Web app for a Basketball league manager

Team Status Report Document: T3_status_092220 (sample).xlsx
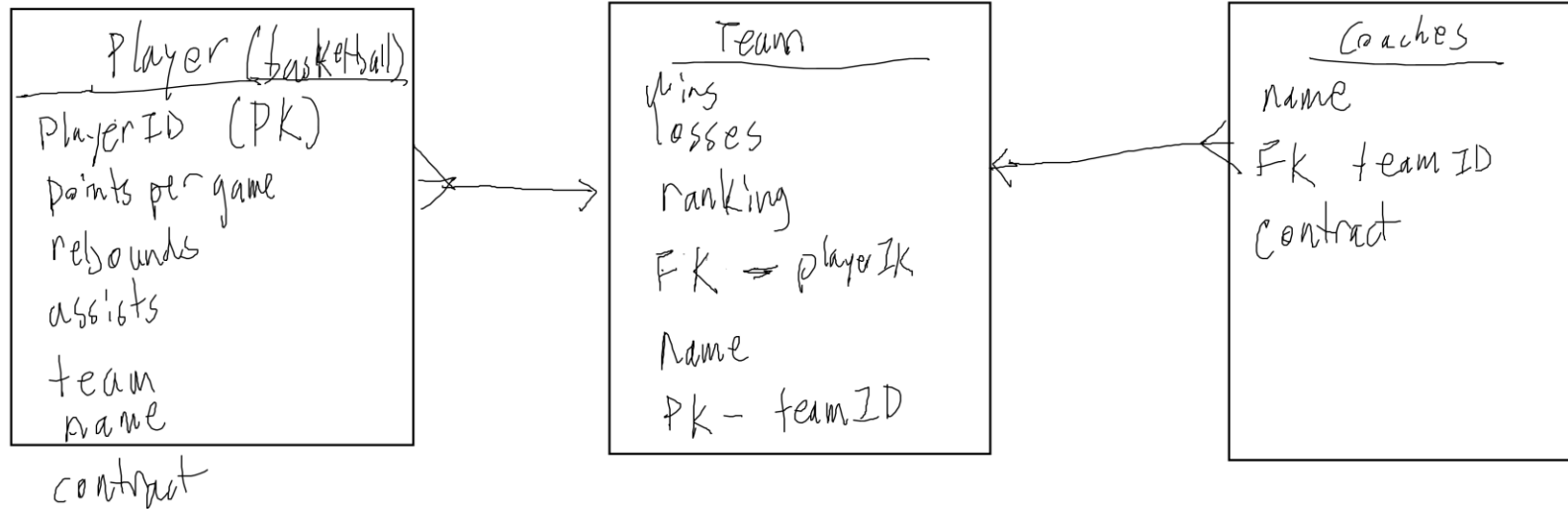
Project Timeline:

- Friday 1/19: homepage complete, stub pages implemented, database with data created (through SQL workbench)

```
sports-web-app/

|-- src/

|   |-- main/

|       |-- java/

|           |-- com.example.sportswebapp/

|               |-- SportsWebAppApplication.java

|               |-- controller/

|                   |-- CoachController.java

|                   |-- TeamController.java

|                   |-- PlayerController.java

|               |-- service/

|                   |-- CoachService.java

|                   |-- TeamService.java

|                   |-- PlayerService.java

|               |-- dao/

|                   |-- CoachDAO.java

|                   |-- TeamDAO.java

|                   |-- PlayerDAO.java

|               |-- model/

|                   |-- Coach.java

|                   |-- Team.java
```

```
|              |-- Player.java
|      |-- resources/
|          |-- application.properties
|-- pom.xml
```

Database

Player (basketball)
PlayerID (PK)
points per game
rebounds
assists
team
name

contract

Team
wins
losses
ranking
FK → player Ik
Name
PK - teamID

Coaches
name
FK team ID
contract

Views: Home

Basketball Stats

| Coach | Team | Player |

buttons to
take you
to a table of
coaches/teams/players

Coaches page

Coaches

Name	Salary	Team		
Bill	$1,000,000	Celtics	Edit	delete

scroll

Edit Coach Info

Name   [   ]

Salary [   ]

Info   [   ]

        [   ]

Edit form will be prefilled with data

Name

Alert:
input error

Salary

Info
Team   dropdown

Submit

leave all
data uncleared,
alert will notify
specific error

On valid submit,
send data to
/coach/id   POST

/coaches   GET

/coach/new   GET

Coaches

Name	Salary	Team		
Bill	$1,000,000	Celtics	Edit	delete

Add new

scroll

New Coach

Name [   ]

Salary [   ]

Team [▾]

[Submit]

Same alert error style as edit

Teams and players will follow the same structure

Team

View:
   Javascript to create HTML

Controller:
   RESP endpoints, call
   service layer

Service:
   Validation logic

DAO:
   Persist data

Coach

Follow same

MVC stack
structure as
team

Player

Follow same

MVC stack
structure as
team

Team

View:
   Javascript to create HTML

Controller:
   RESP endpoints, call
      service layer

Service:
   validation logic

DAO:
   Persist data

DAO interface

List<Team> getAllTeams();

Team getTeam(int id);

Team updateTeam(Team editTeam);

Team addTeam(Team newTeam);

// if FK constraints exist, throw Error

Team deleteTeam(int id);