

CASE STUDY: POKETEAM ACADEMY

By: Venus Teah

**A fun interactive experience meant for young
users**

USER STORIES AND BUSINESS USE CASE

- I envisioned this project to be an interactive website where fans of the Pokemon franchise would have an additional outlet they can use to explore their love of these little creature.
- The website would emulate a fictional school, where Users are enrolled as “Pokemon Trainers”.
- They would be able to create an account, choose a “Team” where they can connect with other Users, and have their own Pokemon.



TECHNOLOGIES USED

- JAVA
- SPRING INITIALIZER
- MYSQL/WORKBENCH
- SPRING BOOT
- SPRING DATA JPA
- MAVEN
- SPRING SECURITY
- THYMELEAF
- DRAW.IO
- CSS
- JS

Dependencies ADD DEPENDENCIES... 36 + B

Spring Web **WEB**

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

—

Lombok **DEVELOPER TOOLS**

Java annotation library which helps to reduce boilerplate code.

—

Spring Boot DevTools **DEVELOPER TOOLS**

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

—

Spring Security **SECURITY**

Highly customizable authentication and access-control framework for Spring applications.

—

MySQL Driver **SQL**

MySQL JDBC driver.

—

Spring Data JPA **SQL**

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

—

JDBC API **SQL**

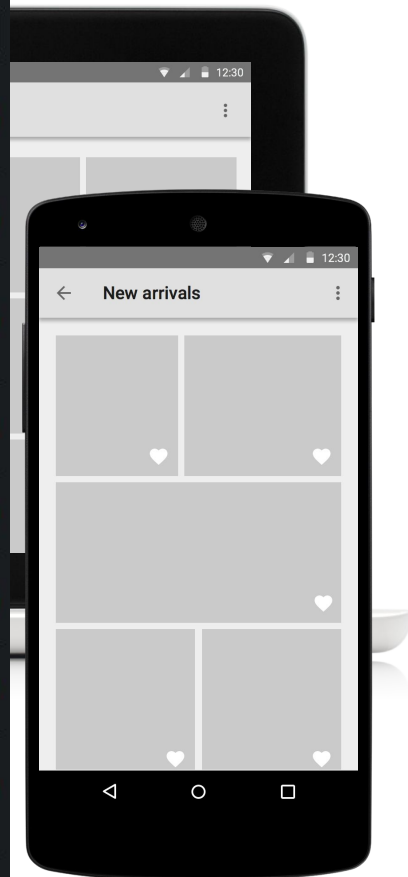
Database Connectivity API that defines how a client may connect and query a database.

—

Thymeleaf **TEMPLATE ENGINES**

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

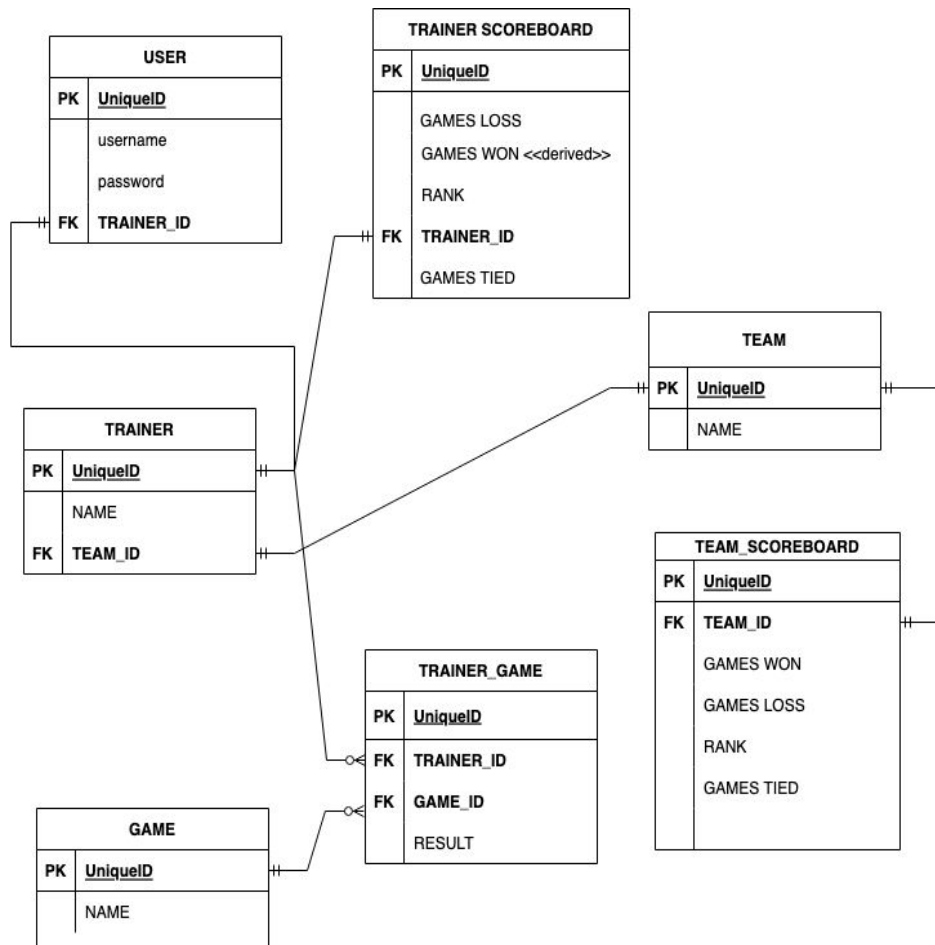
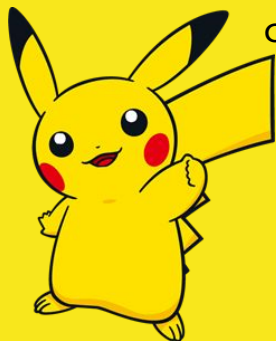
—



ORIGINAL ERD

Some Changes:

- Adding Role Table For Spring Security
- Leaving off game focused entities because of time constraints



Interesting Features

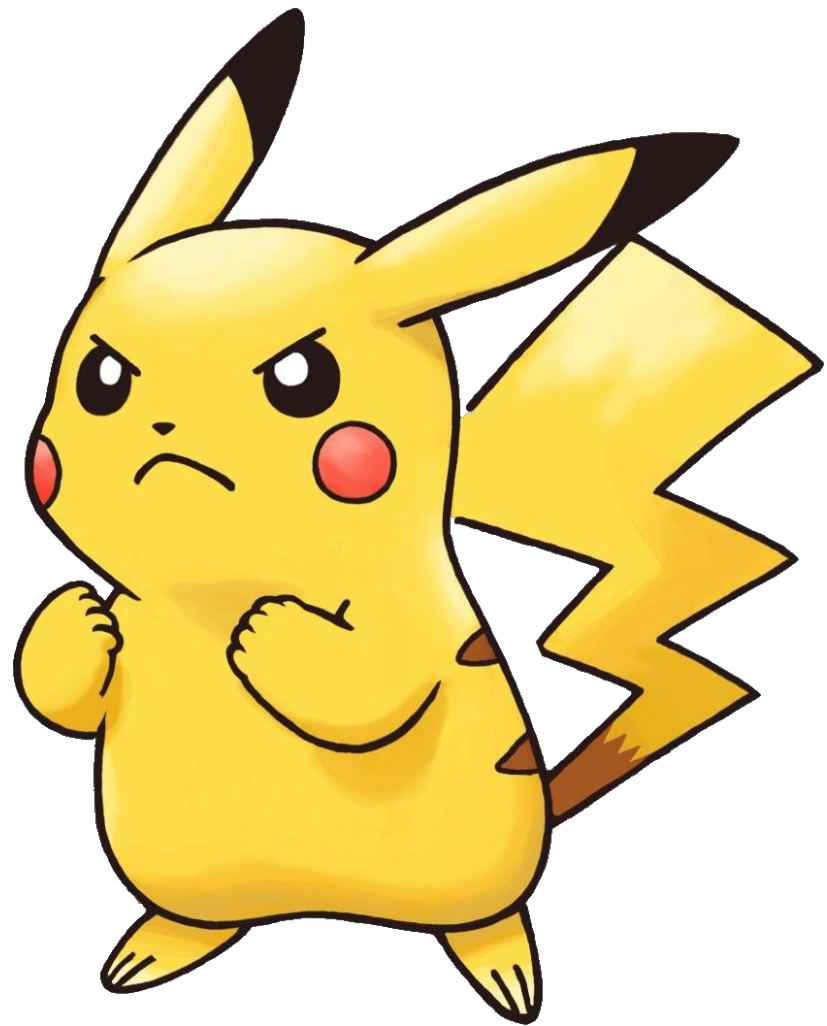
Poke API

- I challenged myself to attempt to extract data from an external api to use in my web app.



CHALLENGES

- **SPRING SECURITY**
 - Learned to take a step back to do additional research even when under pressure
- **FRONT END**
 - More problems than backend
 - In hindsight it would have been best to work on the front end first, in order to achieve better time management



Future Features



- Implementing the other entities in my original ERD –
 - Simple Pokemon Based Games
 - ScoreBoard
 - Rankings
- Add “Social” features
 - Chats

— — —

Resources

- Front end
 - <https://getbootstrap.com/docs/5.0/examples/>
 - <https://fontawesome.com/search?q=trash&o=r>
 - <https://getbootstrap.com/docs/4.0/examples/grid/>
 - <https://mdbootstrap.com/freebies/three-column/>
 -
- Spring Security
 - <https://www.youtube.com/watch?v=R76S0tfv36w&t=1970s>
 - https://www.youtube.com/watch?v=L9oWG6aj_U8&t=5s
 - <https://javatechonline.com/spring-security-userdetailsservice-using-spring-boot-3/>
 - <https://www.javacodegeeks.com/2018/02/securitycontext-securitycontextholder-spring-security.html>

-