



CSCC01 Final Project

TODO: Team Name

Ryan Blasetti

Lazar Glumac

King Lai

Yara Radwan

Jaedon Wong

Aum Patel

Parth Patel

December 4, 2020

Table of Contents

Current CRC Cards.....3

 Front-End CRC Cards.....3

 Back-End CRC Cards.....7

System Design.....12

 Architecture Diagram of the System.....12

 Components of the Diagram.....12

Current CRC Cards

Front-End CRC Cards

EditProfile	EntryPage	SoloTrivia
Responsibilities <ul style="list-style-type: none"> - Display the My account page - Show user their biography and survey answers - Send request to update database - Send request to get user info 	Responsibilities <ul style="list-style-type: none"> - Renders login and register page - Prompts user for appropriate fields - Sends post requests to backend to validate data and adjust model - Stores username for use in the frontend of the application 	Responsibilities <ul style="list-style-type: none"> - Displays instructions for solo trivia. - Renders the quiz with the questions and answer options. - Displays the final quiz score. - Sends the user's quiz data to the backend to calculate ACS score.
Collaborators <ul style="list-style-type: none"> - React (useState, useEffect) - XMLHttpRequest - Popup 	Collaborators <ul style="list-style-type: none"> - { Route, NavLink, Router } - ForgotPassword - Index - React, { useState, Effect } - axios 	Collaborators <ul style="list-style-type: none"> - React, { useEffect, useState } - axios
Parent class: App	Parent class: index	Parent class: App.js
Child class(es): <ul style="list-style-type: none"> - Radar 	Child class(es): App	Child class(es): N/A

OnlinePostTrivia	OnlineChallengeTrivia	Index
Responsibilities <ul style="list-style-type: none"> - Allow user to post their trivia score. - Renders the quiz with the questions and answer options. - Displays the final quiz score. - Sends user quiz data to the backend to display for others to challenge. 	Responsibilities <ul style="list-style-type: none"> - Allow user to challenge another person's trivia score. - Renders the quiz with the questions and answer options. - Displays the final quiz score, winner and loser. - Sends quiz data to the backend to update the winner and loser's ACS. 	Responsibilities <ul style="list-style-type: none"> - Renders the entry page upon first visiting the URL
Collaborators <ul style="list-style-type: none"> - React, { useEffect, useState } - axios 	Collaborators <ul style="list-style-type: none"> - React, { useEffect, useState } - axios 	Collaborators <ul style="list-style-type: none"> - React - ReactDOM
Parent class: App.js	Parent class: App.js	Parent class:
Child class(es): N/A	Child class(es): N/A	Child class(es): Entry Page

App	TheZone	
Responsibilities <ul style="list-style-type: none"> - Hosts main page of app - Allows for navigation between different tabs - Displays current users' profile picture, name and ACS score - Displays a running ticker of weekly latest games 	Responsibilities <ul style="list-style-type: none"> - Display a feed for SPORTCRED users to view - Allow SPORTCRED users to create a Zone post - Allow SPORTCRED users to comment on a Zone post - Allow SPORTCRED users to agree/disagree on a Zone post - Send user posts, comments and agrees/disagrees to backend 	
Collaborators <ul style="list-style-type: none"> - React { useEffect } - { NavLink, Switch, Router } - axios 	Collaborators <ul style="list-style-type: none"> - React (useState, useEffect) - axios 	
Parent class: EntryPage	Parent class: App	
Child class(es): <ul style="list-style-type: none"> - EditProfile - SoloTrivia - OnlinePostTrivia - OnlineTriviaChallenge - PickPage - DebatePage - LoggedOut 	Child class(es):	

PickPage	TeamInBracket	ForgotPassword
Responsibilities <ul style="list-style-type: none"> - Hosts page to make picks - Allows user to submit NBA playoff picks - Display results which picks were right and wrong 	Responsibilities <ul style="list-style-type: none"> - Render the pick boxes - Displays selector with radio buttons if pick not made for the that specific tournament series - Display chosen team when choice made 	Responsibilities <ul style="list-style-type: none"> - Renders the "Forgot Password" page - Allows a user to submit their email address - Sends an email to the user containing their password
Collaborators <ul style="list-style-type: none"> - React { useState, useEffect } - XMLHttpRequest 	Collaborators	Collaborators <ul style="list-style-type: none"> - React, { useState } - EmailJS - XMLHttpRequest
Parent class: App	Parent class: PickPage	Parent class: EntryPage
Child class(es): <ul style="list-style-type: none"> - TeamInBracket 	Child class(es):	Child class(es):

DebatePage	Post	PostBox
Responsibilities <ul style="list-style-type: none"> - Allow user to post their opinion on a debate question. - Allow user to view a debate group's responses to a question. 	Responsibilities <ul style="list-style-type: none"> - Contains the pop-up which allows the user to respond to a debate question. 	Responsibilities <ul style="list-style-type: none"> - Let user pick a question and type a response. - Post the user's response to the database. - Warn user if they have already made a post for the day.
Collaborators <ul style="list-style-type: none"> - React, { useEffect, useState } - Axios - Popup 	Collaborators <ul style="list-style-type: none"> - React 	Collaborators <ul style="list-style-type: none"> - React, { useEffect, useState } - axios
Parent class: App.js	Parent class: DebatePage.js	Parent class: Post.js
Child class(es): <ul style="list-style-type: none"> - Post.js - FetchPosts.js 	Child class(es): <ul style="list-style-type: none"> - PostBox.js 	Child class(es): N/A

FetchPost	RatingSlider	Latest
Responsibilities <ul style="list-style-type: none"> - Retrieve posts from a debate group and display them for the user to view. - Let user rate posts via slider. 	Responsibilities <ul style="list-style-type: none"> - Allow using to rate a debate post by sliding a slider horizontally. 	Responsibilities <ul style="list-style-type: none"> - Fetches games from api is displays them - Filters games by team - Filters games by year
Collaborators <ul style="list-style-type: none"> - React, { useEffect, useState } - axios 	Collaborators <ul style="list-style-type: none"> - React - styled 	Collaborators <ul style="list-style-type: none"> - React { useEffect, useState } - axios
Parent class: DebatePage.js	Parent class: FetchPosts.js	Parent class: App
Child class(es): RatingSlider.js	Child class(es): N/A	Child class(es): N/A

LoggedOut	RadarBox	Radar
Responsibilities <ul style="list-style-type: none"> - Restrict the user from accessing their account after they log out - Notify the user that they logged out - Instruct the user on how to log back in 	Responsibilities <ul style="list-style-type: none"> - Allow user to view who's under their radar. - Allow user to search for other users to add to their radar. 	Responsibilities <ul style="list-style-type: none"> - Display a popup box in the middle of the screen which will contain the radar.
Collaborators <ul style="list-style-type: none"> - React 	Collaborators <ul style="list-style-type: none"> - React, { useEffect, useState} - axios 	Collaborators <ul style="list-style-type: none"> - React
Parent class: App	Parent class: Radar	Parent class: EditProfile
Child class(es): N/A	Child class(es): <ul style="list-style-type: none"> - FetchRadar 	Child class(es): <ul style="list-style-type: none"> - RadarBox

FetchRadar
Responsibilities <ul style="list-style-type: none"> - Retrieves a user's radar from the database and displays it in the radar box popup. - Displays the full profiles of the people under a user's radar. - Allow user to delete users from their radar.
Collaborators <ul style="list-style-type: none"> - React - axios
Parent class: RadarBox
Child class(es): N/A

Back-End CRC Cards

UpdateUserProfile	GetProfilePicture
Responsibilities <ul style="list-style-type: none"> - Handle requests to update user data 	Responsibilities <ul style="list-style-type: none"> - Handle requests to retrieve user profile picture
Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB
Parent class: HttpHandler	Parent class: HttpHandler
Child class(es):	Child class(es):

SportcredBackend	Neo4JDB
Responsibilities <ul style="list-style-type: none"> - Run the server for the application - Establish endpoints for HTTP requests 	Responsibilities <ul style="list-style-type: none"> - Represent the database for the application - Retrieve data from the database - Manipulate data from the database - Create/Manage nodes in the database
Collaborators <ul style="list-style-type: none"> - GetQuestions - AddQuestions - UserRegistration - UpdateUserProfile - GetProfilePicture - FinalSoloTriviaScore - UserLogin - NewChallenger - RoomList - GetOpponent - FinalDuoScore - GetACS - GetUserInfo - AddDebateQuestion - GetDebateQuestions - PlayoffBracket - PredictPlayoffBracket - CheckPlayoffPrediction - AddZoneComment - AddZonePost - GetZoneFeed - LikeDislikeZonePost - DebateGroup - ReturnDebateGroup - ResetDebate - ValidateDebate - ScoreDebates - UserVote - GetGameFeed - RecoverPassword - HttpServer - SearchRadar - EditRadar - RadarSize - GetRadar - GetWeekGames - GetProfile 	Collaborators <ul style="list-style-type: none"> - Driver - UserNode
Parent class:	Parent class:
Child class(es):	Child class(es):

AddQuestions	PredictPlayoffBracket	GetACS
Responsibilities <ul style="list-style-type: none"> - Accept requests to add new trivia questions to database - Handling for new questions sent to database 	Responsibilities <ul style="list-style-type: none"> - Handle requests to predict a playoff series for a specific user 	Responsibilities <ul style="list-style-type: none"> - Handle requests for ACS and send user's score and rank in return
Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - UserNode
Parent class: HttpHandler	Parent class: HttpHandler	Parent class: HttpHandler
Child class(es):	Child class(es):	Child class(es):

GetOpponent	GetQuestions	NewChallenger
Responsibilities <ul style="list-style-type: none"> - Handle requests to find room ID for head to head trivia - Send back room questions and opponent info 	Responsibilities <ul style="list-style-type: none"> - Handle requests to return trivia questions from database 	Responsibilities <ul style="list-style-type: none"> - Handles requests to create a new room for head to head trivia - Sends information to be handled by database to setup room info/room id
Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB
Parent class: HttpHandler	Parent class: HttpHandler	Parent class: HttpHandler
Child class(es):	Child class(es):	Child class(es):

PopulateDatabase	RoomList	UserLogin
Responsibilities <ul style="list-style-type: none"> - Send requests to server to add new questions to database (trivia/debate) - Clear and score debate rooms/questions for database for daily reset 	Responsibilities <ul style="list-style-type: none"> - Handles requests to retrieve available room id numbers/username for head to head trivia 	Responsibilities <ul style="list-style-type: none"> - Handles requests to login users based on username - Verifies and returns back whether login info is valid (matches credentials in database)
Collaborators <ul style="list-style-type: none"> - HttpURLConnection - URLConnection 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB
Parent class:	Parent class: HttpHandler	Parent class: HttpHandler
Child class(es):	Child class(es):	Child class(es):

UserNode	UserRegistration	Utils
Responsibilities <ul style="list-style-type: none"> - Represents a user node in database - Stores user info from database in fields and allows for manipulation by other classes - Updates individual scores for the user 	Responsibilities <ul style="list-style-type: none"> - Handles requests to create new users - Sends info to database to create new nodes - Returns back whether account creation was successful or not 	Responsibilities <ul style="list-style-type: none"> - Decodes incoming requests for use by code
Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators
Parent class:	Parent class: <code>HttpHandler</code>	Parent class:
Child class(es):	Child class(es):	Child class(es):

GetUserInfo	AddDebateQuestion	ReturnDebateGroup
Responsibilities <ul style="list-style-type: none"> - Handle http requests - Send a response with the user's survey answers and biography 	Responsibilities <ul style="list-style-type: none"> - Accept requests to add new debate questions to database - Handling for new questions sent to database 	Responsibilities <ul style="list-style-type: none"> - Sends back a group for the user to vote on in debates - Sends back all posts from decided group including usernames
Collaborators <ul style="list-style-type: none"> - UserNode 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB
Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>
Child class(es):	Child class(es):	Child class(es):

ValidateDebate	AddZoneComment	AddZonePost
Responsibilities <ul style="list-style-type: none"> - Sends back whether current user has already made a post for current debate topics 	Responsibilities <ul style="list-style-type: none"> - Adds a comment onto a post in the database given the author, content, and parent post 	Responsibilities <ul style="list-style-type: none"> - Adds a post to the database containing the author, content, comments, postID, likes and dislikes
Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB
Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>
Child class(es):	Child class(es):	Child class(es):

GetZoneFeed	LikeDislikeZonePost	RecoverPassword
Responsibilities <ul style="list-style-type: none"> - Returns all posts in the database with the content, author, comments - Returns which posts the user has already liked/disliked 	Responsibilities <ul style="list-style-type: none"> - Increments the number of dislikes/likes of a ZonePost in the database - Prevents double liking/disliking on a post, returning back and error code 	Responsibilities <ul style="list-style-type: none"> - Handle requests to recover a user's password
Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB
Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>
Child class(es):	Child class(es):	Child class(es):

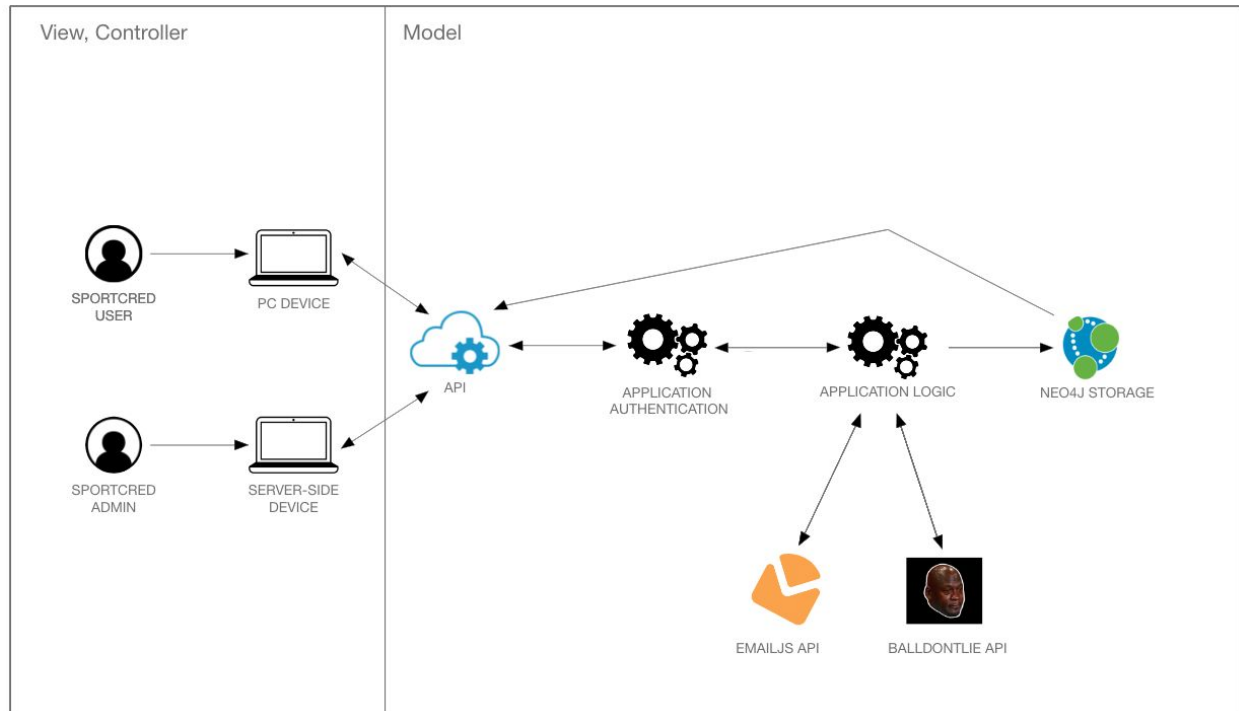
UserVote	ScoreDebates	GetGameFeed
Responsibilities <ul style="list-style-type: none"> - Handle requests to let a user vote on a debate post - Creates a relationship between a user and a post 	Responsibilities <ul style="list-style-type: none"> - Handle request to score debate - Initiates debate scoring period through password protect context 	Responsibilities <ul style="list-style-type: none"> - Handles request to receive list of games and scores to return based on selected season and team
Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB - <code>HttpURLConnection</code>
Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>
Child class(es):	Child class(es):	Child class(es):

GetWeekGames	RadarSize	EditRadar
Responsibilities <ul style="list-style-type: none"> - Creates a time interval of week, centering the current day as the 4th day of the week. - Handles get request to receive list of games and scores to return based on selected time period 	Responsibilities <ul style="list-style-type: none"> - Handles requests to check size of Radar - Sends back size of Radar to client 	Responsibilities <ul style="list-style-type: none"> - Allows client to send request for changes to radar - Allows to handles requests to add/remove users from radar
Collaborators <ul style="list-style-type: none"> - <code>DateTime</code> - <code>HttpURLConnection</code> 	Collaborators <ul style="list-style-type: none"> - Neo4JDB 	Collaborators <ul style="list-style-type: none"> - Neo4JDB
Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>	Parent class: <code>HttpHandler</code>
Child class(es):	Child class(es):	Child class(es):

SearchRadar	GetRadar	GetProfile
<div>Responsibilities<ul style="list-style-type: none">- Handles requests to check for users to add to radar- sends back info of users with the requested username</div>	<div>Responsibilities<ul style="list-style-type: none">- Handles requests to get all users on radar- Sends all radar info back to client</div>	<div>Responsibilities<ul style="list-style-type: none">- Handles requests to get profile of another user- Sends back non-vital profile information to be displayed on radar</div>
<div>Collaborators<ul style="list-style-type: none">- Neo4JDB</div>	<div>Collaborators<ul style="list-style-type: none">- Neo4JDB</div>	<div>Collaborators<ul style="list-style-type: none">- Neo4JDB</div>
<div>Parent class: HttpHandler</div>	<div>Parent class: HttpHandler</div>	<div>Parent class: HttpHandler</div>
<div>Child class(es):</div>	<div>Child class(es):</div>	<div>Child class(es):</div>

System Design

Architecture Diagram of the System



*Note: The image presented for balldontlie API is the official logo

Components of the Diagram:

- **Sportcred User:** A user that will be engaging with the React web-app (represents the view and controller)
- **Sportcred Admin:** A user from SPORTCRED that is able to make additions to the site from the program by having special privileges (in terms of sending HTTP requests)
- **PC Device:** A compatible device that can host the React web-app (represents the view and controller)
- **Server-Side Device:** A device authorized by SPORTCRED to carry out HTTP requests to make additions to the overall React web-app (including adding trivia questions, etc.)
- **API:** The network endpoints that are accessed by various HTTP requests
- **BallDontLie API:** The API used to access seasonal game information
- **EmailJS API:** The API used to send emails to SPORTCRED users regarding password recovery
- **Application Authentication:** The credentials of the user are base 64 encoded upon registration, and upon login decoded in order to validate the user's login
- **Application Logic:** The code that facilitates the exchange and modification of data from the Neo4j database
- **Neo4j Storage:** The designated backend database to hold all user information