Iskreno, malo me overwhelmao zadatak u ovom vremenskom roku. Kao što sam ti rekao, u svojim projektima i na poslu radio sam samo REST aplikacije u Springu, tako da sam prvi put radio Spring MVC i koristio Thymeleaf, ostale stvari glede baza podataka (poput repositorya i JPA) također nikad nisam koristio. S previše bugova sam se susreo prvi put iako sam se igrao s MVC i Thymeleaf u zadnja 24h, to me dodatno usporilo kao i činjenica da me u startu bila panika jer nisam znao koristiti knjižnice da bazu podataka koju si naveo u zadataku.

Ja sa produktom nisam zadovoljan nisam stigao posvetiti ni vremena stvarima s kojima sam dobar (da naprimjer pristup Mongo bazi bude uređeniji, konzistentniji i fault-proof).

**Baza:**

Baza je NoSQL. Zove se PokemonDatabase, ima dvije kolekcije: „Pokemon“ i „Trainer“

Izgledaju ovako:

Pokemon



Trainer



Odnos koji sam izmodelirao je da neki trener može imati 0..N pokemona. Koje pokemone ima piše u arrayu PokemonIDs.

URL-ovi kojima se može pristupiti kada se stranica pokrene su:

localhost:8080/pokemon – Prikaz svih pokemona.

localhost:8080/pokemon/{id} – Prikaz pokemona s tim ID-om.

localhost:8080/trainer – Prikaz svih trenera te koje pokemone posjeduje taj trener.

localhost:8080/trainer/{id} – Prikaz trenera s tim ID-om.

localhost:8080/addpokemon – Sučelje za dodati novog pokemona. Pokemoni s postojećim ID-om neće biti dodani.

localhost:8080/addtrainer – Sučelje za dodati novog trenera. Treneri s postojećim ID-om neće biti dodani. Nisam uspio napraviti da se preko ovog sučelja mogu definirati pokemoni koje trener posjeduje, u svrhu testiranja da to ipak radi pokemone sam dodavao trenerima ručno preko Mongo sučelja.