CSCC01 Final Project

TODO: Team Name

Ryan Blasetti
Lazar Glumac
King Lai
Yara Radwan
Jaedon Wong
Aum Patel
Parth Patel

November 20, 2020

Table of Contents

Current CRC Cards

Front-End CRC Cards

EditProfile
Responsibilities - Display the My account page - Show user their biography and survey answers - Send request to update database - Send request to get user info
Collaborators - React (useState, useEffect) - xmlHtmlRequest
Parent class: App
Child class(es):

EntryPage
Responsibilities - 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
Collaborators - { Route, NavLink } - ForgotPassword - index
Parent class: index
Child class(es): App

SoloTrivia
Responsibilities - 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 - React, { useEffect, useState } - axios
Parent class: App.js
Child class(es): N/A

OnlinePostTrivia
Responsibilities - 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.
Collaborators - React, { useEffect, useState } - axios
Parent class: App.js
Child class(es): N/A

OnlineChallengeTrivia
Responsibilities - 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.
Collaborators - React, { useEffect, useState } - axios
Parent class: App.js
Child class(es): N/A

Index
Responsibilities - Renders the entry page upon first visiting the URL
Collaborators - React - ReactDOM
Parent class:
Child class(es): Entry Page

App
Responsibilities - Hosts main page of app - Allows for navigation between different tabs - Displays current users' profile picture, name and ACS score
Collaborators - React { useEffect } - { Navlink, Switch, Router } - axios
Parent class: EntryPage
Child class(es): - EditProfile - SoloTrivia - OnlinePostTrivia - OnlineTriviaChallenge - PickPage - DebatePage

TheZone
Responsibilities - 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 - React (useState, useEffect) - axios
Parent class: App
Child class(es):

PickPage
Responsibilities - Hosts page to make picks - Allows user to submit NBA playoff picks - Display results which picks were right and wrong
Collaborators - React { useState, useEffect } - XMLHttpRequest
Parent class: App
Child class(es): - TeamInBracket

TeamInBracket
Responsibilities - 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
Collaborators
Parent class: PickPage
Child class(es):

ForgotPassword
Responsibilities - Renders the "Forgot Password" page - Allows a user to submit their email address - Sends an email to the user containing their password
Collaborators - React, { useState } - EmailJS - XMLHttpRequest
Parent class: EntryPage
Child class(es):

DebatePage
Responsibilities - Allow user to post their opinion on a debate question. - Allow user to view a debate group's responses to a question.
Collaborators - React, { useEffect, useState } - Axios - Popup
Parent class: App.js
Child class(es): - Post.js - FetchPosts.js

Post
Responsibilities - Contains the pop-up which allows the user to respond to a debate question.
Collaborators - React
Parent class: DebatePage.js
Child class(es): - PostBox.js

PostBox
Responsibilities - 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 - React, { useEffect, useState } - axios
Parent class: Post.js
Child class(es): N/A

FetchPost
Responsibilities - Retrieve posts from a debate group and display them for the user to view. - Let user rate posts via slider.
Collaborators - React, { useEffect, useState } - axios
Parent class: DebatePage.js
Child class(es): RatingSlider.js

RatingSlider
Responsibilities - Allow using to rate a debate post by sliding a slider horizontally.
Collaborators - React - styled
Parent class: FetchPosts.js
Child class(es): N/A

Latest
Responsibilities - Fetches games from api is displays them - Filters games by team - Filters games by year
Collaborators - React { useEffect, useState } - axios
Parent class: App
Child class(es): N/A

Back-End CRC Cards

UpdateUserProfile
Responsibilities - Handle requests to update user data
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

GetProfilePicture
Responsibilities - Handle requests to retrieve user profile picture
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

SportcredBackend
Responsibilities - Run the server for the application - Establish endpoints for HTTP requests
Collaborators - 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
Parent class:
Child class(es):

Neo4JDB
Responsibilities - Represent the database for the application - Retrieve data from the database - Manipulate data from the database - Create/Manage nodes in the database
Collaborators - Driver - UserNode
Parent class:
Child class(es):

AddQuestions
Responsibilities - Accept requests to add new trivia questions to database - Handling for new questions sent to database
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

PredictPlayoffBracket
Responsibilities - Handle requests to predict a playoff series for a specific user
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

GetACS
Responsibilities - Handle requests for ACS and send user's score and rank in return
Collaborators - UserNode
Parent class: HttpHandler
Child class(es):

GetOpponent
Responsibilities - Handle requests to find room ID for head to head trivia - Send back room questions and opponent info
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

GetQuestions
Responsibilities - Handle requests to return trivia questions from database
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

NewChallenger
Responsibilities - 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 - Neo4JDB
Parent class: HttpHandler
Child class(es):

PopulateDatabase
Responsibilities - Send requests to server to add new questions to database (trivia/debate) - Clear and score debate rooms/questions for database for daily reset
Collaborators - HttpURLConnection - URLConnection
Parent class:
Child class(es):

RoomList
Responsibilities - Handles requests to retrieve available room id numbers/usernames for head to head trivia
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

UserLogin
Responsibilities - Handles requests to login users based on username - Verifies and returns back whether login info is valid (matches credentials in database)
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

UserNode	UserRegistration	Utils
Responsibilities - 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 - Handles requests to create new users - Sends info to database to create new nodes - Returns back whether account creation was successful or not	Responsibilities - Decodes incoming requests for use by code
Collaborators - Neo4JDB	Collaborators - Neo4JDB	Collaborators
Parent class:	Parent class: HttpHandler	Parent class:
Child class(es):	Child class(es):	Child class(es):

GetUserInfo	AddDebateQuestion	ReturnDebateGroup
Responsibilities - Handle http requests - Send a response with the user's survey answers and biography	Responsibilities - Accept requests to add new debate questions to database - Handling for new questions sent to database	Responsibilities - Sends back a group for the user to vote on in debates - Sends back all posts from decided group including usernames
Collaborators - UserNode	Collaborators - Neo4JDB	Collaborators - Neo4JDB
Parent class:	Parent class: HttpHandler	Parent class: HttpHandler
Child class(es):	Child class(es):	Child class(es):

ValidateDebate	AddZoneComment	AddZonePost
Responsibilities - Sends back whether current user has already made a post for current debate topics	Responsibilities - Adds a comment onto a post in the database given the author, content, and parent post	Responsibilities - Adds a post to the database containing the author, content, comments, postID, likes and dislikes
Collaborators - Neo4JDB	Collaborators - Neo4JDB	Collaborators - Neo4JDB
Parent class: HttpHandler	Parent class: HttpHandler	Parent class: HttpHandler
Child class(es):	Child class(es):	Child class(es):

GetZoneFeed
Responsibilities - Returns all posts in the database with the content, author, comments - Returns which posts the user has already liked/disliked
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

LikeDislikeZonePost
Responsibilities - Increments the number of dislikes/likes of a ZonePost in the database - Prevents double liking/disliking on a post, returning back and error code
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

RecoverPassword
Responsibilities - Handle requests to recover a user's password
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

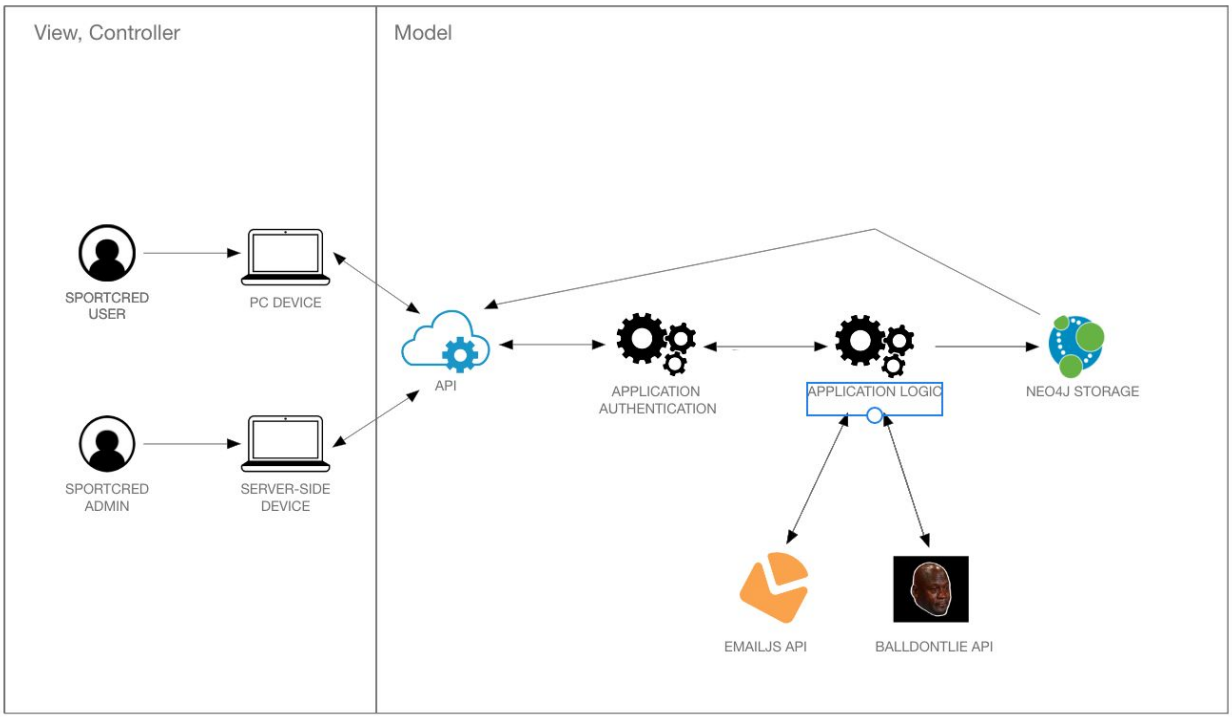
UserVote
Responsibilities - Handle requests to let a user vote on a debate post - Creates a relationship between a user and a post
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

ScoreDebates
Responsibilities - Handle request to score debate - Initiates debate scoring period through password protect context
Collaborators - Neo4JDB
Parent class: HttpHandler
Child class(es):

GetGameFeed
Responsibilities - Handles request to receive list of games and scores to return based on selected season and team
Collaborators - Neo4JDB - HttpUrlConnection
Parent class: HttpHandler
Child class(es):

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