

Prowadzący
Doktor inżynier Tomasz Kubik

Projekt zespołowy ActiveEmployee

Skład grupy:

Jakub Dereń, 235791
Grzegorz Tomasik, 235094
Krzysztof Kapala, 235060
Kacper Jaros, 238905
Karolina Madej, 238893
Michał Bańka, 235051

Spis treści

1. OPIS PROJEKTU.....	3
2. SŁOWNIK POJEĆ	3
3. PERSPEKTYWA FUNKCJONALNA	3
4. PERSPEKTYWA PROCESÓW	4
5. OPIS TECHNICZNY	4
5.1. BACKEND	4
5.2. FRONTEND	5
6. PERSPEKTYWA WDROŻENIOWA	5
7. BAZA DANYCH	5
7.1. OPIS ENCJI	5
8. ZAŁOŻENIA PROJEKTU	6
8.1. WYMAGANIA FUNKCJONALNE	6
8.2. WYMAGANIA NIEFUNKCJONALNE.....	7
9. ETAPY POWSTANIA PROJEKTU.....	7
9.1. ETAP PIERWSZY	8
9.2. ETAP DRUGI.....	8
9.3. ETAP TRZECI.....	8
9.4. ETAP CZWARTY	8
9.5. ETAP PIĄTY	8

1. Opis projektu

ActiveEmployee jest aplikacją webową służącą do motywacji pracowników w celu polepszenia i poprawy wyników pracy i zaangażowania bazującej na zasadzie grywalizacji. Technika

ta pozwala zaangażować ludzi do zajęć, które są zgodne z oczekiwaniami autora projektu, nawet jeśli są one uważane za nudne lub rutynowe.

2. Słownik pojęć

O ile w niniejszym dokumencie wyraźnie inaczej nie wskazano, następujące wyrażenia będą miały następujące znaczenie:

Activity – aktywność
Czynności składające się na realizację projektu

Project – projekt
Projekt realizowany poprzez aktywności.

Name – nazwisko użytkownika

Surname – imię użytkownika

Points – punkty
Punkty zdobyte przez użytkownika po zrealizowanej aktywności.

Active – aktywne
Aktywne czynności użytkownika.

Historical – zarchiwizowane
Zakończone czynności.

Award's name – nazwa nagrody
Nagroda, którą można zdobyć po osiągnięciu odpowiedniej liczby punktów.

Points cost – punkty
Ilość punktów potrzebna do zdobycia nagrody.

3. Perspektywa funkcjonalna

Użytkownicy

Omawiany system będzie wykorzystywany przez pracowników firm. Użytkownik będzie miał przydzieloną rolę w systemie (administrator, użytkownik, organizator projektu), w zależności od pełnionej funkcji.

Informacje

System będzie przetwarzał różne rodzaje informacji. Będą to między innymi raporty o wykonanych aktywnościach czy uczestnictwie w różnych projektach. Program będzie zawierać listę wszystkich pracowników danej firmy. Każdy użytkownik ma przypisane informacje do odczytania takie jak ilość punktów, historia wykonanych aktywności czy obecne aktywności lub projekty.

Funkcje

Każdy użytkownik ma dostęp do różnych operacji w systemie w zależności od uprawnień. Zwykły użytkownik będzie mógł się zapisać do organizowanych projektów a w zapisanym projekcie może uczestniczyć w konkretnych aktywnościach. Po zakończonej aktywności użytkownik otrzymuje punkty za jej wykonanie. Otrzymane punkty może wymienić na nagrody. Organizator projektu może dodatkowo nie zaakceptować wykonanej aktywności przez użytkowników uczestniczących w jego projekcie. Administrator posiada dostęp do wszystkich funkcji w systemie.

Systemy zewnętrzne

System podzielony jest na dwie sekcje takie jakie backend oraz frontend. Backend realizowany jest za pomocą języka programowania Java w wersji 1.8. W celu łatwiejszego zarządzania kodem użyty został również framework SpringBoot, za automatyzację budowy oprogramowania odpowiada Apache Maven. Dzięki oprogramowaniu Hibernate otrzymujemy łatwiejszy dostęp do danych i zapewnia on nam translację danych między bazą danych a Javą. Frontend aplikacji realizowany jest w języku JavaScript z użytym frameworkiem React który tworzy nam szablony wyglądu naszej aplikacji. Za pomocą Bootstrapa tworzenie interfejsu graficznego jest ułatwione dzięki rozwiniętej bibliotece CSS.

4. Perspektywa procesów

Przy pomocy systemu „ActiveEmployee” realizujemy procesy takie jak :

- Rejestracja użytkowników
- Logowanie do systemu
- Stworzenie projektu
- Dołączenie do projektu
- Opuszczenie projektu
- Dołączenie do aktywności
- Opuszczenie aktywności
- Ranking aktywnych użytkowników
- Zdobywanie punktów
- Odbieranie nagród

5. Opis techniczny

Aplikację można podzielić na 2 główne moduły backend i frontend.

5.1. Backend

- Java8
- Spring Boot
- Spring Security
- Hibernate
- Maven
- JPA
- MySQL
- Apache Tomcat

5.2. Frontend

- React
- Redux
- HTML5
- CSS3
- JavaScript
- Bootstrap

6. Perspektywa wdrożeniowa

Baza danych

Aby skorzystać z projektu naszego schematu bazy danych należy stworzyć nową bazę danych obsługującą MS SQL (np. w Microsoft SQL Server) podając odpowiednią nazwę bazy i użytkownika. Posiadając aktywne połączenie z serwerem bazy danych należy uruchomić skrypt w języku Java. Dzięki temu Hibernate automatycznie wygeneruje bazę danych i połączy się z nią.

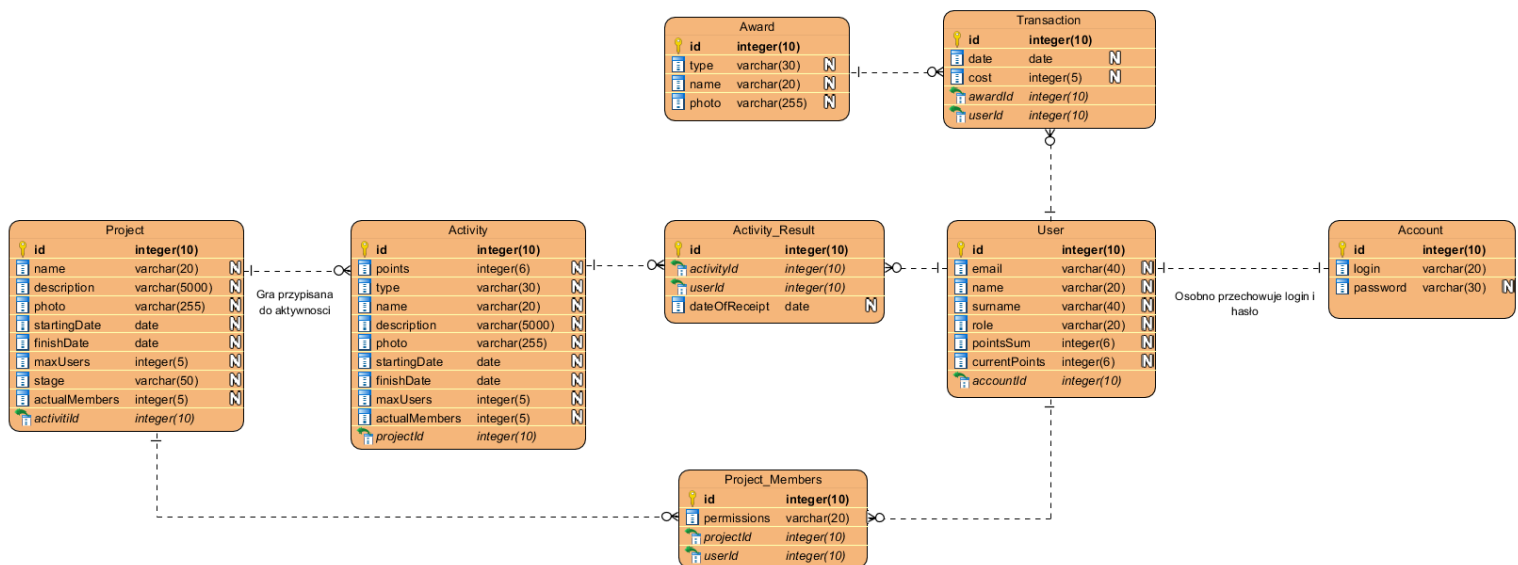
Backend

Aby backend działał poprawnie w pierwszej kolejności należy pobrać aktualny kod z repozytorium github. Następnie należy prawidłowo skonfigurować program *IntelliJ IDEA* po przez dodanie do niego frameworka SpringBoot oraz Hibernate. Po wykonaniu wszystkich tych czynności należy skompilować nasz kod a następnie uruchomić skrypt.

Frontend

Aby frontend działał poprawnie w pierwszej kolejności należy pobrać aktualny kod z repozytorium github. W następnej kolejności będąc w folderze z projektem musimy skorzystać z komendy `npm start`, która utworzy nam zoptymalizowaną wersję naszego frontendu. Należy pamiętać, aby w tle uruchomiony był cały czas nasz serwer oraz backend.

7. Baza danych



7.1. 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 kiedy ją otrzymaliśmy oraz jaki jest jej koszt.

Account:

Encja ściśle związana z encją „Users”, ponieważ przechowywane są w niej hasło i login do konta użytkowników. Stworzona została, aby oddzielić 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 data 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).

8. Założenia projektu

8.1. Wymagania funkcjonalne

Dostęp chroniony

Dostęp chroniony wymaga rejestracji użytkownika.

- Umożliwi definiowanie projektów (zwanych grami), do których przyporządkowane będą aktywności
- Każda aktywność ma przypisaną określoną wartość punktową, za wypełnienie której dany użytkownik będzie zdobywał punkty
- Każdy użytkownik może zarejestrować się w systemie i wziąć udział w wielu grach
- Zdobyć punkty w danej aktywności można zdobyć na wiele sposobów, np.
 - o Poprzez wpisanie unikalnego kodu wygenerowanego przez właściciela gry (administratora)
 - o Zeskanowaniu odpowiedniego kodu QR
 - o Odwiedzenie określonej strony internetowej,
 - o Ręczne przypisanie punktów przez administratora, etc.

- Na podstawie aktywności użytkownicy mogą zdobywać trofea i uzyskiwać odpowiednie rangi
- Punkty wyświetlane są w ogólnodostępnym rankingu, do którego dostęp mają wszyscy użytkownicy
- Punkty i statystyki prowadzone są na poziomie globalnym (zbiór wszystkich gier) lub lokalnym (pojedyncza gra)
- W rozszerzonej wersji grywalizacji punkty wymieniane mogą zostać na określone nagrody (bony, gadżety, etc.)
- Gry można resetować i archiwizować ich wyniki (celem robienia podsumowań rocznych, kwartalnych, etc.)
- Grywalizacja powinna być elastyczna, a kolejne możliwości rozwoju projektu powinny zostać zaproponowane.

Dostęp administratora

W systemie istnieje użytkownik o identyfikatorze „administrator”. W kontekście tego użytkownika można wykonywać wszystkie czynności administracyjne. System udostępnia dwie podstawowe funkcje administracyjne:

- Administrator ma możliwość dodawania do systemu nagród, które mogą kupić użytkownicy.

Autoryzacja i uwierzytelnianie

- Użytkownik ma możliwość rejestrowania i logowania do systemu.
- Login i hasło użytkownika przechowywane jest w tabeli Account i jest powiązane z konkretnym użytkownikiem z tabeli User.
- W bazie danych hasło jest haszowane, po logowaniu hasło jest porównywane z haszem.

8.2. Wymagania niefunkcjonalne

Wydajność

- Maksymalny czas odpowiedzi aplikacji na zapytanie użytkownika nie może być dłuższy niż 3 sekundy.
- Aplikacja będzie umożliwiała korzystanie z niej przez 1000 osób równocześnie.

Skalowalność

- Pożądane jest, aby podsystem umożliwiał zwiększanie objętości obsługiwanych danych.
- Pożądane jest, aby zapewnić skalowalność i możliwość skalowania na maszynach wieloprocesorowych.

Niezawodność

- Aplikacja będzie dostępna 24/7 średnio 99,9% czasu.
- Aplikacja będzie dostępna dla wszystkich użytkowników
- Niezbędne wyłączenia serwisów całego Systemu mogą nastąpić jedynie w godzinach nocnych.
- Wszyscy użytkownicy mogą zgłaszać błędy na dodatkowy adres email.

Bezpieczeństwo

- Bezpieczeństwo komunikacji. Komunikacja z aplikacją tylko uprawnionych użytkowników.
- Bezpieczeństwo fizyczne. Realizowane na poziomie uprawnień dostępów do plików.

9. Etapy powstania projektu

9.1. Etap pierwszy

Podczas pierwszego spotkania w firmie Capgemini zostało nam przedstawione zagadnienie grywalizacji, na którym będzie opierać się działanie aplikacji. Na kolejnym spotkaniu zostały nam przedstawione wymagania odnośnie projektu i interfejsu. Na tej podstawie stworzyliśmy makiety aplikacji.

Naszym zadaniem było opisanie wymagań niefunkcjonalnych i funkcjonalnych oraz projekt bazy danych. Po konsultacjach z opiekunem projektu i ewentualnych zmianach przeszliśmy do etapu drugiego.

9.2. Etap drugi

Kolejny etap rozpoczęliśmy od spotkania z firmie, w której omówiliśmy wymagania funkcjonalne i ich implementację. Początkowo zrezygnowaliśmy z implementacji logowania, rejestracji i skupiliśmy się na pierwszej funkcjonalności.

Dalsze zmiany polegały na modyfikacji bazy danych i relacji między tabelami. Zadaniem do wykonania była zmiana opisu użytych technologii oraz rozwinięcie wymagań niefunkcjonalnych.

9.3. Etap trzeci

Kolejne spotkanie polegało na weryfikacji dotychczasowej pracy i jej zatwierdzeniu. Następnym punktem spotkania było przedstawienie kolejnych zadań do zrealizowania m. in. profilu użytkownika i rankingu oraz powrót tematu logowania i rejestracji.

9.4. Etap czwarty

Etap czwarty polegał na ostatecznych poprawkach i weryfikacjach błędów aplikacji.

9.5. Etap piąty

Zakończenie projektu.