**Predicting MLS Results**



Alexandre Moreira Batista - 300344513

Armando Defendi Rossi - 300340211

Applied Research Project – 4495-001

Stephen Chiong

Douglas College

Table of Contents

[Introduction 3](#_Toc130979380)

[Executive summary 3](#_Toc130979381)

[Statement of the problem 3](#_Toc130979382)

[Significance of the study 3](#_Toc130979383)

[Technologies 3](#_Toc130979384)

[Project timelines 4](#_Toc130979385)

[Project implementation 4](#_Toc130979386)

[Development and testing of database architecture 4](#_Toc130979387)

[GitHub configuration 6](#_Toc130979388)

[Development of web scrapping 7](#_Toc130979389)

[Development of the Android app 8](#_Toc130979390)

[Web scrapping for machine learning 11](#_Toc130979391)

[Plots and descriptive analysis 14](#_Toc130979392)

[Data cleaning and preparation 17](#_Toc130979393)

[Modeling data to features engineering 19](#_Toc130979394)

[Machine learning models 22](#_Toc130979395)

[How to setup 24](#_Toc130979396)

[Software Design architecture 26](#_Toc130979397)

[Front end 26](#_Toc130979398)

[Back end 26](#_Toc130979399)

[Integration and development 26](#_Toc130979400)

[Tests 27](#_Toc130979401)

[Workload distribution 28](#_Toc130979402)

[Citations 28](#_Toc130979403)

[Progress reports 28](#_Toc130979404)

# Introduction

## Executive summary

This project aims to give sports fans accurate predictions for Major League Soccer (MLS) games using robust machine-learning models. By utilizing web scraping, database management, machine learning analysis, and an Android application, this project will deliver valuable predictions to users without needing them to pay exorbitant fees to sports betting companies. This project will cover all aspects of the Data Analytics program at Douglas College. The primary beneficiaries will be soccer fans with access to accurate game predictions at no cost.

## Statement of the problem

The sports betting industry has seen significant growth recently, with large companies and start-ups competing for customers and revenue. While many companies offer predictions for game results, access to this valuable information often comes at a cost. This project aims to provide a cost-effective solution for all sports fans to access accurate predictions for MLS games.

## Significance of the study

This project aims to deliver predictions that improve the odds ratio when determining the winner of a game. By utilizing techniques such as web scraping, database management, machine learning, and analytics, this project will also identify features that impact the accuracy of predictions. Additionally, this project will deliver an Android application, providing users with easy access to past and future game predictions.

## Technologies

The project will make use of the following technologies:

* Programming Language: Python for web scraping and data prediction.
* Database: MySQL
* Front-end and Backend: Android Studio

## Project timelines

* Development and testing of database architecture to receive data - 1 week.
* Development and testing of Python scripts for web scraping - 2 weeks.
* Research on modelling and machine learning uses in soccer results prediction - 2 weeks.
* Preparation and setup of a virtual machine in the cloud (Azure) to run the database and modelling - 1 week.
* Data wrangling in Python - 1 week.
* Data modelling (machine learning to predict and check prediction accuracy) - 2 weeks.
* Development of the Android App - 4 weeks.
* Testing the entire ecosystem and fixing any bugs - 1 week.

## Project implementation

### Development and testing of database architecture

After conducting experiments on Azure SQL (as illustrated in Figure 01), our team decided to utilize MySQL instead, as presented in Figure 02. This web-based solution is easily accessible via Python scripts (for web scraping) and our Android application.

During the development of our Android application and data collection from the MLS website, we observed frequent updates to the website, posing a significant challenge to our web scraping approach. Due to the use of JavaScript in constructing pages, changes occurred rapidly, necessitating a switch to a more stable data source. Thus, we opted to leverage ESPN's website, which offers comparable data. We made this transition when updating our database with the latest scores of recent matches. As a result, scores are now updated one day after the matches are played.

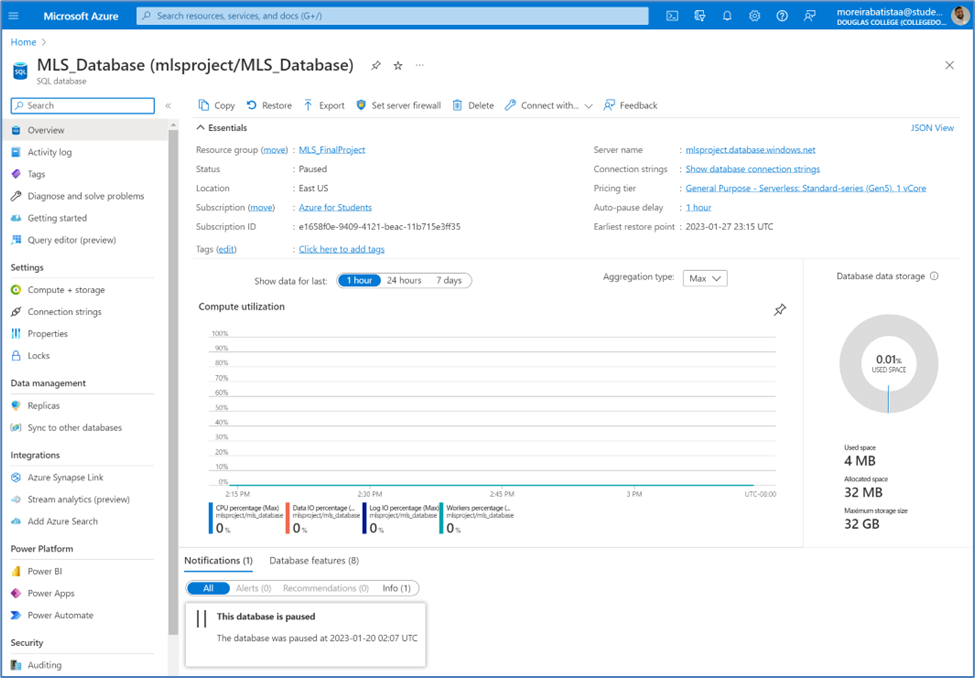


Figure 1 - Azure SQL Server

Graphical user interface

Description automatically generated with low confidenceGraphical user interface

Description automatically generated with medium confidence

Figure 2 - MySQL structure

The first tables were created, and the connection tested.

Database configuration: Alexandre on 16-20/Jan/2023

Database creation of tables/fields: Armando on 16-20/Jan/2023

### GitHub configuration

A repository was created to manage the versioning and save all the scripts and codes online as shown on figure 03.

GitHub creation: Armando on 16-20/Jan/2023

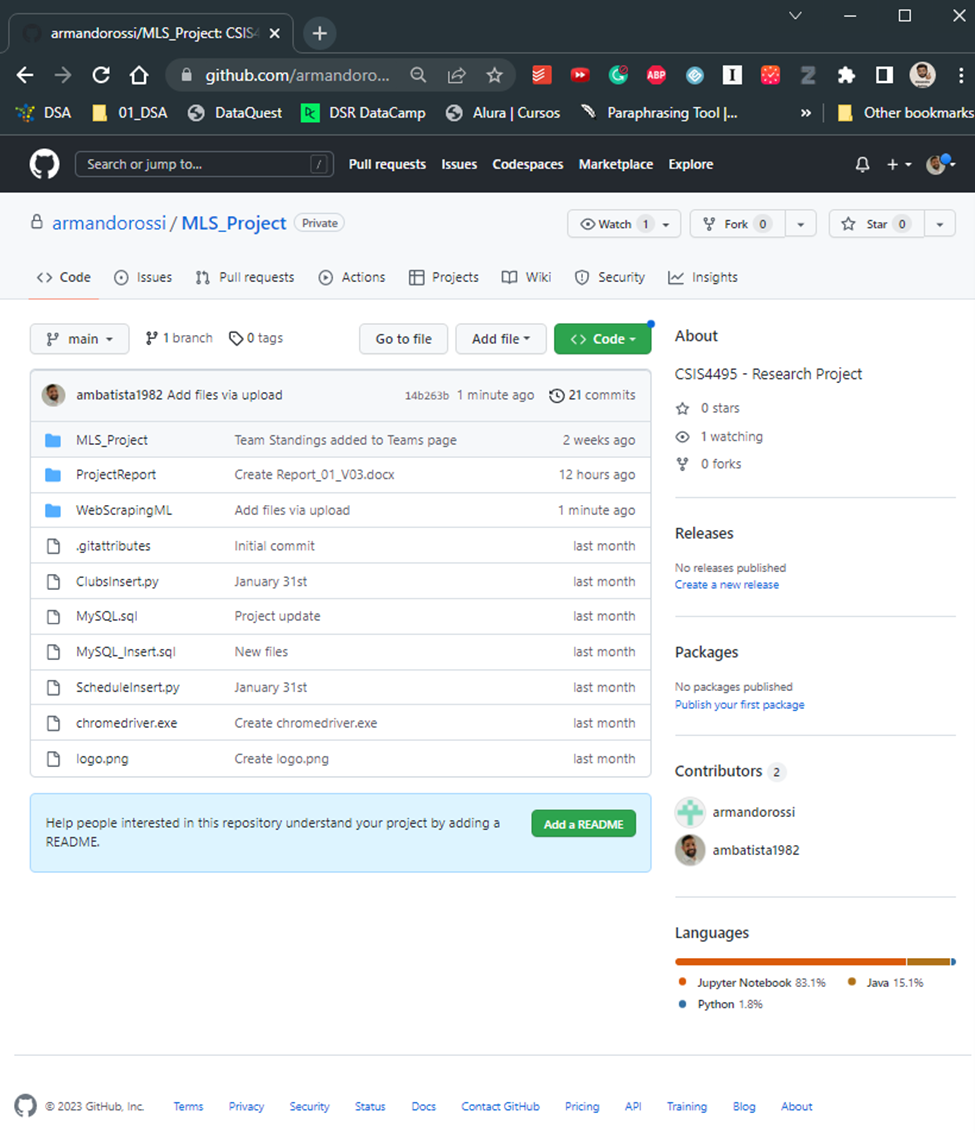


Figure 3 - Github directory

### Development of web scrapping

Using certain Python libraries, the scripts were created to extract schedule, team, and standings information from both the MLS Soccer and ESPN websites.

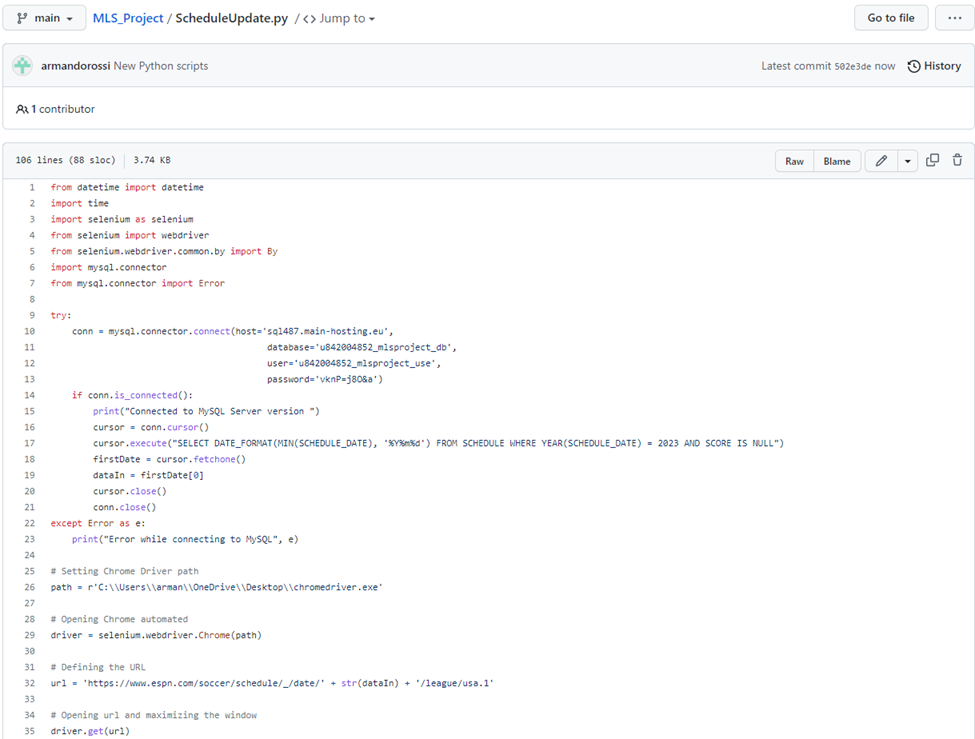


Figure 4 - Python script

A recently developed Python script has been updated to incorporate the latest changes. This updated version can now effectively insert, and update schedules based on the date provided, retrieving schedule information from the ESPN website.

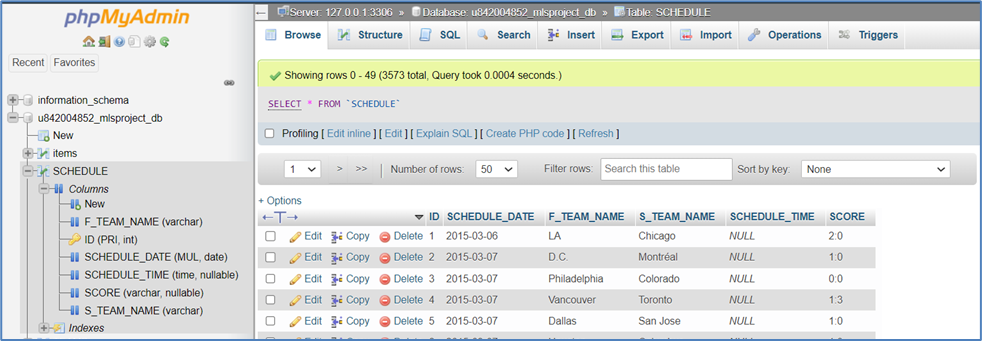


Figure 5 - Schedule table

Python scripts: Armando on 23-26/Jan/2023

### Development of the Android app

The first visuals were developed, including the Login, Register, Schedule, and User administration page layout.

|  |  |  |
| --- | --- | --- |
| Graphical user interface, text, application, email  Description automatically generated  Figure 6 - Login page | Graphical user interface, text, application, email  Description automatically generated  Figure 7 - Registration page | |
| Graphical user interface, application  Description automatically generated  Figure 8 - User page | | Graphical user interface, application  Description automatically generated  Figure 9 - Schedule page | |

To accommodate the upcoming modifications, the application has undergone a revamp. A fresh menu has been introduced to enable users to switch between the User (admin access only), Schedule, and the newly developed Team screen. The Team screen showcases a comprehensive list of all MLS teams and their standings. It also allows users to choose a specific year to review each team's standings.

|  |  |
| --- | --- |
| Figure 10 - Menu | Graphical user interface, application  Description automatically generated  Figure 11 - Teams page |

The Teams page now features an added functionality that enables users to view the upcoming scheduled matches for their teams by simply clicking on them.

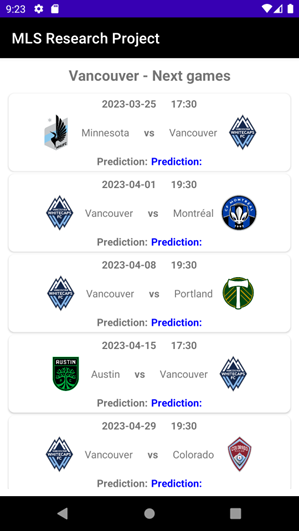


Figure 12 - Team schedule page

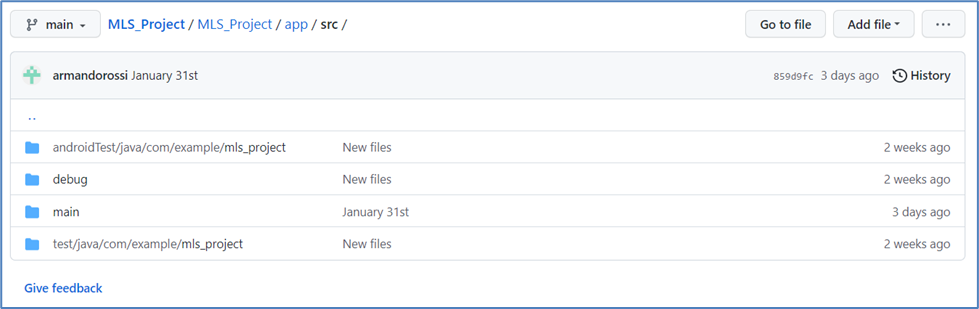


Figure 13 - Github repository

App Development: Armando on 23/Jan to 13/Mar/2023

### Web scrapping for machine learning

To enhance the machine learning model's prediction accuracy, it was concluded that supplementary data beyond game outcomes was required. Consequently, Python scripts were developed to extract data from the FB Ref website.

The structure of the Machine Learning Scripts was create using the following steps:

* Import the additional Libs
* Defining the standard Site Addresses
* Creating Function to Extract the Data from the site
* Creating SQL Functions to Query, Insert and Truncate tables
* Function GetInsert (grouping together all Functions)
* Testing Functions
* Get and Inserting Data from web to Database (2015-2023)

*Graphical user interface, text, application

Description automatically generated*

Figure 14 - Python web scrapping for ML

Text

Description automatically generated

Figure 15 - Python web scrapping for ML

Graphical user interface, text, application

Description automatically generated

Figure 16 - Python web scrapping for ML

Web scraping Development: Alexandre on 23/Jan to 26/Feb/2023

### Plots and descriptive analysis

Graphical user interface, text

Description automatically generated with medium confidence

Figure 17 - Python web scrapping for ML

A picture containing text

Description automatically generated

Figure 18 - Python web scrapping for ML

Graphical user interface, application, table

Description automatically generated

Figure 19 - Python web scrapping for ML

Chart, line chart

Description automatically generated

Figure 20 - Python web scrapping for ML

### Data cleaning and preparation

Text

Description automatically generated

Figure 21 - Python web scrapping for ML

A picture containing calendar

Description automatically generated

Figure 22 - Python web scrapping for ML

Table

Description automatically generated

Figure 23 - Python web scrapping for ML

Text

Description automatically generated

Figure 24 - Python web scrapping for ML

Data analysis and cleaning: Alexandre on 26/Feb to 12/Mar/2023

### Modeling data to features engineering

Text

Description automatically generated

Figure 25 - Python for ML

Graphical user interface, application, table

Description automatically generated

Figure 26 - Python for ML

A picture containing table

Description automatically generated

Figure 27 - Python for ML

### Machine learning models

Text

Description automatically generated

Figure 28 - Python for ML

Text

Description automatically generated

Figure 29 - Python for ML

The dataset has been divided in 4 groups:

G01 => Training data: 2016-2018 and Testing Data: 2019

G02 => Training data: 2016-2019 and Testing Data: 2020

G03 => Training data: 2016-2020 and Testing Data: 2021

G04 => Training data: 2016-2021 and Testing Data: 2022

The scores shown in the picture above are the results of the machine learning classifier models on the Group 01 of the Split dataset. The goal of the classification task was to predict the classes of the samples, which belong to one of three classes: 'H', 'A', or 'D'. This is an example of a multi-class classification problem.

Multi-class classification can be more challenging than binary classification because there are more possible classes that the model needs to distinguish between. In this case, the main difficulty was finding the optimal hyperparameters for the machine learning classifier models. Hyperparameters are parameters that are set before training the model and affect the learning process and the model's ability to generalize to new data.

In addition to hyperparameter tuning, other techniques can be used to improve the performance of multi-class classifiers. For example, ensemble methods, such as bagging and boosting, can combine multiple classifiers and improve their performance. This option will be tested in the next steps of the project.

Next steps: - 13. Measure accuracies, create Predictions and export to Mobile Applications.

## How to setup

To access our database, you can configure a new connection in MySQL Workbench as described above since it is hosted online.

* Hostname: sql487.main-hosting.eu
* Port: 3306
* Username: u842004852\_mlsproject\_use
* Password: vknP=j8O&a

To execute our Android application project developed on Android Studio Electric Eel | 2022.1.1 Patch 1, please follow these steps:

* Extract the folder MLS\_Project\_Android.zip
* Open Android Studio
* Go to File and select Open, then choose the folder MLS\_Project and click OK
* In the Project section, expand the Gradle Scripts folder and open the file local.properties
* Copy and paste the following lines into the local.properties file

URL = "jdbc:mysql://sql487.main-hosting.eu/u842004852\_mlsproject\_db"  
USER = "u842004852\_mlsproject\_use"  
KEY = "vknP=j8O&a"

Text

Description automatically generated

Figure 30 - Local properties file

These steps need to be followed in order to create a virtual device and run the app:

* Open the Device Manager
* Click on "Create device"
* Choose "Pixel 2" and click "Next"
* If necessary, download R by clicking the download button (Image 31) next to the R option, then click "Next" and "Finish"
* Click on the "Run" button and wait for the virtual device to launch

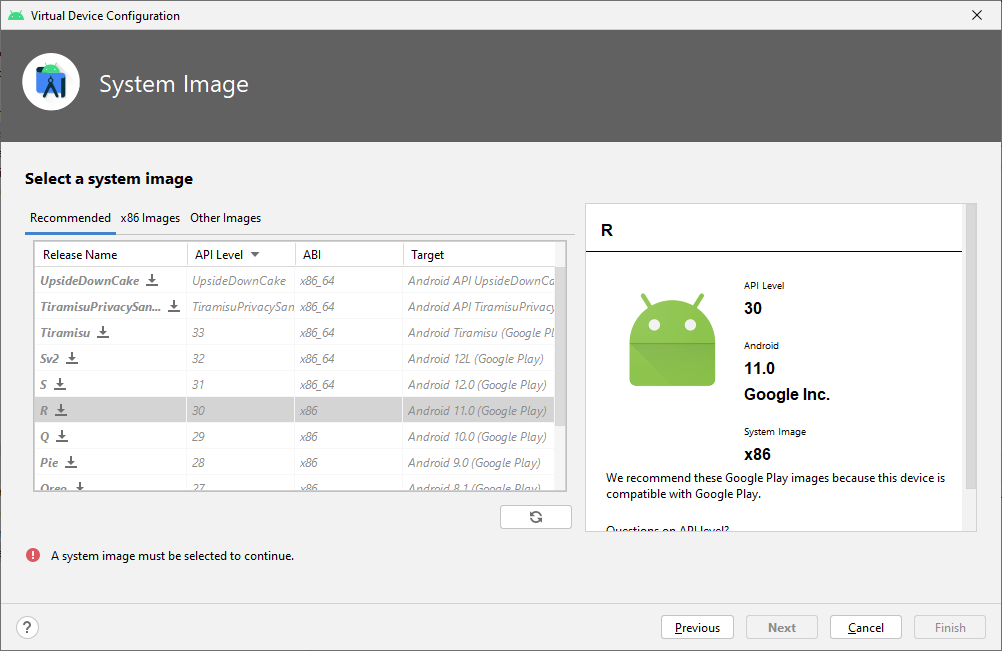


Figure 31 - R installation page

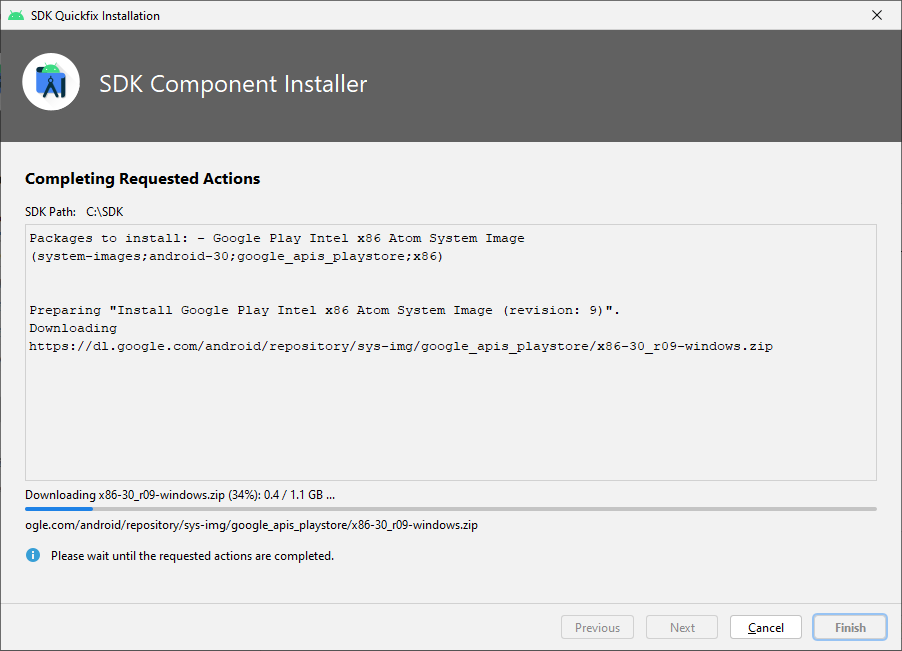


Figure 32 - SDK component installer

Admin credentials

* Username: admin@admin.com
* Password: Admin@123

## Software Design architecture

### Front end

* Android application for users to access the sports predictions and other features.
* User interface design to allow easy navigation and interaction.

### Back end

* Web scraping module to collect data on MLS games, including past results and upcoming fixtures.
* Machine learning module to analyze the data and generate accurate predictions for future games.
* Database management module to store and manage the data collected from web scraping and machine learning modules.
* Analytics module to identify the features that impact the accuracy of predictions.
* API layer to expose the prediction data and other features to the Android application.

### Integration and development

* Integration of the front-end and back-end modules to create a seamless user experience.
* Deployment of the application on cloud platforms or other hosting environments to ensure scalability and availability.

The software design architecture depicted above adopts a client-server style, in which the Android application serves as the client while the back-end modules function as the server. The system will gather and analyze data by leveraging web scraping, machine learning, and database management techniques, enabling it to generate precise predictions for MLS games.

Furthermore, the analytics module will pinpoint the factors that influence the precision of the predictions. The Android application will offer effortless access to past and future game predictions and other functionalities. The system will be deployed on cloud platforms or other hosting environments to ensure scalability and availability.

The Android app includes several models that are used to connect the data to the application's views. The models are ScheduleAdapter, TeamAdapter, and UserAdapter. There are also several classes that include LoginActivity, RegisterActivity, ScheduleActivity, TeamActivity, TeamScheduleActivity, and UserActivity. These classes are responsible for handling different aspects of the MLS Project's functionality. It also includes a database component, represented by ConnectionSQL and HashPassword. ConnectionSQL is responsible for handling the connection between the MLS Project and the database, while HashPassword is responsible for hashing user passwords to protect their sensitive information.

## Tests

During the database development process, we encountered challenges with Azure SQL, which charged a fee to create an instance and was difficult to access through Python and Android Studio. Therefore, we made the decision to switch to MySQL. Also, the database structure is consistent and has been built to store all MLS data required for our project and the Machine Learning processing results.

As we progressed with developing Python codes for web scraping, we faced issues with the MLS website, which was frequently updated due to its use of JavaScript for building the website in real-time. The need for frequent updates to our code led us to switch to the ESPN website, which contained similar data but did not use JavaScript for building the site in real-time.

Our primary focus while building the app was to ensure user accessibility. We aimed to create a simple and clear interface that displayed all information on the screen, without any hidden functionality. To evaluate user acceptance, we conducted an in-person survey with some classmates, and we received excellent feedback on the app.

## Workload distribution

|  |  |  |
| --- | --- | --- |
| **Task title** | **Assignee** | **Efforts** |
| Creation of online database | Alexandre | 3 days |
| Development and testing of database architecture | Armando | 1 week |
| Research on modelling and machine learning uses | Alexandre | 2 weeks |
| Preparation and setup of a virtual machine | Alexandre | 1 week |
| Data wrangling in Python | Alexandre | 1 week |
| Data modelling | Alexandre | 1 week |
| Development of the Android App | Armando | 4 weeks |
| Testing the entire ecosystem and fixing any bugs | Armando | 1 week |

## Citations

Major Soccer League. Schedule & Scores: <https://www.mlssoccer.com/schedule>

FB Ref – Soccer References Databases: [https://fbref.com/en/comps/22/history/Major-League-Soccer-Seasons](https://www.mlssoccer.com/schedule)

ESPN. Schedule & Scores: [https://www.espn.com/soccer/schedule/\_/league/usa.1](https://www.mlssoccer.com/schedule)

## Progress reports