Serena Jiao, Constance Wang, Aydan Gooneratne, Henry Chen

Milestone 2

In this phase, you will set up your version control environment, explain your project idea in more detail, and assign responsibilities to each group member.

First, make sure to create a GitHub repository and share it with your Project Mentor. During development, you will use the repository to share code and perform version control.
Your project mentor will also use the repository during grading to review the quality of your code and ensure you haven't plagiarized any of it.

Next, write your project outline. The outline should contain the following:

1. Motivation for the idea/description of the problem the application solves

The chess community has two main groups - people who play officially rated tournaments under a chess organization like FIDE (Fédération Internationale des Échecs) usually in in-person venues, and people who play chess online on popular websites like Lichess.org, which have their own profiles and rating systems.

1) We want to connect the two databases, to allow users to compare and contrast between 1 online user's profile and 1 official tournament player's profile, and possibly be able to predict players' online handles and/or analyze how similar they are based on skill progression, openings, and demographics. A player could use our database to search for another tournament player's online games or to speculate what their online username might be.

2) We're focusing primarily on part 1, but as an extension, the community is also missing a lot of insights on different online and official tournament play in general. We can compare aggregate values like average rating difference (between a player's online and tournament ratings), draw rates, etc.

2. List of features you will definitely implement in the application
- Pull up an online and official player profile side by side
- Analyze online and official player similarity based on the following criteria
    - User data, including nationality, gender, age if available
    - Rating analysis, comparing the historical progression of rating
        - We can display both rating progressions together in line graph
- Be able to play through some samples of the players' games in chessboard UI
- Search other players by elo, username, location(?)

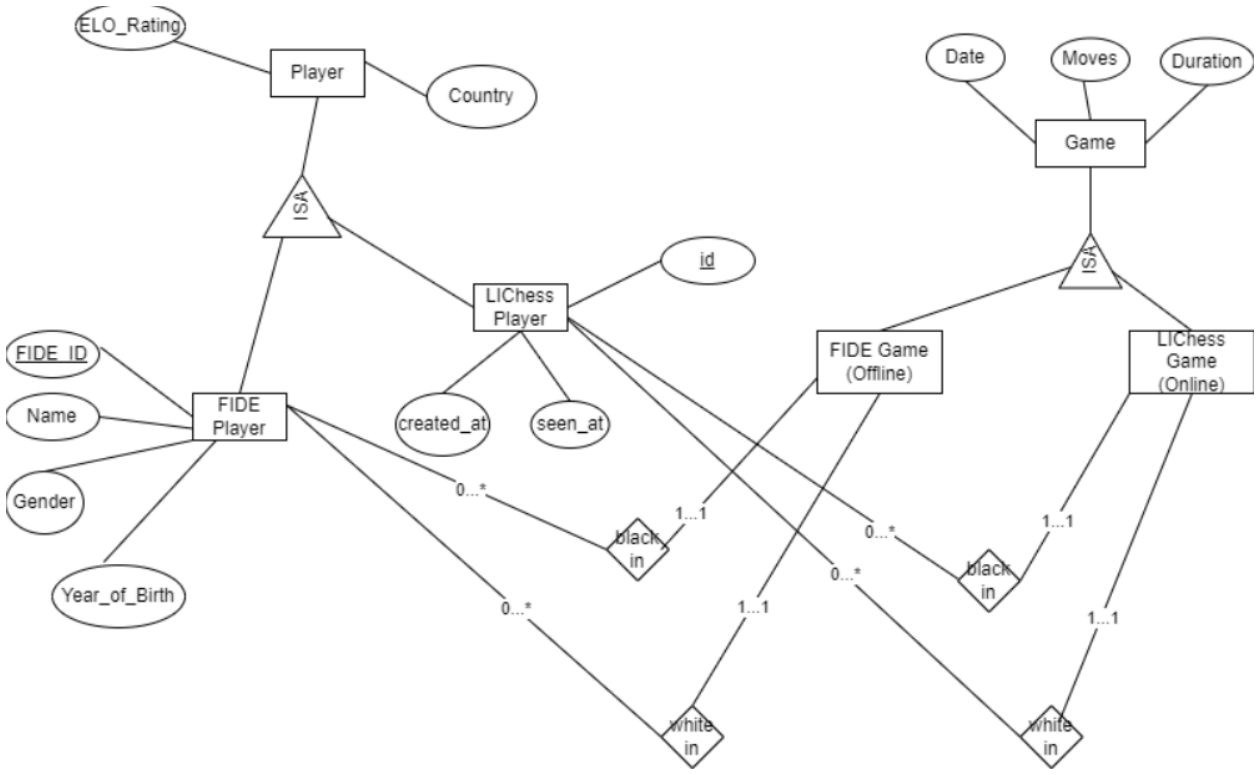
3. List of features you might implement in the application, given enough time

- Extension - using known chess players' online usernames as labels for a machine learning model that would predict whether a given chess player could be player under a given online username
    - Can find these mappings on the section of lichess profile where you can link to a fide profile

4. List of pages the application will have and a 1-2 sentence description of each page. We expect that the functionality of each page will be meaningfully different than the functionality of the other pages.

- Comparison page that has most of core analysis functionality
    - Shows an online and official username side by side
- Search and filter page
    - A page to search for an online/official player filtered by name, elo, country, etc. The search returns a list of matching players with links to their profile page.
- Directory of online players page
- Directory of official players page

5. Relational schema as an ER diagram



6. SQL DDL for creating the database

CREATE DATABASE chess;
USE chess;

```
CREATE TABLE FIDE_Player (
        FIDE_ID   INT,
        Name   VARCHAR(20),
        Gender   VARCHAR(20),
        Year_of_Birth   DATE,
        ELO_Rating   INT,
        Country   VARCHAR(20),
        PRIMARY KEY (FIDE_ID)
);

CREATE TABLE LiChess_Player (
        id   INT,
        created_at   DATE,
        seen_at   DATE,
        ELO_Rating   INT,
        Country   VARCHAR(20),
        PRIMARY KEY (id)
);

CREATE TABLE FIDE_Game (
        Date   DATE,
        Moves VARCHAR(100),
        Duration   INT,
        black_in   INT,
        white_in   INT,
        CONSTRAINT Fide_Constraint PRIMARY KEY (Date, black_in, white_in),
        FOREIGN KEY black_in REFERENCES FIDE_Player (FIDE_ID),
        FOREIGN KEY white_in REFERENCES FIDE_Player (FIDE_ID)
);

CREATE TABLE LiChess_Game (
        Date   DATE,
        Moves   VARCHAR(100),
        Duration    INT,
        black_in   INT,
        white_in   INT,
        CONSTRAINT LiChess_Constraint PRIMARY KEY (Date, black_in, white_in),
        FOREIGN KEY black_in REFERENCES LiChess_Player (id),
        FOREIGN KEY white_in REFERENCES LiChess_Player (id)
);
```

7. Explanation of how you will clean and pre-process the data. This tutorial

demonstrates how to do simple pre-processing in Python.

We will scrap Lichess games to get a list of usernames, and then send API requests to Lichess to get user/game info for that username. For Lichess and FIDE shared fields but different names, we will give them the same field name before storing them in our database.

8. List of technologies you will use. You must use some kind of SQL database. We recommend using MySQL or Oracle specifically because you will use MySQL in HW2, and we will provide guidance for setting up a MySQL database. Some groups in the past have had issues with MySQL, but Oracle is another option.

Database: MySQL
Frontend: Typescript, React
Backend: Express.js, Node.js
Parsing/Scraping: Python

9. Description of what each group member will be responsible for

Aydan - Backend, Parsing/Scraping || Features: Chess Game Display & Moves, Online Players
Constance - Parsing/Scraping, Frontend || Features: Rating Graphs & Analytics, Official Players
Henry - Full Stack || Features: Comparison Page
Serena - Frontend, Databases || Features: Search/Filter Players
**may separate by features instead, full stack for each feature