

Projekt Zespołowy Backend
przy współpracy z firmą Capgemini

"ActiveEmployee"

Jakub Dereń 235791
Michał Bańka 235051

Prowadzący projekt: dr inż. Tomasz Kubik

13 czerwca 2019

Spis treści

1	Architektura	3
1.1	Warstwa repozytoryjna	3
1.2	Warstwa prezentacji	3
2	Instrukcja obsługi	3
2.1	Wymagania	3
2.2	Uruchomienie	3
3	Baza danych	4
3.1	Schemat	4
3.2	Opis encji	5
4	Literatura	6

1 Architektura

1.1 Warstwa repozytoryjna

Odpowiedzialna jest za przechowywanie i operacje na danych. W naszym projekcie zrealizowana jest dzięki frameworkowi Hibernate, który zapewnia mapowanie obiektowo relacyjne.

1.2 Warstwa prezentacji

W naszej aplikacji odpowiedzialna jest za przygotowywanie danych do prezentacji oraz wystawienie ich dzięki REST API. Dzięki zastosowaniu Spring Boota posiadamy zbiór wszystkich wymaganych bibliotek oraz konfiguracji. Ponadto dostarcza on również wbudowany serwer aplikacyjny Tomcat. Każdy model w naszej aplikacji ma przypisany do siebie kontroler.

2 Instrukcja obsługi

2.1 Wymagania

Do uruchomienia aplikacji wymagane są następujące zasoby:

- IntelliJ IDEA 2018.3 Ultimate lub nowszy
- MySQL Community Server
- Java JDK 1.8
- Maven
- Windows, Mac lub Linux
- MySQL Workbench 8.0

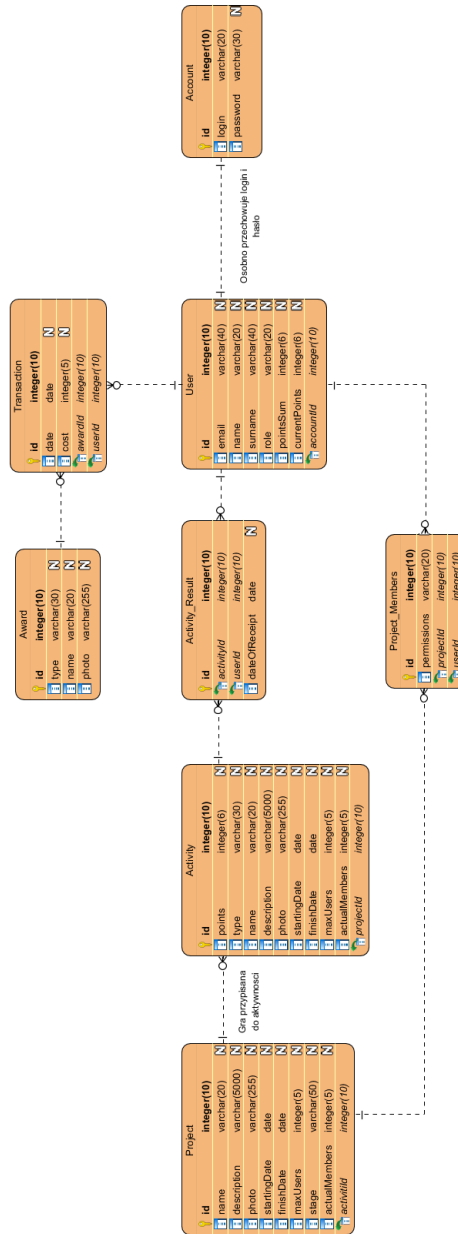
2.2 Uruchomienie

Po zainstalowaniu wszystkich wymaganych zasobów wykonujemy następujące kroki:

- Pierwszym krokiem jest stworzenie bazy o nazwie `gamification_db`.
- Za pomocą IntelliJ otwieramy projekt o nazwie `gamification`.
- W pliku `application.properties` należy ustawić dane do autentykacji z bazą danych oraz adres i port na którym aplikacja będzie uruchamiana.
- Po pobraniu przez Mavena wszystkich potrzebnych zależności uruchamiamy projekt poprzez klasę `GamificationApplication`.

3 Baza danych

3.1 Schemat



Rysunek 1: Schemat bazy danych

3.2 Opis encji

- User
Encja przechowująca dane konkretnego użytkownika (profil). Znajdują się w niej między innymi dane osobowe oraz jego ranga. Ponadto pole `pointsSum` określa łączną ilość punktów zdobytych przez użytkownika w całej historii istnienia konta, zaś pole `currentPoints` określa ilość punktów, które zostały użytkownikowi do wymiany za nagrody. Określa się w niej także status użytkownika (Administrator czy Pracownik).
- Award
Encja definiująca nagrody które może otrzymać użytkownik za określoną ilość punktów. Dowiemy się z niej jaki jest to typ nagrody oraz jak się nazywa.
- Transaction
Encja przechowująca transakcje zakupów nagród. Przechowuje ona datę zakupu oraz koszt nabytej nagrody.
- Account
Encja ściśle związana z encją „Users”, ponieważ przechowywane są w niej hasło i login do konta użytkowników. Stworzona została aby odzielić od siebie pewne dane.
- Project
Encja definiująca projekty na które użytkownik może się zapisać. Dowiemy się z niej ilu użytkowników zapisało się na dany projekt i ilu może zapisać się na dany projekt, oraz na jakim etapie on się znajduje (zapisy, w trakcie oraz zakończony). Posiada opis na czym będzie polegać oraz przewidywana datę rozpoczęcia i zakończenia. W każdym projekcie będzie można wykonywać jeszcze różne aktywności.
- Activity
Rodzaje aktywności jakie możemy wykonywać podczas projektów np. wygłoszenie mowy, przedstawienie prezentacji, pomoc przy organizacji itp. Każda aktywność jest premiowana dodatkową ilością punktów. Posiada prawie takie same atrybuty jak encja „Project”.
- Activity_Result
Encja przejściowa między „User” a „Activity”. Posiada atrybut który pokazuje datę otrzymania punktów po zakończonej aktywności.
- Project_Members
Encja przejściowa między „User” a „Project”. Określone są w niej uprawnienia danego użytkownika podczas projektów (organizator, użytkownik).

4 Literatura

Literatura

- [1] Spring documentation
<https://spring.io/docs>.
- [2] Maven documentation
<https://maven.apache.org/guides/index.html/>.
- [3] Java documentation
<https://docs.oracle.com/javase/8/docs/>.
- [4] Hibernate documentation
<http://hibernate.org/>
- [5] stackoverflow
<https://stackoverflow.com>

.