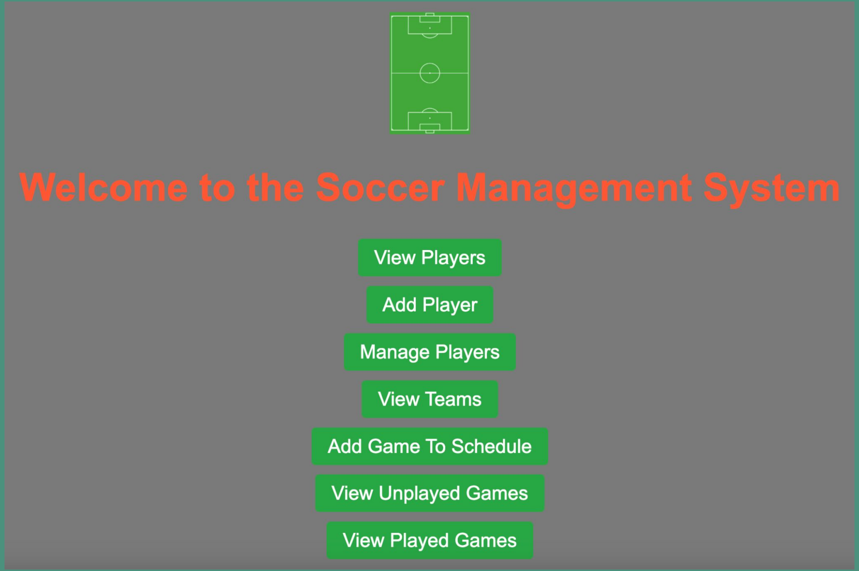
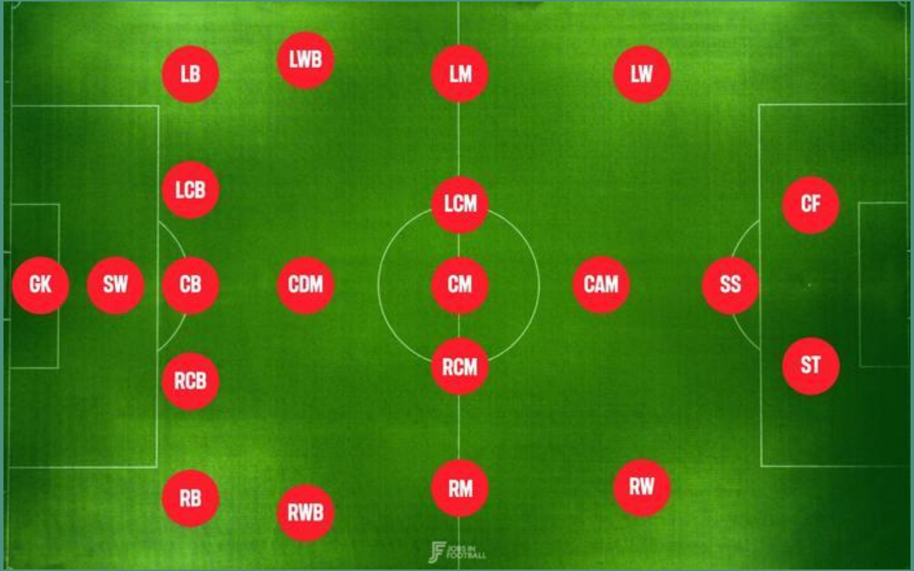
Soccer Manager


Welcome to the Soccer Management System



Zack Klopukh

Dec 12, 2024

Index & Content

1. Business Problem
2. Executive Summary
3. Key Processes
4. Technical Solution
5. Demo
6. Questions

Problem: Teams and managers need to be able to contact each other, and coaches need to be able to see the latest report on a team's players, tactics, and records.

Challenges:

- Keeping track of player stats, teams, and game results manually can lead to errors and inefficiencies.
- A need for an organized, easy-to-use system that helps users manage sports data, such as players, games, teams, and tactics.

Why It Matters: An intuitive solution can save time and provide accurate information for coaches, managers, and team members.

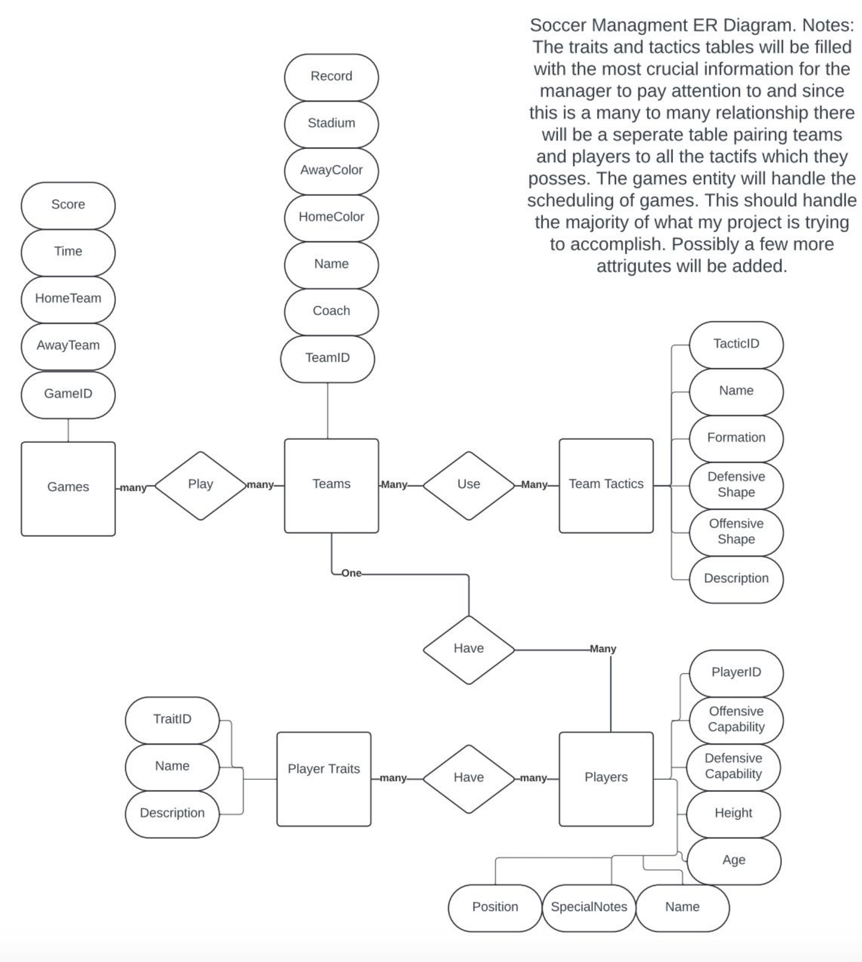
1. Business Problem

Soccer Manager helps sports teams and coaches manage their players, games, and team tactics.
The system provides a simple interface to view player stats, schedule games, and track team progress.
By automating data management, the system allows users to focus on improving team performance rather than dealing with manual processes.

Advantages: Coaches can view other teams reports including their tactics and dangerous players. Managers can add, and score games.

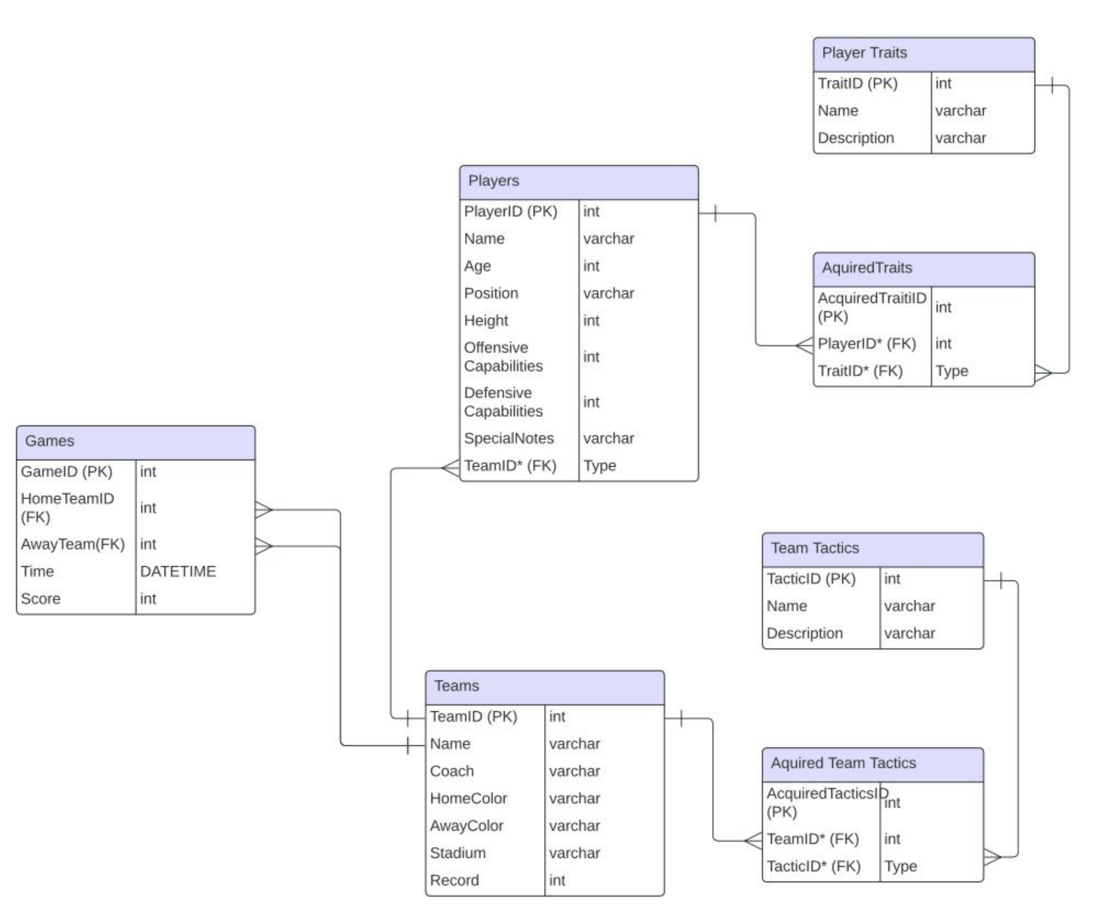
Executive Summary

Technical Solution



Soccer Managment ER Diagram. Notes: The traits and tactics tables will be filled with the most crucial information for the manager to pay attention to and since this is a many to many relationship there will be a seperate table pairing teams and players to all the tactifs which they posses. The games entity will handle the scheduling of games. This should handle the majority of what my project is trying to accomplish. Possibly a few more attrigutes will be added.

ER Diagram



Player Traits	
TraitID (PK)	int
Name	varchar
Description	varchar

Players	
PlayerID (PK)	int
Name	varchar
Age	int
Position	varchar
Height	int
Offensive Capabilities	int
Defensive Capabilities	int
SpecialNotes	varchar
TeamID* (FK)	Type

AquiredTraits	
AcquiredTraitiID (PK)	int
PlayerID* (FK)	int
TraitID* (FK)	Type

Games	
GameID (PK)	int
HomeTeamID (FK)	int
AwayTeam(FK)	int
Time	DATETIME
Score	int

Team Tactics	
TacticID (PK)	int
Name	varchar
Description	varchar

Teams	
TeamID (PK)	int
Name	varchar
Coach	varchar
HomeColor	varchar
AwayColor	varchar
Stadium	varchar
Record	int

Aquired Team Tactics	
AcquiredTacticsID (PK)	int
TeamID* (FK)	int
TacticID* (FK)	Type

Schema

Backend: handles the logic for managing teams, players, games, and tactics. This communicates with the database to fetch and update data.
Frontend: A simple user interface that allows users to interact with the system. It displays information like team stats, players, and tactics in an easy-to-navigate way.
Key Components:

- Python and Flask
- App in the form of html link

Organization Breakdown

Demo

Questions