

Programowanie aplikacji mobilnych

*Quiz Game*

**Opiekun projektu:** mgr inż. Marcin Chyła

**Autor Projektu:** Jakub Kuśnierz

**Data wykonania:** Semestr zimowy rok akademicki 2018/2019

1. **Cel projektu**

W ramach przedmiotu „Programowanie aplikacji mobilnych” wykonana została gra quizowa na urządzenia z systemem android oraz serwis REST wykonany w technologii Java. Baza danych gry umieszczona została w serwisie Amazon AWS, dzięki czemu każdy gracz posiada taką samą pulę pytań.

1. **Opis Aplikacji**

Aplikacja pozwala na rozwiązanie quizu z wybranej przez użytkownika kategorii pytań. Rozgrywka toczy się do momentu udzielenia błędnej odpowiedzi lub wyczerpania puli pytań z danej kategorii.   
Za poprawną odpowiedź gracz otrzymuje adekwatną dla danego poziomu ilość punktów.

Gracze posiadają także możliwość dodawania własnych pytań do bazy, w celu powiększenia ich liczby zwiększając przy tym poziom trudności gry.

W grze dostępny jest ranking, do którego zostają zapisane wszystkie wyniki osiągnięte przez graczy. Dzięki niemu możemy porównywać swoje wyniki z wynikami osiągniętymi przez innych graczy oraz zobaczyć najlepsze wyniki osiągnięte w grze.

1. **Wykorzystane technologie**

- AWS Amazon – RDS MySQL

- Android API 23

- Java 10

- Spring Boot

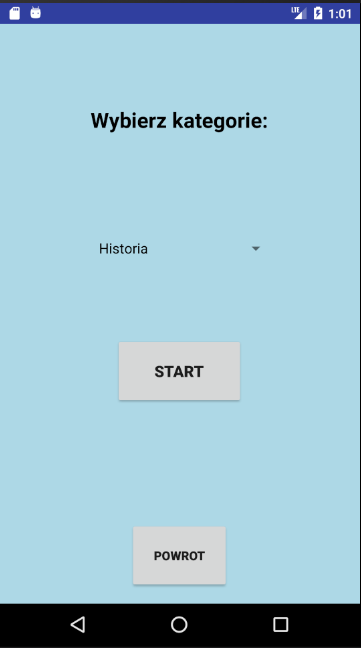
- Hibernate

1. **Ekrany aplikacji wraz z opisem funkcjonalności**
2. **Menu główne**



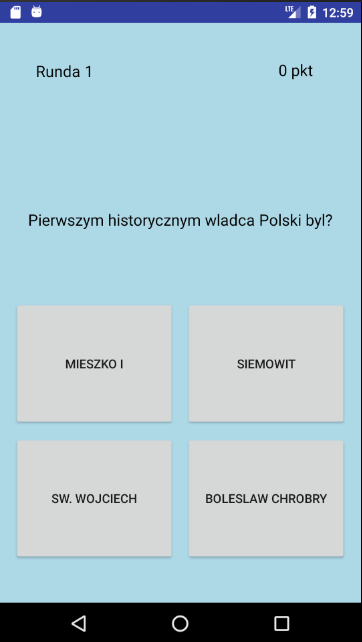
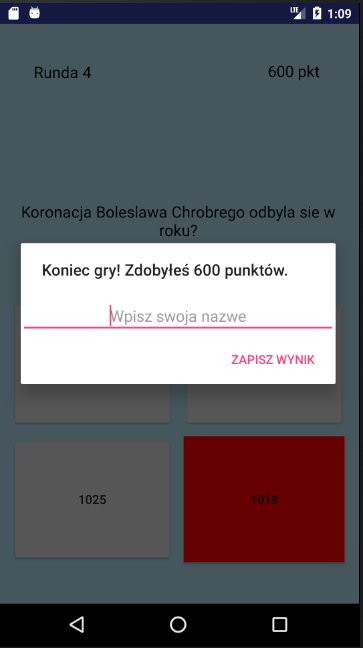
W menu głównym aplikacji znajdują się przyciski pozwalające rozpocząć nową grę, dodać pytanie do bazy, wyświetlić aktualny ranking z najlepszymi wynikami oraz zamknąć naszą aplikację. Jest to ekran startowy naszej aplikacji, który otwiera się po uruchomieniu aplikacji.

1. **Wybór kategorii quizu**



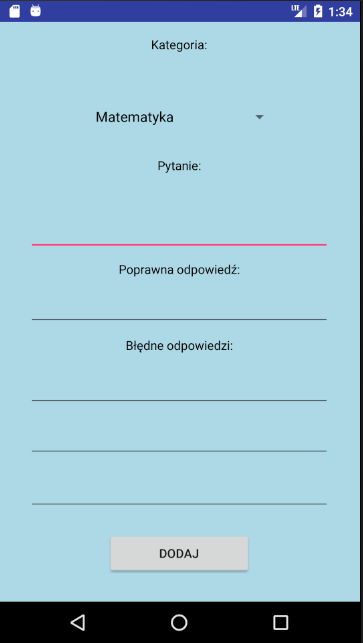
Ekran ten pojawi się na ekranie telefonu po kliknięciu   
w przycisk „Nowa gra” w menu głównym gry. Dokonujemy na nim wyboru kategorii z listy wszystkich które posiadają minimalną ilość pytań aby quiz był możliwy do wykonania. Po wyborze kategorii możemy rozpocząć nową grę klikając w przycisk „START”. Ekran umożliwia również możliwość powrotu do menu głównego przez kliknięcie przycisku „Powrót”.

1. **Rozgrywka**



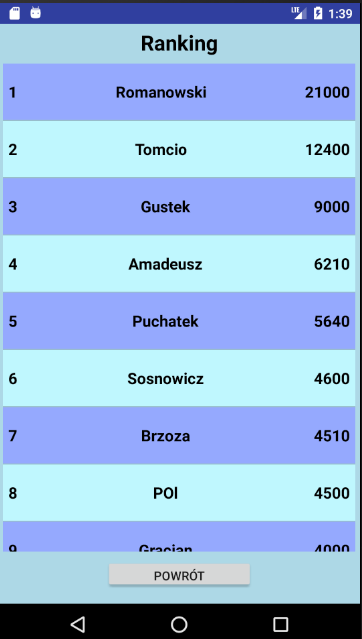
Ekran po lewej stronie ukazuję przykładowe pytanie podczas rozgrywki. W górnej części ukazana jest informacja o aktualnej rundzie oraz sumie zgromadzonych do tej pory punktów. Poniżej znajduję się wylosowane pytanie a następnie 4 przyciski z wariantami odpowiedzi   
z których zawsze tylko jedna jest poprawna. Po wybraniu poprawnej odpowiedzi gracz przechodzi do następnej rundy. Natomiast po udzieleniu błędnej odpowiedzi proszony jest o podanie swojej nazwy (ekran po prawej stronie) w celu zapisania wyniku do rankingu. Jeżeli odpowiedź jest prawidłowa przycisk zmienia swój kolor na zielony, jeżeli błędna – na czerwony.

1. **Dodawanie pytania**



Ekran ten ukazany zostaje po wybraniu opcji „Dodaj pytanie” w menu głównym aplikacji. Umożliwia on dodanie nowego pytania do bazy. Aby dokonać musimy wybrać kategorię do której przynależeć będzie pytanie, podać treść pytania, podać poprawną odpowiedź oraz podać trzy błędne odpowiedzi. Po wypełnieniu powyższych pól formularza całość należy potwierdzić klikając w przycisk „DODAJ”. Wszystkie pola formularza są walidowane przez aplikację co uniemożliwia wpisania niepoprawnych danych oraz wysłanie niepełnego formularza.

1. **Ranking**



Ekran ten ukazany zostaje po wybraniu opcji „Ranking” w menu głównym aplikacji. Ukazuję on 20 najlepszych wyników osiągniętych przez graczy podczas rozwiązywania quizów. W lewej kolumnie ukazane jest miejsce gracza w rankingu, w środkowej podana jest jego nazwa, natomiast w prawej kolumnie wynik który udało mu się osiągnąć.

1. **Baza danych**

Baza danych została stworzona przy wykorzystaniu silnika MySQL. Baza jest przetrzymywana w chmurze AWS Amazon dzięki czemu w grę można grać z kilku urządzeń jednocześnie posiadając dostęp do aktualnej bazy pytań oraz rankingu. Można także dodawać pytania które są od razu dostępne dla innych graczy.

Baza danych składa się z 4 tabel które są wykorzystywane przez aplikacje do przechowywania danych. W bazie występują wyłącznie relację jeden do wielu. Jednak z tabel jest tabelą słownikową służącą do przechowywania nazwy kategorii.

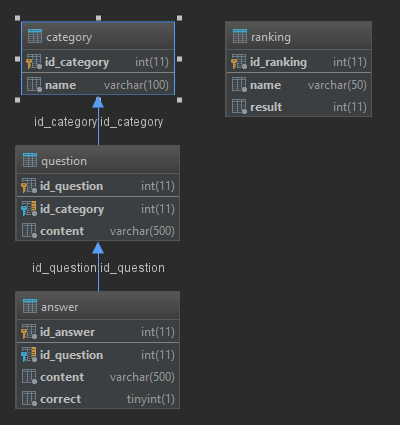
**Tabele:**

- **category** – tabela słownikowa do przechowywania nazw kategorii

- **question** – tabela do przechowywania pytań oraz kategorii do której należą

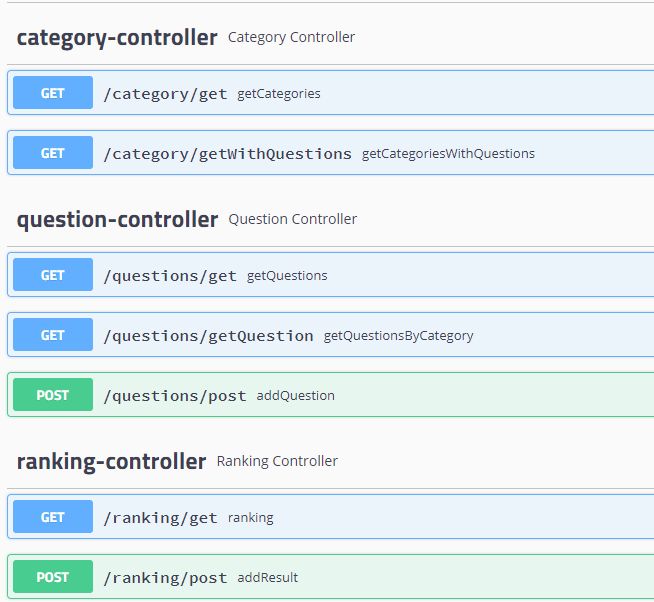
- **answer** – przechowywane są w niej odpowiedzi do pytań

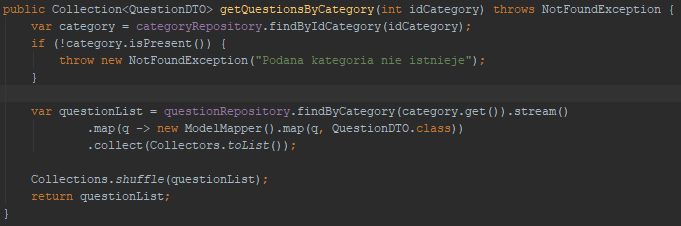
- **ranking** – zawiera wyniki osiągane przez graczy wraz z ich nazwami

**Schemat encji:**

1. **Dokumentacja serwisu restowego**

Serwis restowy zawierający logikę aplikacji oraz odpowiadający za obsługę bazy danych wykonany został w technologii Java oraz jej najpopularniejszych wymienionych wcześniej bibliotekach. Dokumentacja serwisu została wykonana przy użyciu biblioteki „Swagger” która pozwala w łatwy sposób wizualizować oraz korzystać z dostępnego w aplikacji API przy okazji tworząc jego dokumentację.

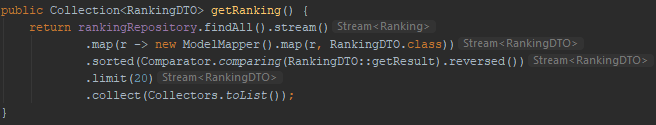


1. **Przykładowe fragmenty kodu aplikacji**
2. **Wyszukiwanie wszystkich pytań z danej kategorii**

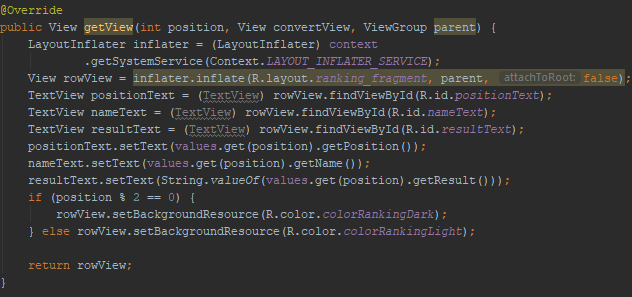
Powyższa metoda jest częścią serwisu restowego. Zwraca nam ona wszystkie pytania wraz z odpowiedziami z kategorii o id podanym jako parametr wyjściowy. Pytania te na końcu są mieszane metodą *shuffle* aby zostały wyświetlane w quizie w losowej kolejności.

1. **Zapytanie *Get* do serwisu restowego**

Powyższa metoda wykorzystywana jest to wykonywania zapytań typu *Get* do serwisu restowego naszej aplikacji. Wymaga ona otrzymania w parametrze dokładnego adresu URL do którego ma wysłać nasze zapytanie. W odpowiedzi zwraca zawartość w formacie JSON jeżeli kod odpowiedzi wynosił 200 lub sam kod odpowiedzi http jeżeli był on inny niż oczekiwany kod 200.

1. **Lista najlepszych wyników**

Metoda ta również jest częścią serwisu restowego. Zwraca nam ona najlepsze 20 wyników graczy w kolejności od największego do najmniejszego. Wyniki te są wykorzystane do wyświetlania rankingu w grze.

1. **Adapter layoutu rankingu**

Powyższa metoda jest częścią klasy dzięki której tworzymy adapter layoutu. Adapter ten wykorzystany jest w naszej aplikacji do stworzenia rankingu najlepszych wyników w oparciu o fragmenty. W powyższej metodzie przypisane są odpowiednie wartości dla poszczególnych elementów widoku oraz kolor tła w zależności od tego czy element znajduję się na pozycji parzystej czy nieparzystej.