# Overview

Create a program to assist with the process of ranking CFB teams each week. Short-term the program will collate the data and perform an initial pass to assist me with determing rankings. Long-term, the program may be a standalone ranking system.

# Minimum Viable Product

* Obtain data necessary for basic ranking calculation (weekly game data)
* Store the data for each team
* Output the ranking results in a .csv file

# Long-term Enhancements

* Web-server to host a front-end for managing data collection and distribution
* Increase complexity of ranking algorithm so that can provide definitive ranking
* Containerized so that the entire system can be hosted on any server
* Include outside polls for comparison/use

# Application

## Workflow

1. Obtain game data for all games in a given week
2. For each game, determine the winning and losing team, point differential and location
3. Update win/loss record for team in DB
   1. How can I get the win/loss streak if I store the data that way? What if I had a streak column that started 1 to -1 and onwards?
   2. It’ll be tricky for me to capture trends in a team’s performance if I don’t store the weekly game data in some fashion.
   3. What’s the best way to store a record if not using a team record class.
4. Add opponent played in most recent week to DB (array or dict) and return updated list of all opponents played to object attribute
5. Get wins/losses for each opponent from DB
6. Calculate opponent win/loss %
7. Store overall win % in object attribute and in DB (along with win-loss record for given week)
8. Output win-loss record and opponent win % as csv

### Output

* Write calculated ranking to database
* Write calculated win-loss to database
* Write complete rankings for current week to .csv file

### Classes

#### Team

* Attributes
  + ID, Name, Record
* Methods
  + Update Wins
  + Update Losses
  + Get wins
  + Get losses
  + Update opponent list
  + Get opponent list

#### Record

Record is an aggregated total of the types of wins/losses, opponents and point differential for a given season. Each team will have one record per season.

* Attributes
  + Home/Away/Neutral Wins/Losses
  + Season
  + Opponents
  + Point differential
* Methods
  + Update opponent list
  + Get opponent list

### Modules

#### DB Manager?

* Open connection
* Method that accepts sql statement, executes it and returns result
* Method that accepts sql statement, executes it (redundant?)

#### Game Fetcher

* Get game info from API
* Write to DB

#### Ranking

* Get list of teams
* Do ranking

## Database

Teams - Generic, mostly static team information

|  |  |  |  |
| --- | --- | --- | --- |
| TeamID | Name | Conference | Location Capacity |
| 001 | Michigan | Big10 | 100000 |
| 002 | Rutgers | Big10 | 50000 |

Record - Aggregate of each team’s record for a given season

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TeamID | Home Wins | Opponents | Point Diff | Season |
| 001 | 3 | Rutgers, Purdue | 30 | 2021 |
| 002 | 5 | Rutgers, PeenSt | 10 | 2020 |

Ranking - Output of my weekly rankings for each team

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TeamID | Year | Week | MyRanking | Wins | Losses | Opponents win % |
| 001 | 2020 | 1 | 5 | 1 | 0 | 100% |
| 001 | 2020 | 2 | 20 | 1 | 1 | 75% |

***MAY NOT BE NEEDED*** Games - Details for each game of the week

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GameID | Week | Date | HomeTeam | AwayTeam | HomeTeam Points | AwayTeam Points | NeutralSite | Overtime |
| 456 | 1 | 9/14/2020 | 001 | 002 | 24 | 7 | false | true |
| 457 | 1 | 9/14/2020 | 003 | 004 | 13 | 30 | false | false |