Documentation for CA3

The Idea:

The idea was to make a pokémon trainer, where you can create your own teams of pokemons. Users who login can see builds and teams other users have made. The user can follow other users if they like that users build etc. Furthermore it is possible to browse all pokemon and buy one (or more) for points that you earn.

Errors:

Rest-Services:

PostMan: We haven't really used Postman that much, because of the transient problems. We could see we have problems with our Rest by using Postman.

Circular Reference:

The entity classes pokemon, user and team have references to each other, so that we can fetch a list of teams, pokemon etc. from one entity and the other depending on the particular situation in front-end. Whenever we had to parse this data using gson.toJson(), this would cause a stackoverflow. We have to create our own json conversion using JsonObject in our rest services instead to avoid it.

Seed:

We had to refactor the seed extensively to make it compatible with our prototype. The biggest issue was the token-based authentication. We had to incorporate it into our own facades and entity classes.
Some problems still remain that needs to be considered before the seed can fully support our architecture that was added on top.

Rest-Assured: We haven't really made rest-assured because we have had, so many problems with circular references and couldn't get json objects out properly. We have tried with some easy test, but we made the decision to spend the time, to try to fix the problems we had.

FrameWorks:

JPA <-> MySQL caused some issues with a non-normalized database in used. Namely circular references, as described above.

To implement later:

User stories

1. As a trainer, I want to be able to create and delete teams, so that I can have teams for all of my pokemon and undo mistakes
2. As a trainer, I want to be able to browse the pokemon store, so I can choose which one to buy

3. As a trainer, I want to be able spend points, so that I can purchase pokemon and I can add them to teams
4. As a trainer, I want to be able to earn points, so that I can spend them on pokemon
5. As a trainer, I want to be able to see other trainers, so that I can follow them.
6. As a trainer, I want a follow-list, so I can add a trainer to my list.
7. As a trainer, I want to be able to add a follower to my follow list, so I can follow their progress.

Core functionality

We want to implement functionality so that a user can have detailed information about another user. Furthermore we want a user to be able to see detailed information about his own team of pokémon. This should be implemented by making an external api call to pokeapi.co, so that we get information such as name, types etc. for each pokemon owned and currently being displayed on screen from a team.

Study points:

Lucas

For his efforts during the project, he should be given full study points (how many they are). He primarily contributed with JPA, MySql and AngularJS Front-end.

Thomas

For his efforts during the project, he should be given full study points (how many they are). He primarily contributed with RESTFUL services, facades and attempt at RESTassured tests. Furthermore he worked on documentation.

Patrick

For his efforts during the project, he should be given full study points (how many they are). He primarily contributed with facades, JUnit tests and AngularJS front-end. Did not write as much new code as the others, but spend his time refactoring and debugging/testing existing codes to make sure it worked and if not: *make it work.* Furthermore he worked on documentation.

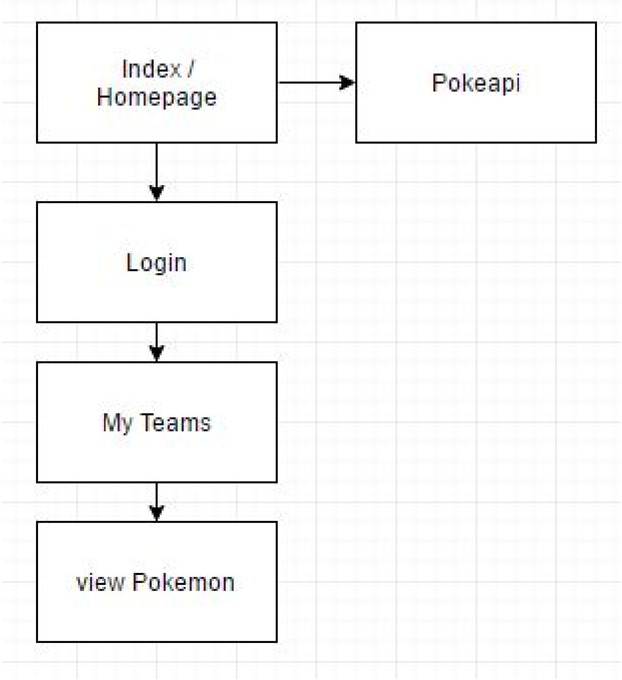
Junit-tests:

Methods we have written tests for:

• `testGetAllUsers()`	PASSSED
• `testGetUserByName()`	PASSSED
• `testGetfollowList()`	PASSSED
• `testAddUserToFollowList()`	PASSSED
• `testCreateTeam()`	PASSSED
• `testGetTeams()`	PASSSED
• `testGetTeamById()`	PASSSED
• `testCreatePokemon()`	PASSSED
• `testGetAllPokemon()`	PASSSED
• `testGetPokemonById()`	PASSSED
• `testGetPokemonByTeam()`	PASSSED
• `testAddPoints()`	FAILED

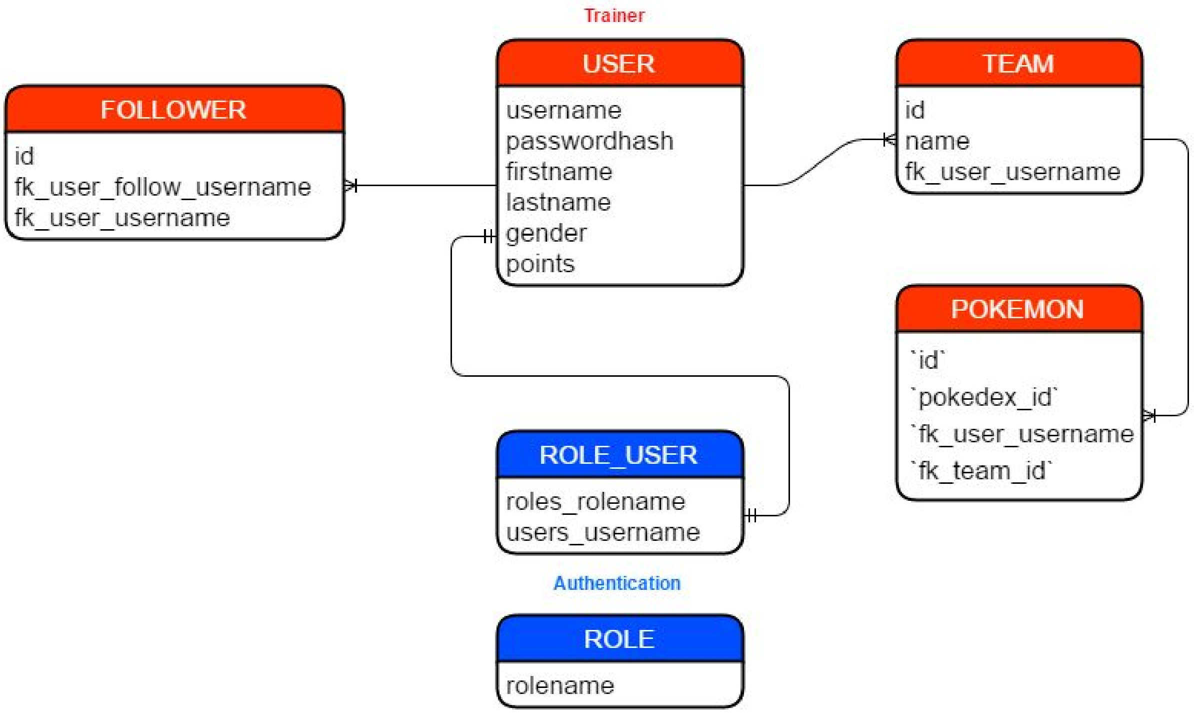
Diagrams:2

Navigational diagram



ER-diagram

MyPokeTrainer ER-diagram (database relations)


Trainer
USER
username
passwordhash
firstname
lastname
gender
points
FOLLOWER
id
fk_user_follow_username
fk_user_username
TEAM
id
name
fk_user_username
POKEMON
`id`
`pokedex_id`
`fk_user_username
`fk_team_id`
ROLE_USER
roles_rolename
users_username
Authentication
ROLE
rolename