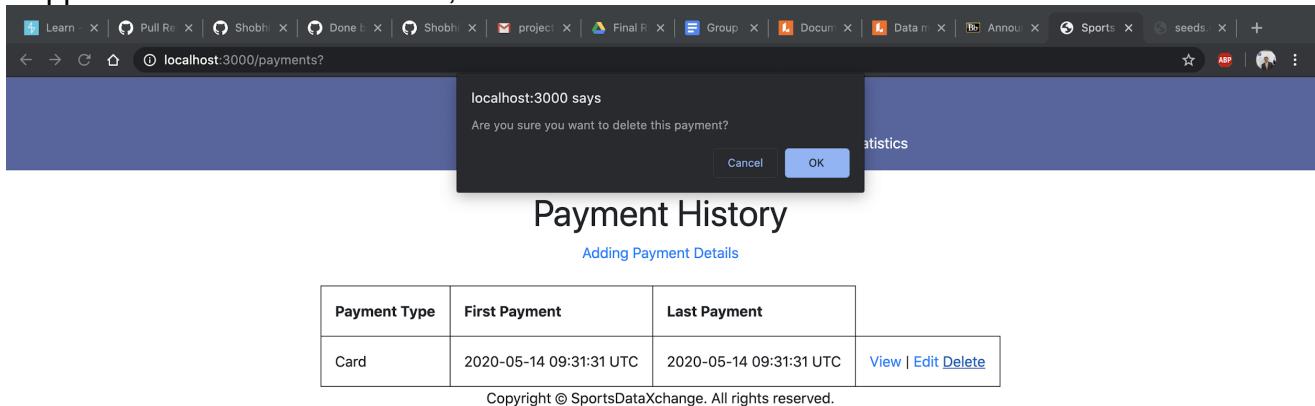
Adding Payment Details

Payment Type	First Payment	Last Payment	
Card	2020-05-14 09:31:31 UTC	2020-05-14 09:31:31 UTC	View Edit Delete

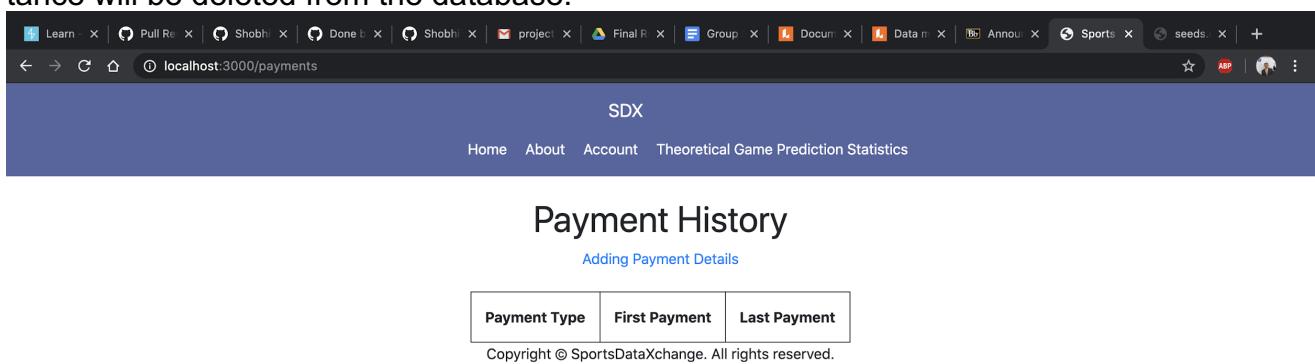
Copyright © SportsDataXchange. All rights reserved.

Payment History page and example of the delete functionality. As soon as the user clicks on the delete button a validation will pop up confirming whether they want to delete an instance from the payments table.

Suppose we click the ok button, the ins



tance will be deleted from the database.



Payment History Page after deleting the Payment information.

EXPLORER DEBUG CONSOLE OUTPUT TERMINAL PROBLEMS

OPEN EDITORS seeds.rb Gemfile application.css home.html.erb addresses.scss new.html.erb show.html.erb games.scss index.html.erb

SDX bin config db migrate 20200510033550_create_users.rb 20200510033640_create_addresses... 20200510033747_create_payment... 20200510033841_create_teams.rb 20200510033856_create_confere... 20200510033930_create_divisio... 20200510033948_create_gamed... 20200510034000_create_games... 20200510034014_create_seaso... 20200510034032_create_curren... development.sqlite3 schema.rb seeds.rb lib log .keep development.log node_modules public storage OUTLINE TIMELINE NPM SCRIPTS

Started GET "/payments" for ::1 at 2020-05-14 05:31:34 -0400
 Processing by PaymentsController#index as HTML
 Parameters: {"authenticity_token": "d4e01b0onei5nu5hAbG2crLjH6r0xSpI/lUKR/PuFQwzJ3oVLr+NoMpAvXUKV1bm0a850umj7TFkD0WqoMTw==", "id": "2"}
 User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id": 32}, {"LIMIT": 1}]
 CACHE User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id": 32}, {"LIMIT": 1}]
 4 app/controllers/application_controller.rb:6:in `current_user'
 Rendering payments/index.html.erb within layouts/application
 Payment Load (0.1ms) SELECT "payments".* FROM "payments" WHERE "payments"."user_id" = ? [{"user_id": 32}]
 4 app/views/payments/index.html.erb:15
 Rendered payments/index.html.erb within layouts/application (Duration: 2.3ms | Allocations: 932)
 [Webpacker] Everything's up-to-date. Nothing to do
 Rendered layouts/_navigation.html.erb (Duration: 0.3ms | Allocations: 213)
 Rendered layouts/_messages.html.erb (Duration: 0.0ms | Allocations: 17)
 Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)
 Completed 200 OK in 19ms (Views: 13.6ms | ActiveRecord: 0.4ms | Allocations: 8411)

Started DELETE "/payments/2" for ::1 at 2020-05-14 05:36:10 -0400
 Processing by PaymentsController#destroy as HTML
 Parameters: {"authenticity_token": "d4e01b0onei5nu5hAbG2crLjH6r0xSpI/lUKR/PuFQwzJ3oVLr+NoMpAvXUKV1bm0a850umj7TFkD0WqoMTw==", "id": "2"}
 User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id": 32}, {"LIMIT": 1}]
 CACHE User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id": 32}, {"LIMIT": 1}]
 4 app/controllers/application_controller.rb:6:in `current_user'
 Rendering payments/index.html.erb within layouts/application
 Payment Load (0.1ms) SELECT "payments".* FROM "payments" WHERE "payments"."id" = ? [{"id": 2}]
 4 app/views/payments/index.html.erb:15
 Payment Destroy (0.5ms) DELETE FROM "payments" WHERE "payments"."id" = ? [{"id": 2}]
 4 app/controllers/payments_controller.rb:37:in `destroy'
 (1.3ms) commit transaction
 4 app/controllers/payments_controller.rb:37:in `destroy'
 (1.3ms) commit transaction
 4 app/controllers/payments_controller.rb:37:in `destroy'
 Redirected to http://localhost:3000/payments
 Completed 302 Found in 10ms (ActiveRecord: 2.2ms | Allocations: 3469)

Started GET "/payments" for ::1 at 2020-05-14 05:36:10 -0400
 Processing by PaymentsController#index as HTML
 User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id": 32}, {"LIMIT": 1}]
 CACHE User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id": 32}, {"LIMIT": 1}]
 4 app/controllers/application_controller.rb:6:in `current_user'
 Rendering payments/index.html.erb within layouts/application
 Payment Load (0.1ms) SELECT "payments".* FROM "payments" WHERE "payments"."user_id" = ? [{"user_id": 32}]
 4 app/views/payments/index.html.erb:15
 [Webpacker] Everything's up-to-date. Nothing to do
 Rendered layouts/_navigation.html.erb (Duration: 0.3ms | Allocations: 213)
 Rendered layouts/_messages.html.erb (Duration: 0.0ms | Allocations: 17)
 Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)
 Completed 200 OK in 19ms (Views: 15.7ms | ActiveRecord: 0.3ms | Allocations: 7768)

Ln 192, Col 1 Spaces: 4 UTF-8 LF Ruby ⌂ ⌂

Corresponding SQL query after user clicks the delete button.

localhost:3000/conferences

SDX

Home About Account Theoretical Game Prediction Statistics

Conferences

Name	
American Football Conference (AFC)	View
National Football Conference (NFC)	View

Copyright © SportsDataXchange. All rights reserved.

NFL Conferences Index Page

All 4 division links of the AFC Conference after user clicks on the View link in the index page.

```

1 <div class="container">
2   <h1><=? @conference.name %>
3
4   <h2>Divisions in this Conference:</h2>
5   <% @conference.divisions.uniq.each do |division| %>
6     <p><=? link_to division.region, division_path(division) %>
7   <% end %>
8
9   <h2><=? button_to "Back", conferences_path, method: :get %></h2>
10 </div>

```

Embedded Ruby and HTML code for the AFC Conference Page.

```

1 <div class="container">
2   <h1><=? @conference.name %>
3
4   <h2>Divisions in this Conference:</h2>
5   <% @conference.divisions.uniq.each do |division| %>
6     <p><=? link_to division.region, division_path(division) %>
7   <% end %>
8
9   <h2><=? button_to "Back", conferences_path, method: :get %></h2>
10 </div>

Started GET "/payments" for ::1 at 2020-05-14 04:54:32 -0400
Processing by PaymentsController#index as HTML
User Load (0.3ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [ ["id", 32], [ "LIMIT", 1] ]
app/controllers/application_controller.rb:6:in `current_user'
CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [ ["id", 32], [ "LIMIT", 1] ]
app/controllers/application_controller.rb:6:in `current_user'
Rendering layouts/_index.html.erb within layouts/application
Payment Load (0.3ms)  SELECT "payments".* FROM "payments" WHERE "payments"."user_id" = ? [ ["user_id", 32] ]
app/views/payments/index.html.erb:15
Rendered payments/index.html.erb within layouts/application (Duration: 6.0ms | Allocations: 2339)
[Webpacker] Everything's up-to-date. Nothing to do
Rendered layouts/_navigation.html.erb (Duration: 0.4ms | Allocations: 219)
Rendered layouts/_messages.html.erb (Duration: 0.1ms | Allocations: 17)
Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)
Completed 200 OK in 22ms (Views: 17.1ms | ActiveRecord: 0.4ms | Allocations: 8242)

Started GET "/payments" for ::1 at 2020-05-14 04:54:32 -0400
Processing by PaymentsController#index as HTML
User Load (0.3ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [ ["id", 32], [ "LIMIT", 1] ]
app/controllers/application_controller.rb:6:in `current_user'
CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [ ["id", 32], [ "LIMIT", 1] ]
app/controllers/application_controller.rb:6:in `current_user'
Rendering layouts/_index.html.erb within layouts/application
Payment Load (0.3ms)  SELECT "payments".* FROM "payments" WHERE "payments"."user_id" = ? [ ["user_id", 32] ]
app/views/payments/index.html.erb:15
Rendered payments/index.html.erb within layouts/application (Duration: 6.0ms | Allocations: 2339)
[Webpacker] Everything's up-to-date. Nothing to do
Rendered layouts/_navigation.html.erb (Duration: 0.4ms | Allocations: 219)
Rendered layouts/_messages.html.erb (Duration: 0.1ms | Allocations: 17)
Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)
Completed 200 OK in 22ms (Views: 17.1ms | ActiveRecord: 0.4ms | Allocations: 8242)

Started GET "/conferences" for ::1 at 2020-05-14 04:54:49 -0400
Processing by ConferencesController#index as HTML
User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [ ["id", 32], [ "LIMIT", 1] ]
app/controllers/application_controller.rb:6:in `current_user'
CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [ ["id", 32], [ "LIMIT", 1] ]
app/controllers/application_controller.rb:6:in `current_user'
Rendering conferences/index.html.erb within layouts/application
Conference Load (0.3ms)  SELECT "conferences".* FROM "conferences"
app/views/conferences/index.html.erb:11
Rendered conferences/index.html.erb within layouts/application (Duration: 8.1ms | Allocations: 3564)
[Webpacker] Everything's up-to-date. Nothing to do
Rendered layouts/_navigation.html.erb (Duration: 0.3ms | Allocations: 219)
Rendered layouts/_messages.html.erb (Duration: 0.1ms | Allocations: 17)
Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)
Completed 200 OK in 25ms (Views: 17.0ms | ActiveRecord: 1.2ms | Allocations: 10250)

Started GET "/conferences/1" for ::1 at 2020-05-14 04:56:46 -0400
Processing by ConferencesController#show as HTML
Parameters: {"id": "1"}
User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [ ["id", 32], [ "LIMIT", 1] ]
app/controllers/application_controller.rb:6:in `current_user'
Conference Load (0.2ms)  SELECT "conferences".* FROM "conferences" WHERE (1) LIMIT ? [ ["LIMIT", 1] ]
app/controllers/conferences_controller.rb:7:in `show'
Rendering conferences/show.html.erb within layouts/application
Division Load (0.3ms)  SELECT "divisions".* FROM "divisions" INNER JOIN "teams" ON "divisions"."id" = "teams"."division_id" WHERE "teams"."conference_id" = ? [ ["conference_id", 1] ]
app/views/conferences/show.html.erb:5:in `uniq'
Rendered conferences/show.html.erb within layouts/application (Duration: 35.6ms | Allocations: 12897)
[Webpacker] Everything's up-to-date. Nothing to do
Rendered layouts/_navigation.html.erb (Duration: 0.3ms | Allocations: 218)
Rendered layouts/_messages.html.erb (Duration: 0.1ms | Allocations: 17)
Rendered layouts/_footer.html.erb (Duration: 0.0ms | Allocations: 5)
Completed 200 OK in 55ms (Views: 48.4ms | ActiveRecord: 2.6ms | Allocations: 20173)

```

SQL query showing the conversion of Ruby code into Inner Join statement.

Suppose, the user clicks on the East link on the show page they will be redirected to the Division show page where they can see all the teams in that division as follows:

SDX

Home About Account Theoretical Game Prediction Statistics

East

Teams in this division

[Buffalo Bills](#)
[Miami Dolphins](#)
[New England Patriots](#)
[New York Jets](#)

[Back](#)

Copyright © SportsDataXchange. All rights reserved.

Similarly, we repeated the steps for the teams and games to get to the game stats.

SDX

Home About Account Theoretical Game Prediction Statistics

Buffalo Bills

Orchard Park, New York



All games played by this team

[Buffalo Bills vs. NY Jets](#)
[Buffalo Bills vs. NY Giants](#)
[Buffalo Bills vs. Cincinnati Bengals](#)
[Buffalo Bills vs. New England Patriots](#)
[Buffalo Bills vs. Tennessee Titans](#)
[Buffalo Bills vs. Miami Dolphins](#)

The screenshot shows a browser window with a tab bar at the top containing various tabs like 'Learn', 'Pull Re...', 'Shobhi', 'Done', 'fewpjs', 'Shobhi', 'project', 'Final R...', 'Group', 'Docum...', 'Data m...', 'Annou...', and 'Sports'. The main content area has a dark blue header with the text 'SDX' and navigation links for 'Home', 'About', 'Account', and 'Theoretical Game Prediction Statistics'. Below the header is a section titled 'Buffalo Bills vs. NY Giants' with game details: 'Winner: Buffalo Bills', 'Loser: NY Giants', and 'Game Time: 2019-09-15'. A heading 'Game stats are as follows' is followed by a table comparing the two teams. The table has columns for Team Name, Score, Time of Possession, Passing Yards, Rushing Yards, Total Turnovers, and Penalties. The data shows Buffalo Bills with 28 points, 32:38 possession, 237.0 passing, 151.0 rushing, 2 turnovers, and 9 penalties. New York Giants have 14 points, 27:22 possession, 241.0 passing, 129.0 rushing, 0 turnovers, and 7 penalties.

Based on this game's analysis, we have seen that the winning teams on average have higher passing yards and higher rushing yards than the losing team.

Time of possession is a significant factor in completely one sided games but has a reduced impact in even contests.

[Back](#)

Copyright © SportsDataXchange. All rights reserved.

Based on the stats and after analyzing the data we have reached to the conclusion that winning teams on average tend to have higher passing and rushing yards than the losing team. Additionally, time of possession is a major factor in completely one sided games but has a reduced impact in close games.

Example of how we seeded our data in the SQLite database:

```
12 #
13 #   movies = Movie.create([{ name: 'Star Wars' }, { name: 'Lord of the Rings' }])
14 #   Character.create(name: 'Luke', movie: movies.first)
15 User.destroy_all
16 Division.destroy_all
17 Conference.destroy_all
18 Team.destroy_all
19 Game.destroy_all
20 Gameday.destroy_all
21 Season.destroy_all
22
23 30.times do
24   User.create(name: Faker::Name.unique.name, email:
25   Faker::Internet.unique.email, password: Faker::IDNumber.valid)
26 end
27 east = Division.create(region: "East")
28 north = Division.create(region: "North")
29 west = Division.create(region: "West")
30 south = Division.create(region: "South")
31
32 afc = Conference.create(name: "American Football Conference (AFC)")
33 nfc = Conference.create(name: "National Football Conference (NFC)")
34
35 buffalo_bills = Team.create(conference: afc, division: east, name: "Buffalo
36 Bills", city: "Orchard Park, New York", image: "https://premcom.com/wp-
37 content/uploads/2014/01/buffalo-bills-portfolio-logo.jpg")
38 miami_dolphins = Team.create(conference: afc, division: east, name: "Miami
39 Dolphins", city: "Miami Gardens, Florida", image:
"https://exstreamist.com/wp-content/uploads/2016/08/watch-Miami-Dolphins-
Online.jpg")
40 new_england_patriots = Team.create(conference: afc, division: east, name:
41 "New England Patriots", city: "Foxborough, Massachusetts", image:
42 "https://rfathead-
43 res.cloudinary.com/image/upload/q_auto,f_auto/c_pad,h_900/g_north,c_crop,h_90
44 0,w_900/c_pad,h_900,w_900/roomplus/14-
45 14529_nfl_new_england_patriots_2016_state_of_massachusetts_logo_6628.jpg")
46 new_york_jets = Team.create(conference: afc, division: east, name: "New York
47 Jets", city: "East Rutherford, New Jersey", image:
48 "https://static.clubs.nfl.com/image/private/t_editorial_landscape_12_desktop/
49 jets/nue7vbohnzqlptdk3z6")
```

39 |

Phase 5: System Implementation

Executive Summary

Of the many things we learned while working on this project, we believe the most important was that communication between the team members was paramount to success. No one member of the team could have produced this on their own. As such, we sought a means to communicate and be accountable to each other. We set up both a what's app chat and Slack channel to enable communication among the members of the team. Accordingly, we sought to find the talents of each team member and rely on them to be the leader of different segments of the project under the direction of the project leader. We would document our team meetings as we went along in a specific slack channel and then take the deliverables, timeframe changes and other elements of project management from that communications and apply them to the proper documentation. As a team, we were surprised by the lack of an organized readily available data source to begin the model calibration. In addition, the work to come up with a competitive model with limited error was a little surprising, perhaps if it was easy everyone would have one. The way the team worked together was not surprising, as many of us had prior collaborative experience with at least another member of the team. Leveraging the specific skills of a team member and allowing them the freedom to complete a section of the project proved an important strategy as it allowed the team member to take ownership and pride in that section of the work. While there were obstacles, such as organizing the source data or making sure that our communication was effective, the team collectively was always looking to solve any problems. Early on we decided to keep all communication blame free, with a focus on solutions and the process served us well.

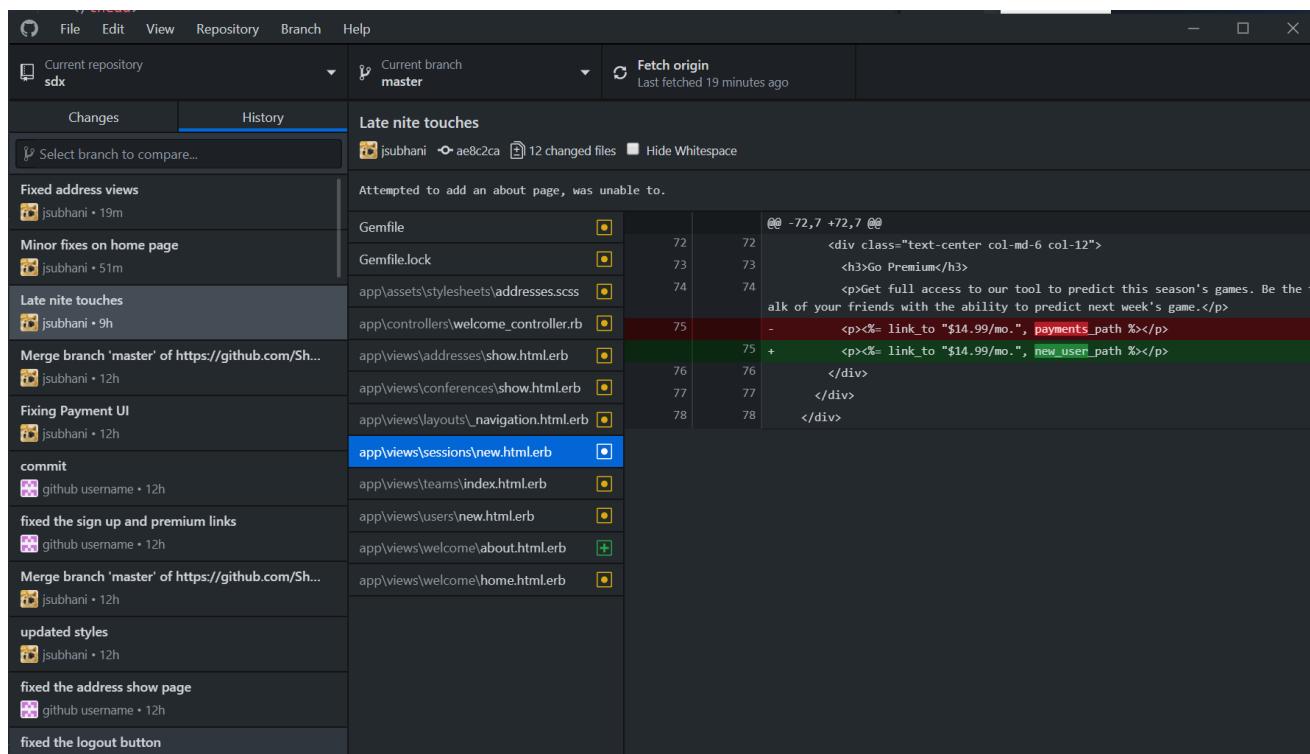
Hardware / Software

To maximize efficiency in the development phase, SDX will be developed in a cloud environment using the Ruby on Rails framework using Amazon Web Services. Ruby on Rails is a server-side web application framework that has an all-in-one package for backend and front-end development. The Ruby programming language streamlines interpretation and implementation of conventional data analytics code. Compared to other popular languages, like Python, the main advantage of Ruby is the flexibility and easier troubleshooting that its environment can provide. Communication between the front-end designers, programmers, and database management staff would have less barriers and this can ensure efficient development and testing. We will be using an Object Relationship Mapper (ORM) called ActiveRecord into SQL code. Due to the current size of the company and small user base, investing on our own server-side hardware is not worth it compared to cloud services. Costs of hardware, maintenance, and security is higher than offloading the responsibility to AWS, where we would be billed monthly for significantly less based on usage.

Testing / Training

Due to budget, time constraints, along with the decision to work on the Ruby on Rails Framework, testing is mostly done during the development phase by the development team. However, the project manager and the entire team manually tested the prototype to keep all things in working order. The Ruby on Rails framework, scope of the project, as well as design choices of the front end and back end, streamlined the Quality Assurance phase enough where automation testing is not necessary for the prototyping phase. Potential testing practices in the future can be done in a number of ways to test the User experience, such as Selenium.

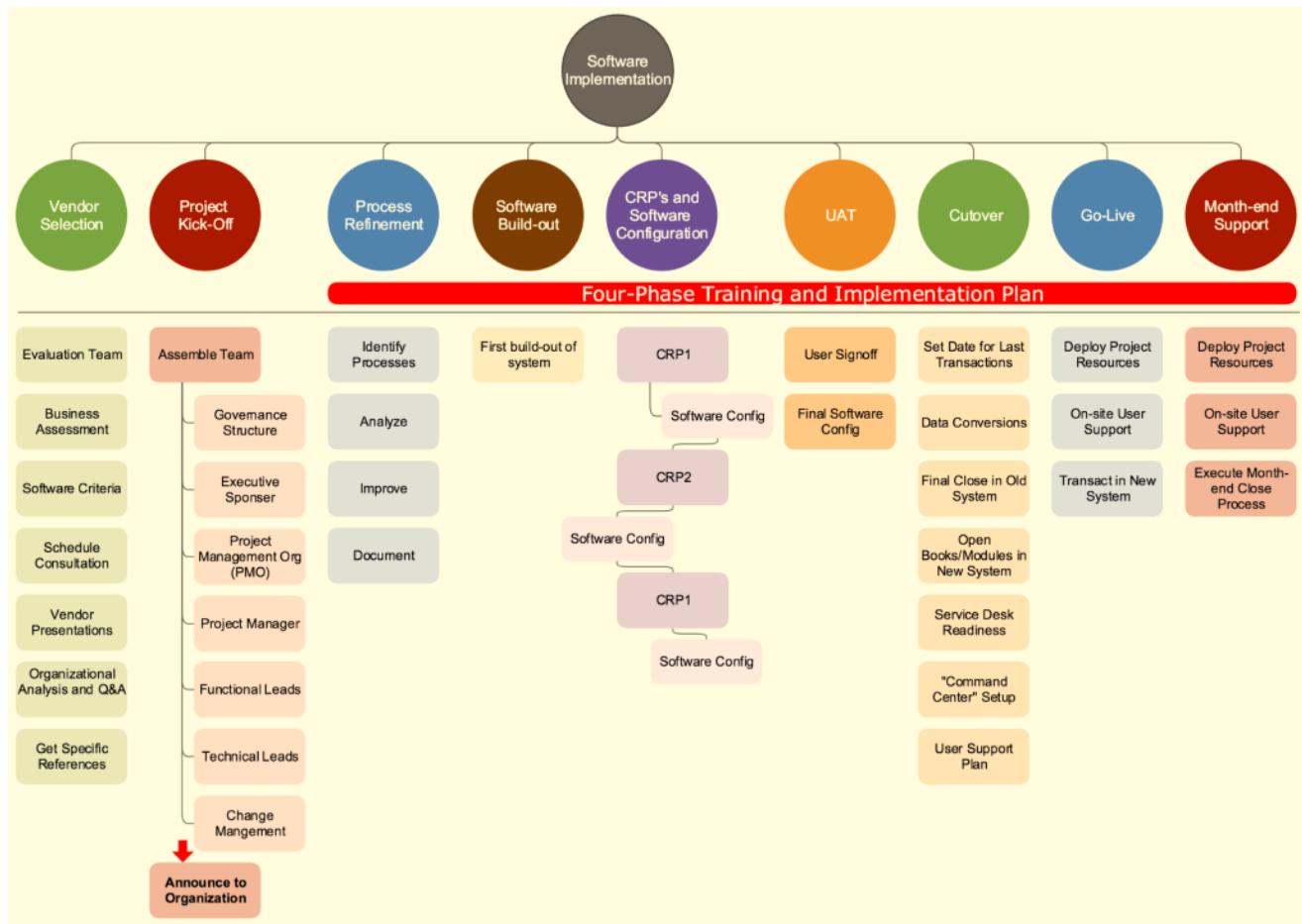
The development process was very dependent on Github, as it acted as a repository that logged all changes that took place during development. Should there be a website crash, Github's version control allows developers to revert to a stable build and see what went wrong.

A screenshot of the GitHub desktop application interface. The top navigation bar includes File, Edit, View, Repository, Branch, and Help. The main window shows a pull request between 'Current repository' (sdx) and 'Current branch' (master). The 'History' tab is selected. A commit titled 'Late nite touches' by 'jsubhani' is highlighted. The commit message says 'Attempted to add an about page, was unable to.' The diff view shows code changes in the Gemfile and app\views\addresses\show.html.erb files. The Gemfile change adds 'payments_path' and removes 'new_user_path'. The show.html.erb change adds 'link_to "\$14.99/mo.", payments_path' and removes 'link_to "\$14.99/mo.", new_user_path'. The commit message also mentions 'Get full access to our tool to predict this season's games. Be the talk of your friends with the ability to predict next week's game.'

Most of the Quality Assurance were in the hands of the development team who collaborated and proofread each other's work via Zoom meetings and sharing the GitHub repository. All changes and edits were documented on Github and both the Front End and Back End developer kept separate branches of the repository to avoid conflicts before merging. In order for the prototype to work, the user would need to run a Rails server with specific dependencies on the Ruby on Rails framework called "gem" files. Among them was rails, yarn, a number of encryption libraries, and bootstrap. Implementation of gems was a collaborative effort. Initially, there was a plan to use web hosting on heroku for the demonstration, however, it was decided

that the prototype needed to be hosted on rails servers of the developers due to stability issues during development. Additionally, Heroku would have conflicts with the github repository and would need constant reconfiguration in a development environment. This would have delayed the prototype demonstration. Both developers had to have similar configurations on their machines in order to collaborate and test properly.

Implementation Plan



Source: [LisoBlog](#)

Post Launch / Support Plan

Our demographic customers look for quick and simple solutions to any technical questions. As our company focuses on the end users' overall experience and responses to the product or service, it is a critical matter to make our product easy to use and convenient. To support users in an effective manner, we created a FAQ page. This will be easy to access and will cover basic information about our product and common questions such as payment issues and specific membership features. We hope to answer about 80% of any tech inquiries with our FAQ guide. Furthermore, we can optimize our FAQs for multiple devices that make

answers to all questions available via smartphone as well. With a well-crafted FAQ section, we can provide a quick and effective response to users.

For the users who can not find their answer in the FAQ we will provide live chat communication. We intend to include this in our budget and the cost of having 24 hour FAQ tech support outsourced to a call center overseas should cost between \$70-\$130K a year, depending on the vendor and location. The largest advantage of live chat outsourcing is the cost benefit and flexibility. Through outsourcing chat support, we can cut down on the number of things we do in-house, but also we can focus on the core elements of our business.

Future Modification Report

The title of the project is Sport Data Xchange, not NFL Football data exchange, and as such our goal is to scale and expand into other sports with gambling interest across the globe. We hope to build the data source to help determine the outcome of American College Football, Professional Football (European Soccer) and Cricket. Our goals are to continue to build a consumer tool that is entertaining but also helps those who choose to gamble. Therefore any expansion of what is next is focused on scaling the system to include these other sports. We do not currently have plans to move into the market as bookmakers but would not rule out a joint venture with established bookmakers in markets where our product adds value.

Appendix

A. MS Project Plan

1	Task Id	Task Name	Milestone Task	Status	Est Duration	Est Start Date	Est End Date	Actual Duration	Actual Start Date	Actual End Date	Resources	Predecessor
2	1	Project Charter									Luke,Hyuk,Shobhit,Jason,Dohun	
3	1.1	Create Project Charter		Complete	1	2/17/20	2/18/20	1	2/17/20	2/17/20	Luke,Hyuk,Shobhit,Jason,Dohun	
4	1.2	Review and Corrections		Complete	0.5	2/18/20	2/18/20	0.50	2/18/20	2/18/20	Luke,Hyuk,Shobhit	1.1
5	1.3	Approval	Milestone	Complete		2/18/20	2/18/20					
6	1.4	Final Project Documentation		Complete	80	2/19/20	5/9/20	84	2/19/20	5/13/20	Luke,Hyuk,Dohun,Shobhit	1.3
7	2	Business Requirements										
8	2.1	Gather Requirements		Complete	18	2/19/20	3/8/20	22	2/19/20	3/12/20	Luke,Hyuk,Shobhit,	1
9	2.2	Documentation		Complete	20	2/20/20	3/11/20	32	2/20/20	3/23/20	Shobhit,Jason	
10	2.3	Approval		Complete		3/11/20			3/23/20			2.2
11	3	Design										2.3
12	3.1	System Architecture		Complete	4	3/11/20	3/15/20	3	3/20/20	3/23/20	Shobhit,Jason,Dohun	2.3 -3days
13	3.1.1	Review and Corrections		Complete	5	3/12/20	3/17/20	3	3/21/20	3/24/20	Luke,Hyuk,Shobhit	2.3.1
14	3.2	Database Design		Complete	30	3/13/20	4/12/20	30	3/23/20	4/22/20	Shobhit, Dohun	2.3.1
15	3.2.1	Review and Corrections		Complete	32	3/14/20	4/15/20	30	3/26/20	4/25/20	Luke,Hyuk,Shobhit	
16	3.3	Web Design		Complete	34	3/20/20	4/23/20	34	3/24/20	4/27/20	Jason, Shobhit	
17	3.3.1	Review and Corrections		Complete	35	3/25/20	4/29/20	32	3/30/20	5/1/20	Luke,Hyuk,Shobhit	
18	3.4	Approval	Milestone		0						N/A	3.1.1,3.2.1,3.3.
19	4	Development & Testing			19	5/5/20	5/24/20		5/5/20			3.3
20	4.1	Iteration 1		Complete	7	5/5/20	5/12/20	8	5/5/20	5/13/20		3.3
21	4.1.1	Development		Complete	5	5/5/20	5/10/20	5	5/5/20	5/10/20	Shobhit,Jason	
22	4.1.2	Testing		Complete	7	5/6/20	5/13/20	7	5/6/20	5/13/20	Luke,Dohun,Hyuk	
23	4.1.3	Defect Fixing		Complete	7	5/6/20	5/13/20	7	5/6/20	5/13/20	Shobhit, Jason	
24	4.2	Deliverable Iteration 1-Der	Milestone		0	5/14/20	5/14/20					
25	4.3	Iteration 2										
26	4.3.1	Development		To start	5	5/15/20	5/20/20				Shobhit,Jason	
27	4.3.2	Testing		To start	6	5/17/20	5/23/20				Luke,Dohun,Hyuk	
28	4.3.3	Defect Fixing		To start	5	5/17/20	5/22/20				Shobhit, Jason	
29	4.3	Deliverable -Iteration -2	Milestone				5/24/20	5/24/20				
30	5	Project Closing					5/25/20	5/25/20				

B. Project Charter

Project Charter

Project Title: SportDataExchange

Project Start Date: 2/18/2020

Projected Finish Date: 5/1/2020

Budget Information: TBD

Project Manager: Name, Shobhit Ratan phone, (516) 528-7905 e-mail

Shobhit.Ratan@baruchmail.cuny.edu

Project Objectives: Create a commercially viable web based application that allows users to predict the outcome of theoretical NFL Football Games and Future NFL Matchups based on prior statistics.

Main Project Success Criteria: Working system that is available at any time.

Approach: Team has responsibility to each other to meet agreed upon deadlines or let the team know that assistance is required to meet the deadline.

Roles and Responsibilities

Role	Name	Organization/ Position	Contact Information
CEO	Luke	Management and Finance	luke.weeks@baruchmail.cuny.edu
CTO	Shobhit	Management and Full-stack	shobhit.ratan@baruchmail.cuny.edu
CDO	Dohun	Systems Design	dohun.kim@baruchmail.cuny.edu
CFO	Hyuk	Finance	hyuk.kim2@baruchmil.cuny.edu
UX/QA	Jason Subhani	Front-End	jason.subhani@baruchmail.cuny.edu

Sign-off: (Signatures of all above stakeholders. Can sign by their names in table above.)

Comments: (Handwritten or typed comments from above stakeholders, if applicable)
One such requirement is that we respond to key stakeholder Thomas Licciardello in writing within 24 hours of any request.

C. Meeting Logs

Information Systems Development **Sport Data Exchange**

Team Meeting Log Sheet Group Project - Team 3 CIS 9590 - Section PTRA

Team Meeting Schedule

The details of Team meetings are as follows:

Meeting Date	Meeting Time	Type of Meeting
2/18/20	6PM	Physical Meeting in Library
2/24/20	6PM	Physical Meeting in Library
2/29/20	1PM	Video / Audio Call on Zoom
3/19/20	6PM	Video / Audio Call on Zoom
3/24/20	6PM	Video / Audio Call on Zoom

4/13/20	6PM	Video / Audio Call on Zoom
4/22/20	6PM	Slack/Zoom
4/28/20	6PM	Slack/Zoom
5/5/20	6PM	Slack/Zoom
5/12/20	6PM	Slack/Zoom
5/13/20	5:30 PM	Slack/Zoom

Meeting Agenda

The Agenda for group meetings on the following dates to complete the activities related to Phase 1 of the Final Project. Salient agenda points for team meeting were as follows:

Sl	Date	Agenda	Meeting type
1	2/18	<ul style="list-style-type: none"> ● Review and Respond to Professor's comments regarding idea submission. ● To finalize exact scope of work for final completion of the project ● To finalize exact scope of work for Phase 1 of the Project (Initial Team Presentation) ● Discuss activities and divide the work for Project Charter (that is only for the execution, all of us have to know everything only then we can deliver our part professionally and to the best satisfaction of all group members) ● Complete working on the Scope Statement ● Identify the Processes and Activities for building the Decomposition Diagram. ● Start building Requirements prior to Meeting #2. ● Start building Project Charter. 	Physical Meeting in Library

2	02/24/2020	<ul style="list-style-type: none"> ● To finalize exact scope of work for Phase 1 of the Project (Initial Team Presentation) ● Identify functional areas and processes for decomposition diagram ● Review the initial team presentation requirements <ul style="list-style-type: none"> ○ Define your project ○ System representation ○ Project & System Analysis ● Prepare required details for the initial team presentation (see CIS 9590 Final Project Guidelines) 	Physical meeting in Library
3	02/29/2020	<ul style="list-style-type: none"> ● Finalize Decomposition Diagram. ● Start preparing Requirements. ● Create a Final Report Template and layout. ● Start filling items corresponding to the Report that are complete. 	Video / Audio Call on Zoom
4	03/19/2020	<ul style="list-style-type: none"> ● Discuss risk mitigation after COVID-19 closure ● Discuss BRD ● Presentation planning ● Set 3/24 for emailing professor and next meeting ● Financial Modeling 	Video / Audio Call on Zoom

5	03/24/2020	<ul style="list-style-type: none"> ● Finalize presentation ● Luke took responsibility for writing script for presentation, and ● Agree on completion of slides 3/25/20 	Video / Audio Call on Zoom
6	4/13/2020	<ul style="list-style-type: none"> ● Slack software ● Ruby on Rails ● Jason is assigned to UI/Wireframes (2.5 weeks) ● Hyuk - procurement of documentation ● Dohun - Data Modelling (2.5 weeks) ● May 11th - prototype deadline ● Next meeting 4/20 6PM ● Have our data ready 	Video / Audio Call on Zoom
7	4/22/2020	<ul style="list-style-type: none"> ● Slack software ● Ruby on Rails ● Phase 5 division of work ● 5A/F- Luke ● 5B/C - Jason and Shobit ● 5D/E - Hyuk ● Aim for Tuesday 	Slack/ Zoom

8	4/28/20	<ul style="list-style-type: none"> ● Phase 5 work review we look good so far just need Hyuk to complete his section. ● Hyuk will do his phase section and we will go over on thursday first order of business ● Work on appendix items ● Deliverables for next meeting: <ul style="list-style-type: none"> ○ Phase 5 Hyuk section E priority one ○ Appendix materials idea and plan assignment ● Next meeting short meeting thursday 4/31 	Slack/Zoom
9	5/5/20	<ul style="list-style-type: none"> ● Luke and Hyuk: <ul style="list-style-type: none"> ○ Finish Report ○ Appendix Items ● Tech Team: <ul style="list-style-type: none"> ○ Finish Front End Development ○ Work on Variance Report (Appendix F) 	Slack/Zoom
10	5/12/20	<ul style="list-style-type: none"> ● Debug conference and team issues ● Add validation ● Change data model in accordance with development ● Developers will also be doing QA 	Slack/Zoom

11	5/13/20	<p>https://drive.google.com/open?id=19ARqc9Vga6pN1z2yFSWbkM3t6ZtxObd3</p> <p>https://drive.google.com/open?id=1qeYykX9FWHDBWnA0ElcCNyqprwXRRC0</p> <p>https://drive.google.com/open?id=10x0kWzChs1vU1xtQimVMoHyrmLLORBdD</p> <p>First link: word document of group project, second: project plan excel sheet, third: presentation slides we could also use for final presentation(maybe some edits needed?)</p>	Slack/Zoom
----	---------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------

Team Sport Data Exchange

Sl	Name
1	Jason Subhani
2	Dohun Kim
3	Shobhit Ratan
4	Hyuk Kim
5	Luke Weeks

D. Change Logs

Project : Sport Data Xchange
 Change Request – CR-001

Change #	1
Change Identification Date	5/10/20
Change Type	Scope
Change Description & Reason	Change in Scope for Iteration 1 deliverable. Theoretical Game Prediction and Future Win Prediction was in the scope for Milestone of 5/14/20 , but owing to technical limitations identified with Ruby on Rails. The design will be changed to implement it on React js.
Impact	<ol style="list-style-type: none"> 1. The complete backend implementation for Theoretical Game feature will be done in Iteration 2 – after 5/14/2020. 2. Iteration 1 Milestone will only have the front end hard coded values to demo the working. 3. There is no impact to overall cost. 4. Impact to efforts – 2 resources need to work in parallel to complete in Iteration 2.
Change Approval	Luke Weeks - CEO

E. Issues Risks Logs

#	Risk/Issue	Description	Probability of Risk	Impact	Impact Description	Mitigation Plan	Date of creation	Status
1	Issue	Initial presentation approval delays owing to Stay at Home order		Schedule	Schedule Over run and Delay in completion of the BRD and subsequent phases of project	Disruption of a few days to be compensated with extra hours	3/11/20	Closed
2	Risk	Impact of COVID-19 on project deliverables	High	Scope,Sch	Working remotely may delay communication with Stakeholders	Have regular meets	3/19/20	Closed
3	Risk	Technical feasibility of Ruby n Rails for betting logic implementation	High	Scope	The Scope of work agreed to demo for End semester may be impacted.	Agree with Group 10 days prior to finalization and raise change request	4/10/20	Closed
4	Issue	Repository corrupted -rework		Schedule	Start date for the development was pushed by 3 days	Cover up time lost by working extra hours and pair programming	5/5/20	Closed

F. Variance Report

Activities	Estimated Effort in Hrs	Actual Effort in Hrs	Variance in Efforts
Project Charter	30	30	0
BRD	360	440	80
Design	460	475	15
Development-Iteration1	100	150	50
Total	950	1095	145
Variance- Effort Over run			15%
Overall Schedule overrun for Iteration 1			0%