**UNIWERSYTET EKONOMICZNY**

**W KATOWICACH**

Kierunek: Informatyka i Ekonometria, stacjonarne I stopnia

Grupa: E-Biznes

**PRZEDMIOT:**

**Programowanie usług w środowisku rozproszonym**

PROJEKT: SCORE & PLAY

WYKONAŁ: Anna Gajos, Andrzej Szuwald, Maksymilian Wrześniewski

DATA: 15.01.2024

Spis treści

[1.Faza Inicjacji 4](#_Toc156492096)

[1.1 Zespół projektowy 4](#_Toc156492097)

[1.2 Use-case 4](#_Toc156492098)

[2.Faza Opracowania 6](#_Toc156492099)

[2.1 Diagram klas 6](#_Toc156492100)

[2.2 Wybranie języka programowania 6](#_Toc156492101)

[3.Faza Konstrukcji 7](#_Toc156492102)

[3.1 Stworzenie pierwszej aplikacji i uruchomienie Android Studio 7](#_Toc156492103)

[3.2 Nauka android studio 9](#_Toc156492104)

[3.3 Instalacja rozszerzeń do aplikacji 10](#_Toc156492105)

[4.Faza Przekazania 12](#_Toc156492106)

[4.1 Ogólna struktura aplikacji 12](#_Toc156492107)

[4.2 Ścieżka w Aplikacji 14](#_Toc156492108)

[4.3 Szczegółowy opis aktywności w aplikacji 14](#_Toc156492109)

[4.3.1 MainActivity 14](#_Toc156492110)

[4.3.2 LoginActivity 16](#_Toc156492111)

[4.3.3 RegisterActivity 18](#_Toc156492112)

[4.3.4 MoiZnajomi 22](#_Toc156492113)

[4.3.5 mojewyniki 26](#_Toc156492114)

[4.3.6 PokazDolkowWynikowych 28](#_Toc156492115)

[4.3.7 PokazWynikowUzytkownika 30](#_Toc156492116)

[4.3.8 DodajWynik 33](#_Toc156492117)

[4.4 Adaptery oraz klasy w aplikacji 36](#_Toc156492118)

[4.4.1 Klasy Dolek i Dolekzurl 36](#_Toc156492119)

[4.4.2 Klasy Polegolfowe i PolegolfoweOdRealTima 36](#_Toc156492120)

[4.4.3 Klasy Uzytkownik i UzytkownikzUrl 37](#_Toc156492121)

[4.4.4 Klasy WynikUzytkownika i WynikUzytkownikaZUrl 37](#_Toc156492122)

[4.4.5 Adaptery 38](#_Toc156492123)

[4.5 Pliki Layout Aktywności 38](#_Toc156492124)

[4.5.1 activity\_dodaj\_wynik: 38](#_Toc156492125)

[4.5.2 activity\_login.xml 41](#_Toc156492126)

[4.5.3 activity\_main.xml 42](#_Toc156492127)

[4.5.4 activity\_moi\_znajomi.xml 44](#_Toc156492128)

[4.5.5 activity\_mojewyniki.xml 45](#_Toc156492129)

[4.5.6 activity\_pokaz\_dolkow\_wynikowych.xml 46](#_Toc156492130)

[4.5.7 activity\_pokaz\_wynikow\_uzytkownika.xml 48](#_Toc156492131)

[4.5.8 activity\_register.xml 49](#_Toc156492132)

[4.6 Pliki Layout Karty 50](#_Toc156492133)

[4.6.1 kartadolkawynikowego 50](#_Toc156492134)

[4.6.2 kartapolagolfowego.xml 52](#_Toc156492135)

[4.6.3 kartawynikuuzytkownika.xml 53](#_Toc156492136)

[4.6.4 kartaznajomego.xml 54](#_Toc156492137)

[4.7 Testowanie i znane problemy 55](#_Toc156492138)

[4.8 Problemy związane z projektem 55](#_Toc156492139)

[4.9 Pokaz aplikacji z wyświetleniem wyniku 56](#_Toc156492140)

[5.Źródła 59](#_Toc156492141)

# 1.Faza Inicjacji

## 1.1 Zespół projektowy

* Andrzej Szuwald – programista, Lider Zespołu, BackEnd
* Anna Gajos – Specjalistka do spraw UI/UE, FrontEnd
* Maksymilian Wrześniewski – Specjalista do spraw BazDanych, Tester, BackEnd

## 1.2 Use-case

Use-Case: Rejestracja rundy golfowej

**1 Brief Description**

Funkcja która umożliwia rejestracje wyników z rundy golfowej w trakcie

rozgrywki.

**2 Actor Brief Descriptions**

2.1 Użytkownik – osoba z chęcią skorzystania z naszej usługi

2.2 Administrator – pracownik obsługujący aplikację

**3 Preconditions**

W przypadku, gdy użytkownik wyrazi chęć do skorzystania z usługi będzie

musiał być zalogowany do naszej aplikacji w celu przypisania jego wyników do

konta w bazie danych.

**4 Basic Flow of Events**

1. Użytkownik wybiera opcję wprowadź wyniki.

2. Użytkownik wybiera na jakim polu golfowym się znajduje.

3. Klient wprowadza wyniki z każdego dołka.

4. Po kliknięciu zapisz wyniki zapisują się w historii użytkownika.

**5 Alternative Flows**

5.1 Przerwanie rundy

5.1.1 Gracz przerywa rundę w dowolnym momencie (np. z powodu złych warunków

pogodowych)

**6 Key Scenarios**

6.1 <scenariusz rejestracji wyników z rundy golfowej>

1. Użytkownik klika opcję rozpoczęcia nowej rundy.

2. Użytkownik wybiera odpowiednie pole golfowe z dostępnych opcji oferowanych przez

aplikację.

3. Aplikacja umożliwia wpisanie wyniku z pierwszego dołka.

4. Gracz wprowadza kolejne wyniki z dołków.

5. Po wprowadzeniu wyniku z ostatniego dołka aplikacja kończy rundę.

6. Aplikacja prezentuje tabelę z wynikami.

**7 Post conditions**

7.1 Runda zostaje zapisana w bazie.

**8 Special Requirements**

Tryb pracy offline, zabezpieczenie danych użytkowników

# 2.Faza Opracowania

## 2.1 Diagram klas

Obraz zawierający tekst, diagram, linia, Czcionka

Opis wygenerowany automatycznie

## 2.2 Wybranie języka programowania

Jednym z pierwszych problemów z jakim się zmierzono w aplikacji był wybór języka programowania. Do wyboru były możliwe dwa języki Kotlin oraz Java. Po przeanalizowaniu wykresów i statystyk na temat języków programowania, wyciągnięto wnioski, że najbardziej korzystnym rozwiązaniem będzie nauka języka java i zastosowanie go w projekcie.

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Źródło: https://slashdata-website-cms.s3.amazonaws.com/sample\_reports/\_TPqMJKJpsfPe7ph.pdf

# 3.Faza Konstrukcji

## 3.1 Stworzenie pierwszej aplikacji i uruchomienie Android Studio

Do stworzenia aplikacji wykorzystano Android Studio. Było wyzwanie biorąc pod uwagę braki w znajomości zarówno języka java, jak i samego Android Studio. Wiedzę pozyskaliśmy dzięki internetowym źródłom oraz darmowemu kursowi Android Studio programowania w Android Studio w języku java.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

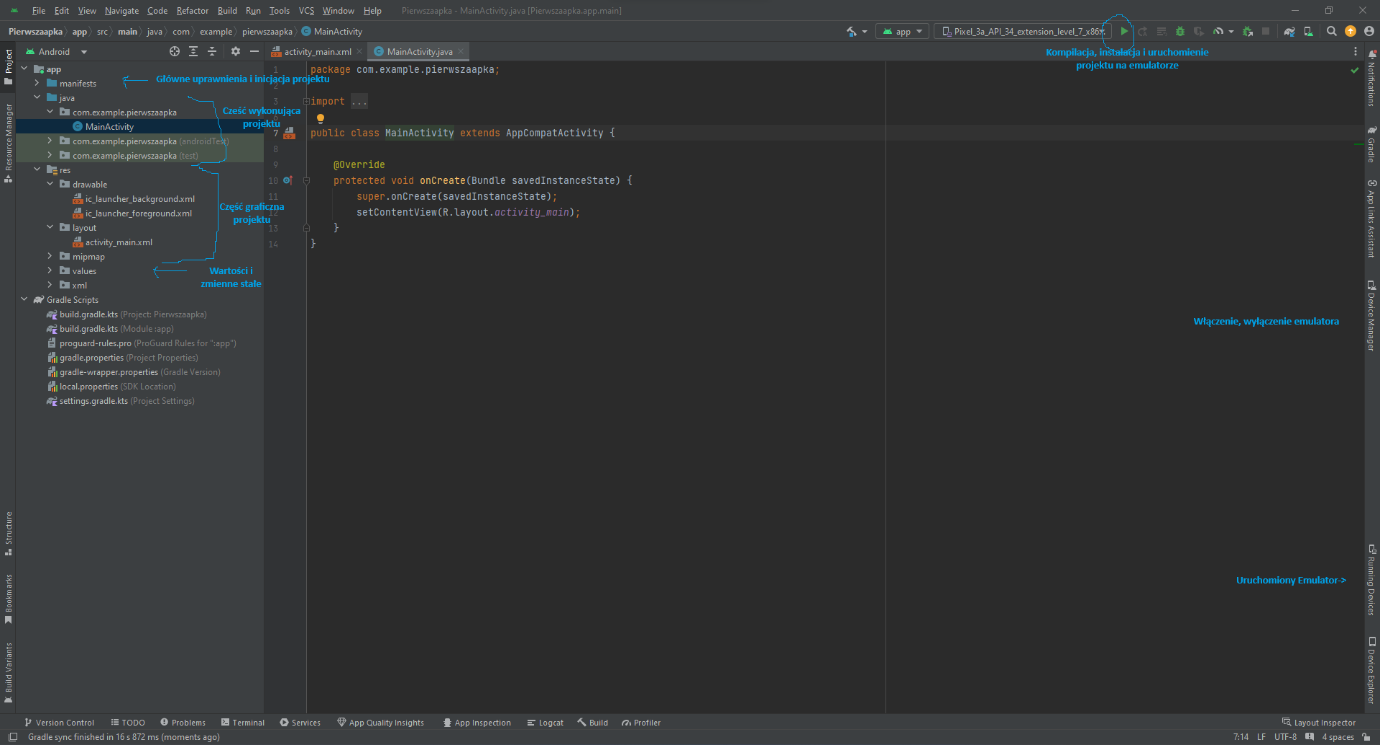
Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie



Lepsza wersja zdjęcia w dołączonym do scoreandplay.rar/PierwszaApkaObrys.png pliku.

## 3.2 Nauka android studio

Przed stworzeniem głównego projektu zajęto się nauką języka programowania i podzielono naukę różnych funkcji oraz baz danych na różne osoby w zespole.

Stworzenie pierwszej bazy danych w programie SQLiteStudio.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Projekty ułatwiające zrozumienie Android Studio.

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

## 3.3 Instalacja rozszerzeń do aplikacji

W Android Studio instalacja niezbędnych rozszerzeń przebiega z pomocą plików i skryptów Gradle.

Implementacja z pomocą Gradle w naszej aplikacji:

dependencies {

// biblioteki firebase do poprawnej obslugi aplikacji

implementation(platform("com.google.firebase:firebase-bom:32.7.0"))

implementation("com.google.firebase:firebase-auth")

implementation(platform("com.google.firebase:firebase-bom:32.7.0"))

implementation("com.google.firebase:firebase-database")

implementation("com.google.firebase:firebase-analytics")

implementation ("com.google.firebase:firebase-database")

implementation ("com.google.firebase:firebase-firestore:24.10.0")

//glide odpowiedzialny za te obrazowe aspkety apki

implementation("com.github.bumptech.glide:glide:4.11.0")

annotationProcessor("com.github.bumptech.glide:compiler:4.11.0")

//bardziej podstawowe

implementation("androidx.appcompat:appcompat:1.6.1")

implementation("com.google.android.material:material:1.11.0")

implementation("androidx.constraintlayout:constraintlayout:2.1.4")

testImplementation("junit:junit:4.13.2")

androidTestImplementation("androidx.test.ext:junit:1.1.5")

androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

}

By projekt mógł działać niezbędne jest połączenie go z Firebase. Konieczne jest dodatnie pliku google-services.json do Folderu z aplikacją.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

# 4.Faza Przekazania

## 4.1 Ogólna struktura aplikacji

MainActivity oraz AndroidManifest.xml to jedne z najważniejszych plików w całej aplikacji, dzięki któremu możliwe jest uruchomienie aplikacji i odesłanie do poszczególnych aktywności.

W AndroidManifest.xml znajdują się wszelkie pozwolenie aplikacji.

[ Dolek, Dolekzurl, PoleGolfowe, PoleGolfoweOdRealTima, Uzytkownik, UzytkownikzUrl, WynikUzytkownika, WynikUzytkownikaZUrl ] są klasami, które służą głównie do tworzenia niezbędnych do funkcjonowania aplikacji tablic np.[ ArrayList<Dolekzurl> dolki = new ArrayList<>(); ]

[ MoiZnajomiAdapterjZnajomych, PokazDolkowAdapter, PokazWynikowUzytkownikaAdapter, PoleGolfoweAdapter ] są adapterami, dzięki, którymi możliwe jest użycie RecyclerView w aplikacji i zastosowania określonych kart [ między innymi: kartadolkawynikowego.xml, kartapolagolfowego.xml, kartawynikuuzytkownika.xml, kartaznajomego.xml ]

[DodajWynik, LoginActivity, MainActivity, MoiZnajomi, mojewyniki, PokazDolkowWynikowych, PokazWynikowUzytkownika, RegisterActivity ] są aktywnościami, które posiadają kody odwołujące do poszczególnych adapterów, i plików .xml odpowiedzialnych za utworzenie części FrontEnd aplikacji.

[activity\_dodaj\_wynik.xml, activity\_login.xml, activity\_main.xml, activity\_moi\_znajomi.xml, activity\_mojewyniki.xml, activity\_pokaz\_dolkow\_wynikowych.xml, activity\_register.xml ] to są pliki .xml powiązane z aktywnościami poruszonymi wcześniej, które są odpowiedzialne za część FrontEndu

Pliki znajdujące się w folderze mipmap są plikami graficznymi wykorzystywanymi w aplikacji.

Obraz zawierający tekst, zrzut ekranu, menu

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

## 4.2 Ścieżka w Aplikacji

Obraz zawierający tekst, diagram, linia, design

Opis wygenerowany automatycznie

## 4.3 Szczegółowy opis aktywności w aplikacji

### 4.3.1 MainActivity

Inicjacja elementów UI:

private ImageView btnMojeWyniki,btnDodajWynik,btnMoiZnajomi,btnWyloguj;

W kodzie MainActivity za pomocą :

FirebaseApp.initializeApp(this);

FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();

Inicjujemy FireBase I pobieramy aktualnego użytkownika.

W odpowiednim zapytaniu warunkowym sprawdzane jest, czy użytkownik nie jest równy nullowi, jeśli nie jest równy to przenosi użytkownika do aktywności LoginActivity w celu zalogowania się. W przeciwnym wypadku następuje inicjacja widoków za pomocą funkcji ‘’initviews();’’, które mają wgląd do część frontendowej aplikacji. W późniejszej części aplikacji zostały ustawione „Nasłuchiwacze”, czyli odpowiednie funkcję śledzące czy użytkownik kliknął w dany przycisk, bądź nie.

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Użytkownik może kliknąć w jeden z 4 dostępnych przycisków

1. Po kliknięciu w przycisk Moje Wyniki użytkownik jest przenoszony do aktywności mojewyniki z dodatkową zmienną w postaci WyborOpcji[1], która pokazuje odpowiednią ścieżkę dla użytkownika
2. Po kliknięciu w przycisk Zacznij Gre użytkownik jest przenoszony do aktywności mojewyniki z dodatkową zmienną w postaci WyborOpcji[2], która pokazuje odpowiednią ścieżkę dla użytkownika
3. Po kliknięciu w przycisk Moi Znajomi użytkownik jest przenoszony do aktywności MoiZnajomi
4. Po kliknięciu w przycisk Wyloguj użytkownik zostaje wylogowany i przenosi się do aktywności LoginActivity.

### 4.3.2 LoginActivity

Inicjacja elementów UI:

private FirebaseAuth mAuth;

Button loginButton, buttonGoToRegister;

private EditText emailEditText, passwordEditText;

Na początku aktywności następuje inicjacja Firebase za pomocą:

FirebaseApp.initializeApp(this);

mAuth = FirebaseAuth.getInstance();

Ustawienie „Nasłuchiwaczy na kliknięcie danego przycisku w aplikacji”:

loginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

loginUser();

}

});

buttonGoToRegister=findViewById(R.id.btngotoregister);

buttonGoToRegister.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent= new Intent(LoginActivity.this, RegisterActivity.class);

startActivity(intent);

}

});

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

1. W przypadku kliknięcie Zarejstruj się. Użytkownik jest przenoszony do aktywności RegisterActivity.
2. W przypadku kliknięcia Zaloguj się następuje wykonanie odpowiedniego kodu, jak i funkcji loginuser().

Funkcja loginuser() jest odpowiedzialna za zalogowanie użytkownika do aplikacji poprzez sprawdzenie poprawności danych w Fire Base

Kod funkcji loginuser():

private void loginUser() {

String email = emailEditText.getText().toString().trim();

String password = passwordEditText.getText().toString().trim();

if (email.isEmpty() || password.isEmpty()) {

Toast.makeText(this, "Wprowadź e-mail i hasło", Toast.LENGTH\_SHORT).show();

return;

}

mAuth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

// Zalogowano pomyślnie

Toast.makeText(LoginActivity.this, "Logowanie udane", Toast.LENGTH\_SHORT).show();

Intent intent= new Intent(LoginActivity.this, MainActivity.class);

} else {

// Logowanie nieudane

Toast.makeText(LoginActivity.this, "Błąd logowania: " + task.getException().getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

}

### 4.3.3 RegisterActivity

Inicjacja elementów UI:

private FirebaseAuth mAuth;  
private EditText editTextEmail;  
private EditText editTextPassword;  
private EditText editTextImie,editTextNazwisko;  
private Button buttonRegister, buttonpowrotdolog;

// Inicjalizacja Firebase Authentication  
mAuth = FirebaseAuth.*getInstance*();

//Inicjacja widokow  
editTextEmail = findViewById(R.id.*editTextEmail*);  
editTextPassword = findViewById(R.id.*editTextPassword*);  
editTextImie = findViewById(R.id.*editTextImie*);  
editTextNazwisko = findViewById(R.id.*editTextNazwisko*);  
buttonRegister = findViewById(R.id.*buttonRegister*);  
buttonpowrotdolog=findViewById(R.id.*buttonwroczrejestracji*);

W poniższym kodzie sprawdzane jest czy użytkownik poprawnie podał wszystkie wartości [ e-mail, imie, nazwisko, hasło ]

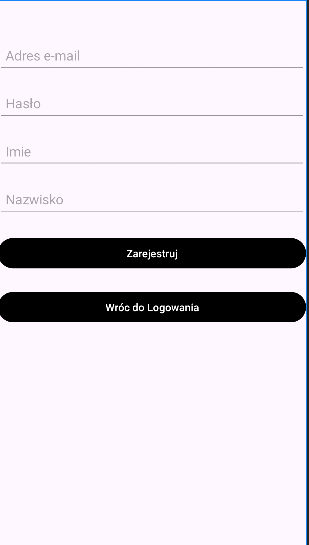
// Obsługa kliknięcia przycisku rejestracji  
buttonRegister.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // Pobierz adres e-mail i hasło z pól tekstowych  
 String email = editTextEmail.getText().toString().trim();  
 String password = editTextPassword.getText().toString().trim();  
 String imie=editTextImie.getText().toString().trim();  
 String nazwisko=editTextNazwisko.getText().toString().trim();  
 // Sprawdź, czy adres e-mail i hasło nie są puste  
 if (email.isEmpty() || password.isEmpty() || imie.isEmpty() || nazwisko.isEmpty()) {  
 Toast.*makeText*(RegisterActivity.this, "Wprowadź ponownie dane. Nie wpisales wszystkich", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }

W poniższym kodzie w pierwszym momencie tworzony jest użytkownik w bazie Firebase Authentication, która jest osobną bazą danych specjalnie zahaszowaną, a drugim momencie tworzony jest, jak na ten moment ogólno dostępnej dla użytkowników tej aplikacji bazie.

// Rejestracja użytkownika w Firebase Authentication  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(RegisterActivity.this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
  
 DatabaseReference uzytkownicyref= FirebaseDatabase.*getInstance*().getReference().child("Uzytkownicy");  
  
 uzytkownicyref.addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 long liczbaElemntow=snapshot.getChildrenCount();  
 liczbaElemntow++;  
  
 Map<String,Object> nowyUzytkownik=new HashMap<>();  
 nowyUzytkownik.put("Email",email);  
 nowyUzytkownik.put("Imie",imie);  
 nowyUzytkownik.put("ListaZnajomych","");  
 nowyUzytkownik.put("Nazwisko",nazwisko);  
  
 uzytkownicyref.child(String.*valueOf*(liczbaElemntow)).setValue(nowyUzytkownik)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 // Pomyślnie dodano wynik do bazy danych  
 System.*out*.println("Udało się dodać do bazy danych");  
 Toast.*makeText*(RegisterActivity.this, "Dodano nowy wynik!", Toast.*LENGTH\_SHORT*).show();  
  
  
 }  
 })  
 .addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 // Nie udało się dodać wyniku do bazy danych  
 Toast.*makeText*(RegisterActivity.this, "Błąd podczas dodawania wyniku.", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
  
  
  
  
 // Rejestracja zakończona sukcesem  
  
 Toast.*makeText*(RegisterActivity.this, "Rejestracja udana!", Toast.*LENGTH\_SHORT*).show();  
 Intent intent= new Intent(RegisterActivity.this, LoginActivity.class);  
 startActivity(intent);  
 } else {  
 // Rejestracja zakończona niepowodzeniem  
 Toast.*makeText*(RegisterActivity.this, "Błąd rejestracji: " + task.getException().getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
});

Poniższy kod jest aktywowany w momencie kliknięcia przycisku Wróć do logowania.

//Powrot do LoginActivity  
buttonpowrotdolog.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
Intent intent= new Intent(RegisterActivity.this,LoginActivity.class);  
 startActivity(intent);  
 }  
});



1. Przy wyborze opcji Zarejestruj aktywowany jest kod z rejestracją użytkownika. W przypadku pozytywnie rozpatrzonej przez kod rejestracji użytkownik jest przenoszony do ekranu logowania. W przeciwnym wypadku jest ukazywany komunikat o błędzie.
2. Przy wyborze opcji Wróć do Logowania użytkownik jest przenoszony do ekranu logowania.

### 4.3.4 MoiZnajomi

Inicjacja zmiennych w celu użycia widoków:

private RecyclerView MoiZnajomiRecView;  
Button btnPowrot;  
private MoiZnajomiAdapterZnajomych adapter;

Pobranie instancji oraz utworzenie widoku dla przycisku BtnPowrot

//Pobranie instacji z Firebase  
FirebaseUser currentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
if (currentUser != null) {  
 String userID = currentUser.getUid();  
 String emailaktualnegouzytkownika =currentUser.getEmail();  
 initViews();

Ustawienie „Nasłuchiwacza” w aplikacji

btnPowrot.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(MoiZnajomi.this, MainActivity.class);  
 startActivity(intent);  
 }  
});

Zainicjowanie adaptera i widoku dla Recyclerview. Wykorzystujemy tutaj dedykowany do tego Adapter MoiZnajomiAdapterZnajomych:

MoiZnajomiRecView = findViewById(R.id.*RecViewMoiZnajomi*);  
MoiZnajomiRecView.setLayoutManager(new GridLayoutManager(this, 2));  
  
adapter = new MoiZnajomiAdapterZnajomych(this);  
MoiZnajomiRecView.setAdapter(adapter);  
  
MoiZnajomiRecView.invalidate();

Utworzenie odpowiednich referencji:

//Utworzenie odpowiedniej referencji do tabeli Uzytkownicy w pliku JSON  
DatabaseReference doleRef = FirebaseDatabase.*getInstance*().getReference().child("Uzytkownicy");  
//Utworzenie referencji do odpowiedniego emaila uzytkownika  
doleRef.orderByChild("Email").equalTo(emailaktualnegouzytkownika).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.exists()) {  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 Uzytkownik uzytkownik = snapshot.getValue(Uzytkownik.class);

W tym momencie jest sprawdzane czy użytkownika posiada znajomych, jeśli nie ma to wyświetlany jest odpowiedni komunikat i użytkownik wraca do menu, czyli MainActivity:

//Wypisanie na czat uzytkownika. Funkcja sprawdzajaca poprawne dzialanie  
System.*out*.println(uzytkownik);  
uzytkownik.getListaZnajomych();  
  
if(uzytkownik.getListaZnajomych()==null){  
 Toast.*makeText*(MoiZnajomi.this, "Nie posiadasz znajomych. Powrót do Menu", Toast.*LENGTH\_SHORT*).show();  
 Intent intent= new Intent(MoiZnajomi.this, MainActivity.class);  
 startActivity(intent);  
}

Podzielenie otrzymanego stringa na poszczególne maile użytkowników:

//Z powodu tego, iż listaZnajomych dostarczana jest za pomocą stringa zawierającego listę emaili znajomych podzielonymi znakiem ',' następuje podzielenie  
String splitnieListaZnajomych[]=uzytkownik.getListaZnajomych().split(",");  
  
ArrayList<UzytkownikzUrl> znajomiuzytkownika=new ArrayList<>();

W poniższym kodzie następuje zapisanie do klasy Uzytkownik każdego znajomego naszego użytkownika:

for(int i=0;i<splitnieListaZnajomych.length;i++){  
  
 doleRef.orderByChild("Email").equalTo(splitnieListaZnajomych[i]).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.exists()) {  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 Uzytkownik uzytkownik = snapshot.getValue(Uzytkownik.class);  
  
 System.*out*.println(uzytkownik);  
 znajomiuzytkownika.add(new UzytkownikzUrl(uzytkownik.getImie(),uzytkownik.getNazwisko(),uzytkownik.getEmail(),uzytkownik.getListaZnajomych(),""));  
  
  
 }  
 System.*out*.println(znajomiuzytkownika);  
  
 adapter.setUzytkownicy(znajomiuzytkownika);  
 } else {  
 // Brak danych dla określonego IdPola  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
 // Obsługa błędów odczytu danych  
 }  
 });  
  
}

Funkcja initViews() inicjująca przycisk btnPowrot

private void initViews() {  
 btnPowrot = findViewById(R.id.*btnPrzyciskPowrotuMoiZnajomi*);  
}

### 4.3.5 mojewyniki

Inicjacja zmiennych w celu użycia widoków:

private RecyclerView PoleGolfoweRecView;  
private PoleGolfoweAdapter adapter;  
private UserSessionManager userSessionManager;  
private DatabaseReference databaseReference;

Button btnPowrot;

Inicjacja widoków dzięki initViews(); oraz odebranie odpowiednich zmiennych z poprzedniej aktywności, którymi są EmailUzytkownika, jak i WyborOpcji. Zmienna wyboropcji jest jedną z ważniejszych zmiennych, ponieważ dzięki przekazie jej do kolejnych aktywności będzie możliwe wybranie późniejszej ścieżki i otworzenie Wyników Użytkownika/Znajomego, bądź też dodanie wyniku.

W ten sposób jest możliwe zaoszczędzenie dużej ilości miejsca i zrobienie bardziej „space efficient” kodu, ponieważ nie potrzebujemy tworzyć 3 osobnych ścieżek z wyborem pola golfowego, dołka itp., a wykorzystujemy do tego jedną ścieżkę, która później dzięki tej jednej zmiennej zmienia swoją drogę na końcu dając odpowiednie wartości.

initViews();  
Intent ZActivity = getIntent();  
//odbieranie z Intenta z PoleGolfowe  
  
if (ZActivity != null) {  
 String emailuzytkownikaaktualnego = ZActivity.getStringExtra("EmailUzytkownika");  
 String wyborcopcji = ZActivity.getStringExtra("WyborOpcji");

Ustawienie przycisku powrotu:

btnPowrot.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(mojewyniki.this, MainActivity.class);  
 startActivity(intent);  
 }  
});

Ustawienie widoków I adaptera:

PoleGolfoweRecView = findViewById(R.id.*poleRecView*);  
adapter = new PoleGolfoweAdapter(this);  
PoleGolfoweRecView.setAdapter(adapter);  
PoleGolfoweRecView.setLayoutManager(new GridLayoutManager(this, 2));

// Inicjalizacja Firebase  
FirebaseUser currentUser = FirebaseAuth.*getInstance*().getCurrentUser();

Sprawdzenie czy użytkownik jest dalej zalogowany:

if (currentUser != null) {  
 String userID = currentUser.getUid();

Poniżej wczytano wszystkie Pola golfowe z tabeli polagolfowe w Real Time Database od FireBase do tablicy polagolfowe, która używa klasy PoleGolfowe. Klasa PoleGolfoweOdRealTima różni się od klasy PoleGolfowe tym, że w tym klasie PoleGolfowe istnieje zmienna imageAdress, która niestety nie jest używana w bazie danych w Real Time Database z powodu ograniczeń darmowej wersji. Przez co była potrzeba oszczędzania pamięci tej bazy danych, jednak przy odpowiedniej modyfikacji jest to jak najbardziej możliwe, by wczytywać odpowiednie zdjęcia z bazy danych.

//Utworzenie referencji do tabeli polagolfowe  
databaseReference = FirebaseDatabase.*getInstance*().getReference("/polagolfowe");  
  
// Dodawanie Listenera do odczytu danych  
databaseReference.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.exists()) {  
 ArrayList<PoleGolfowe> polagolfowe = new ArrayList<>();  
  
 // Pobieranie danych z bazy Firebase  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 PoleGolfoweOdRealTima poleGolfowe = snapshot.getValue(PoleGolfoweOdRealTima.class);  
 System.*out*.println(poleGolfowe);  
  
 polagolfowe.add(new PoleGolfowe(poleGolfowe.getIdPola(), poleGolfowe.getNazwaPola(), poleGolfowe.getMiejscowosc(), poleGolfowe.getLiczbaDolkow(), "\\https://m.media-amazon.com/images/I/91khrk365jL.\_AC\_UF894,1000\_QL80\_.jpg"));  
 }

Poniżej zaktualizowany został adapter i ustawiono potrzebne do wykonania zadań wartości zmiennych, jak i przekazano tabelę polagolfowe.

// Aktualizacja widoku RecyclerView  
adapter.setPolagolfowe(polagolfowe);  
adapter.setWyboropcji(wyborcopcji);  
adapter.setEmailUzytkownika(emailuzytkownikaaktualnego);

Funkcja initviews();

private void initViews() {  
 btnPowrot = findViewById(R.id.*btnPrzyciskPowrotu*);  
}

### 4.3.6 PokazDolkowWynikowych

Inicjacja zmiennych w celu użycia widoków:

private RecyclerView DolkiRecView;  
Button btnPowrot;  
private PokazDolkowAdapter adapter;

Odebranie z poprzedniej aktywności niezbędnych wartości i zmiennych:

Intent Dodolkow = getIntent();  
//odbieranie z Intenta z PoleGolfowe  
  
if (Dodolkow != null) {  
 String IdPolatoString = Dodolkow.getStringExtra("IdPola");  
 String EmailUzytkownika = Dodolkow.getStringExtra("EmailUzytkownika");  
 String Miejscowosc =Dodolkow.getStringExtra("Miejscowosc");  
 String wyboropcji=Dodolkow.getStringExtra("WyborOpcji");  
 String NazwaPola=Dodolkow.getStringExtra("NazwaPola");  
 int IdPola=Integer.*valueOf*(IdPolatoString);

Zainicjowanie widoków i utworzenie „odsłuchacza” do przycisku:

initViews();  
  
btnPowrot.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(PokazDolkowWynikowych.this, MainActivity.class);  
 startActivity(intent);  
 }  
});

Ustawienie widoków I adaptera:

DolkiRecView = findViewById(R.id.*poleRecViewDolki*);  
adapter = new PokazDolkowAdapter(this);  
  
  
DolkiRecView.setAdapter(adapter);  
DolkiRecView.invalidate();  
DolkiRecView.setLayoutManager(new GridLayoutManager(this, 2));

Utworzenie tabeli dla dolkow o klasie Dolekzurl:

ArrayList<Dolekzurl> dolki = new ArrayList<>();

Utworzenie referencji do dolków z określonym IdPola i pokazanie ich. W późniejszym kodzie następuje przekazanie wartości zmiennych do adaptera:

DatabaseReference doleRef = FirebaseDatabase.*getInstance*().getReference().child("Dolki");  
  
doleRef.orderByChild("IdPola").equalTo(IdPola).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.exists()) {  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 Dolek dolek = snapshot.getValue(Dolek.class);  
 dolki.add(new Dolekzurl(dolek.getIdDolka(),dolek.getIdPola(),dolek.getNazwaDolka(),dolek.getNumerDolka(),""));  
 System.*out*.println(dolek);  
  
 }  
 adapter.setWyborOpcji(wyboropcji);  
 adapter.setMiejscowosc(Miejscowosc);  
 adapter.setDoleczki(dolki);  
 adapter.setNazwaPola(NazwaPola);  
 adapter.setEmailUzytkownika(EmailUzytkownika);  
 } else {  
 // Brak danych dla określonego IdPola  
 }  
 }

Funckja initViews();

private void initViews() {  
 btnPowrot=findViewById(R.id.*btnPrzyciskPowrotuDolki*);  
  
}

### 4.3.7 PokazWynikowUzytkownika

Inicjacja zmiennych w celu użycia widoków:

Button btnPowrot;  
private RecyclerView WynikiUztykownikaRecView;  
private PokazWynikowUzytkownikaAdapter adapter;

Otrzymanie z poprzedniej aktywności wartości i zmiennych:

Intent DoWynikow = getIntent();  
//odbieranie z Intenta z PoleGolfowe  
  
if (DoWynikow != null) {  
 String IdDolkatoString = DoWynikow.getStringExtra("IdDolka");  
 String EmailUzytkownika = DoWynikow.getStringExtra("EmailUzytkownika");  
 String NazwaDolka = DoWynikow.getStringExtra("NazwaDolka");  
 System.*out*.println(IdDolkatoString);

Przekształcenie stringa IdDolkatoString na int IdDolka

int IdDolka;  
//proba przeksztalcenia IdDolkatoString na Integer  
try {  
 IdDolka = Integer.*valueOf*(IdDolkatoString);  
} catch (NumberFormatException e) {  
 e.printStackTrace();  
 return;  
}

Zainicjowanie widoków i utworzenie „odsłuchacza” do przycisku:

initViews();  
btnPowrot.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(PokazWynikowUzytkownika.this, MainActivity.class);  
 startActivity(intent);  
 }  
});

Ustawienie widoków I adaptera:

WynikiUztykownikaRecView = findViewById(R.id.*RecViewWynikiUzytkownika*);  
adapter = new PokazWynikowUzytkownikaAdapter(this);  
  
WynikiUztykownikaRecView.setAdapter(adapter);  
WynikiUztykownikaRecView.invalidate();  
WynikiUztykownikaRecView.setLayoutManager(new GridLayoutManager(this, 2));

Utworzenie tablicy z klasą WynikUzytkownikaZUrl:

ArrayList<WynikUzytkownikaZUrl> wynikiuzytkownika=new ArrayList<>();

Utworzenie referencji do wyniki\_uzytkownika w FireBase Real TimeDatabase, a później odwołanie się do tych wyników dla których IdDolka jest równe wyszukiwanemu wynikowi z danego IdDolka:

DatabaseReference doleRef = FirebaseDatabase.*getInstance*().getReference().child("wyniki\_uzytkownikow");  
  
  
doleRef.orderByChild("IdDolka").equalTo(IdDolka).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 boolean czyistniejewynik;  
 czyistniejewynik=false;  
 if (dataSnapshot.exists()) {  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 WynikUzytkownika wynik = snapshot.getValue(WynikUzytkownika.class);

Jeśli dany wynik istnieje to jest dodany do tablicy, a jeśli nie istnieje żaden to wystąpi komunikat o braku wyników:

if(wynik !=null && wynik.getEmailUzytkownika().equals(EmailUzytkownika)){  
 wynikiuzytkownika.add(new WynikUzytkownikaZUrl(wynik.getData(),wynik.getEmailUzytkownika(),wynik.getIdDolka(),wynik.getIdPola(),wynik.getIdWyniku(),wynik.getIloscUderzen(),""));  
 czyistniejewynik=true;  
 }  
 else {  
  
 }  
  
 System.*out*.println(wynik);  
  
}  
if(czyistniejewynik==false){  
 Toast.*makeText*(PokazWynikowUzytkownika.this,"Brak Wynikow z tego dolka",Toast.*LENGTH\_SHORT*).show();  
  
}

Dodanie odpowiednich wartości do adaptera:

adapter.setNazwaDolka(NazwaDolka);  
adapter.setWynikiuzytkownika(wynikiuzytkownika);  
adapter.setEmailUzytkownika(EmailUzytkownika);

funkcja initViews();

private void initViews(){  
 btnPowrot=findViewById(R.id.*btnPrzyciskPowrotuWynikiUzytkownika*);  
}

### 4.3.8 DodajWynik

Inicjacja zmiennych w celu użycia widoków:

ImageView ImgDodaj, ImgOdejmij, ImgDodajWynik;  
EditText editTextLiczbaStrzalow;  
  
TextView TextViewNazwaPola, TextViewNazwaDolka;

Wywołanie funkcji initViews();

initViews();

Pobranie zmiennych i wartości z poprzedniej aktywności

Intent Zdolkow=getIntent();  
  
if(Zdolkow!=null){  
 String IdPolatoString =Zdolkow.getStringExtra("IdPola");  
 String IdDolkatoString=Zdolkow.getStringExtra("IdDolka");  
 String EmailUzytkownikaaktualnego=Zdolkow.getStringExtra("EmailUzytkownika");  
 String NazwaPola=Zdolkow.getStringExtra("NazwaPola");  
 String NazwaDolka=Zdolkow.getStringExtra("NazwaDolka");  
 int IdDolka=Integer.*valueOf*(IdDolkatoString);  
 int IdPola=Integer.*valueOf*(IdPolatoString);

Użyta tutaj zmienna AtomicInteger była niezbędna do zrealizowania tego kodu, a najbardziej do zrealizowania ręcznego dodawania za pomocą plusa i odejmowania za pomocą minusa. Zmienna ta umożliwia zwiększanie o jeden i zmniejszanie o jeden w wielu wątkach.

AtomicInteger iloscUderzen = new AtomicInteger(0);

Wykorzystując pobrane zmienne ustawione zostają odpowiednie wartości w widokach textView.

TextViewNazwaPola.setText(String.*valueOf*(NazwaPola));  
TextViewNazwaDolka.setText(String.*valueOf*(NazwaDolka));

Po aktywacji przycisku następuje realizacja kodu, który pobiera wartość z pola EditText. W późniejszej kolejności następuje pobranie instancji użytkownika, a później referencja do wyniki\_uzytkownikow.

ImgDodajWynik.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String wartoscEditText = editTextLiczbaStrzalow.getText().toString();  
 try {  
 int wartoscLiczba = Integer.*parseInt*(wartoscEditText);  
  
  
 FirebaseUser currentUser = FirebaseAuth.*getInstance*().getCurrentUser();  
  
 if (currentUser != null) {  
 String userID = currentUser.getUid();  
  
 DatabaseReference wynikiRef = FirebaseDatabase.*getInstance*().getReference().child("wyniki\_uzytkownikow");

Później w kodzie, by zapisać odpowiednie IdWyniku wykorzystuje się ilość „dzieci” w kodzie powiększone o jeden. W późniejszym kodzie pobierana jest dzisiejsza data. Gdy pobrane zostały wszystkie wyniki. Tworzony jest obiekt mapa, który zawiera wszystkie niezbędne dane do przesłania do FireBase Real Time Database.

wynikiRef.addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 long liczbaElementow = dataSnapshot.getChildrenCount();  
 liczbaElementow++;  
  
 System.*out*.println("Liczba elementów w tabeli wyniki\_uzytkownikow: " + liczbaElementow);  
  
 String nowyIdWyniku = wynikiRef.push().getKey();  
 DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("uuuu/MM/dd");  
 LocalDate localDate = LocalDate.*now*();  
 String DzisiejszaData= localDate.toString();  
  
 // Przygotowujemy dane wyniku  
 Map<String, Object> nowyWynik = new HashMap<>();  
 nowyWynik.put("Data", DzisiejszaData);  
 nowyWynik.put("EmailUzytkownika", EmailUzytkownikaaktualnego);  
 nowyWynik.put("IdDolka", IdDolka);  
 nowyWynik.put("IdPola", IdPola);  
 nowyWynik.put("IdWyniku", liczbaElementow);  
 nowyWynik.put("IloscUderzen",wartoscLiczba);

Dzięki referencji dodajemy wynik do bazy danych, jeśli wynik zakończy się sukcesem to wyświetla się odpowiedni komunikat i użytkownik wraca do menu, czyli MainActivity. W przypadku błędu zostaje on o nim poinformowany.

wynikiRef.child(String.*valueOf*(liczbaElementow)).setValue(nowyWynik)  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 // Pomyślnie dodano wynik do bazy danych  
 System.*out*.println("Udało się dodać do bazy danych");  
 Toast.*makeText*(DodajWynik.this, "Dodano nowy wynik!", Toast.*LENGTH\_SHORT*).show();  
  
 Intent intent= new Intent(DodajWynik.this, MainActivity.class);  
 startActivity(intent);  
 }  
 })  
 .addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 // Nie udało się dodać wyniku do bazy danych  
 Toast.*makeText*(DodajWynik.this, "Błąd podczas dodawania wyniku.", Toast.*LENGTH\_SHORT*).show();  
 }  
 });

Przycisk dodaj zwiększający liczbę uderzeń.

ImgDodaj.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 iloscUderzen.incrementAndGet();  
 editTextLiczbaStrzalow.setText(String.*valueOf*(iloscUderzen));

Przycisk odejmij zmniejszający liczbę uderzeń.

ImgOdejmij.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (iloscUderzen.get() > 0) {  
 iloscUderzen.decrementAndGet();  
 editTextLiczbaStrzalow.setText(String.*valueOf*(iloscUderzen));  
 } else {  
 Toast.*makeText*(DodajWynik.this, "Nie można mieć mniej niż 0 Uderzeń", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
});

Funkcja initViews();

private void initViews() {  
 ImgDodaj=findViewById(R.id.*ImgDodaj*);  
 ImgOdejmij=findViewById(R.id.*ImgOdejmij*);  
 ImgDodajWynik=findViewById(R.id.*ImgDodajWynik*);  
 editTextLiczbaStrzalow=findViewById(R.id.*editTextLiczbaStrzalow*);  
 TextViewNazwaDolka=findViewById(R.id.*TextViewNazwaDolka*);  
 TextViewNazwaPola=findViewById(R.id.*TextViewNazwaPola*);  
  
  
}

## 4.4 Adaptery oraz klasy w aplikacji

### 4.4.1 Klasy Dolek i Dolekzurl

public class Dolek {  
 private int IdDolka;  
 private int IdPola;  
 private String NazwaDolka;  
 private int NumerDolka;

public class Dolekzurl {  
 private int IdDolka;  
 private int IdPola;  
 private String NazwaDolka;  
 private int NumerDolka;  
 private String imageAdress;

### 4.4.2 Klasy Polegolfowe i PolegolfoweOdRealTima

public class PoleGolfowe {  
 private int IdPola;  
 private String NazwaPola;  
 private String Miejscowosc;  
 private int LiczbaDolkow;  
 private String imageAdress;

public class PoleGolfoweOdRealTima {  
 private int IdPola;  
 private int LiczbaDolkow;  
 private String Miejscowosc;  
 private String NazwaPola;

### 4.4.3 Klasy Uzytkownik i UzytkownikzUrl

public class Uzytkownik {  
 private String Imie;  
 private String Nazwisko;  
 private String Email;  
 private String ListaZnajomych;

public class UzytkownikzUrl {  
 private String Imie;  
 private String nazwisko;  
 private String Email;  
 private String ListaZnajomych;  
 private String imageAdress;

### 4.4.4 Klasy WynikUzytkownika i WynikUzytkownikaZUrl

public class WynikUzytkownika {  
 private String Data;  
 private String EmailUzytkownika;  
 private int IdDolka;  
 private int IdPola;  
 private int IdWyniku;  
 private int IloscUderzen;

public class WynikUzytkownikaZUrl {  
  
 private String Data;  
 private String Email;  
 private int IdDolka;  
 private int IdPola;  
 private int IdWyniku;  
 private int IloscUderzen;  
 private String ImageAdress;

### 4.4.5 Adaptery

Każdy z adapterów jest podobny do siebie z pewnymi różnicami. W każdym mamy jakąś Tablicę związaną z daną klasą np. w MoiznajomiAdapterZnajomych jest to „ private ArrayList<UzytkownikzUrl> uzytkownicy = new ArrayList<>();”. Każda posiada swoją zmienną contextową. Każda powołuje się na odpowiedni sobie recyclerview oraz kartę np. kartaznajomego w przypadku MoiZnajomiAdapterZnajomych. W prawie każdej prócz PokazWynikowUzytkownikaAdapter ustawiony jest „Słuchacz”, który po naciśnieciu danej karty przechodzi do następnej aktywności z odpowiednimi danymi na temat danej karty i nie tylko. W każdej także następuje inicjacja widoków z RecyclerView oraz karty.

**MoiZnajomiAdapterZnajomych** – jest to adapter wykorzystywany przez aktywności MoiZnajomi. Adapter ten jest niezbędny do wykorzystanie RecyclerView oraz kartaznajomego

**PokazDolkowAdapter** – jest to adapter wykorzystywany przez aktywności PokazDolkowWynikow. Adapter ten jest niezbędny do wykorzystanie RecyclerView oraz kartadolkawynikowego

**PokazWynikowUzytkownikaAdapter** – jest to adapter wykorzystywany przez aktywności PokazWynikowUzytkownika. Adapter ten jest niezbędny do wykorzystanie RecyclerView oraz kartawynikuuzytkownika

**PolegolfoweAdapter** – jest to adapter wykorzystywany przez aktywności mojewyniki. Adapter ten jest niezbędny do wykorzystanie RecyclerView oraz kartapolagolfowego

## 4.5 Pliki Layout Aktywności

### 4.5.1 activity\_dodaj\_wynik:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".DodajWynik">  
 <ImageView  
 android:id="@+id/TopFill"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="50dp"  
 android:scaleType="fitXY"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/golfimage"  
 tools:ignore="MissingConstraints" />  
  
 <TextView  
 android:id="@+id/tytulnaglowek"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/score\_play"  
 android:textColor="#FFFFFF"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="@+id/TopFill"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill" />  
  
 <EditText  
 android:id="@+id/editTextLiczbaStrzalow"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="60dp"  
 android:ems="10"  
 android:inputType="number"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TopFill" />  
  
 <ImageView  
 android:id="@+id/ImgDodaj"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="20dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.15"  
 app:layout\_constraintStart\_toEndOf="@+id/editTextLiczbaStrzalow"  
 app:layout\_constraintTop\_toBottomOf="@+id/TopFill"  
 app:srcCompat="@mipmap/dodaj500x500" />  
  
 <ImageView  
 android:id="@+id/ImgOdejmij"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.15"  
 app:layout\_constraintStart\_toEndOf="@+id/editTextLiczbaStrzalow"  
 app:layout\_constraintTop\_toBottomOf="@+id/ImgDodaj"  
 app:srcCompat="@mipmap/minus500x500" />  
  
 <TextView  
 android:id="@+id/TextViewNazwaPola"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:text="Nazwa Pola"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ImgOdejmij" />  
  
 <TextView  
 android:id="@+id/TextViewNazwaDolka"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:text="Nazwa Dolka"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextViewNazwaPola" />  
  
 <ImageView  
 android:id="@+id/ImgDodajWynik"  
 android:layout\_width="376dp"  
 android:layout\_height="240dp"  
 android:layout\_marginTop="8dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextViewNazwaDolka"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/dodajwynik2500x500" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### 4.5.2 activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".LoginActivity">  
  
 <EditText  
 android:id="@+id/editTextEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="E-mail"  
 android:inputType="textEmailAddress" />  
  
 <EditText  
 android:id="@+id/editTextPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editTextEmail"  
 android:layout\_marginTop="16dp"  
 android:hint="Hasło"  
 android:inputType="textPassword"  
 android:minHeight="48dp" />  
  
 <Button  
 android:id="@+id/buttonLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editTextPassword"  
 android:layout\_marginTop="16dp"  
 android:text="Zaloguj się"  
 android:backgroundTint="@color/black"  
 android:layout\_centerHorizontal="true"/>  
 <Button  
 android:id="@+id/btngotoregister"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/buttonLogin"  
 android:layout\_marginTop="16dp"  
 android:text="Zarejestruj się"  
 android:backgroundTint="@color/black"  
 android:layout\_centerHorizontal="true"/>  
</RelativeLayout>

### 4.5.3 activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/TopFill"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="50dp"  
 android:scaleType="fitXY"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/golfimage"  
 tools:ignore="MissingConstraints" />  
  
 <ImageView  
 android:id="@+id/BottomFill"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="1.0"  
 app:srcCompat="@mipmap/golfimage"  
 android:scaleType="fitXY"  
 />  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_marginTop="15dp"  
 android:src="@mipmap/osoba500x500"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TopFill" />  
  
 <TextView  
 android:id="@+id/tytulnaglowek"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/score\_play"  
 android:textColor="#FFFFFF"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="@+id/TopFill"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill" />  
  
 <ImageView  
 android:id="@+id/btnMojeWyniki"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"  
 app:srcCompat="@mipmap/mojewyniki2000x500" />  
  
 <ImageView  
 android:id="@+id/btnDodajWynik"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnMojeWyniki"  
 app:srcCompat="@mipmap/zacznijgre2000x500" />  
  
 <ImageView  
 android:id="@+id/btnmoiznajomi"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnDodajWynik"  
 app:srcCompat="@mipmap/moiznajomi2000x500" />  
  
 <ImageView  
 android:id="@+id/btnWyloguj"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="16dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/BottomFill"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnmoiznajomi"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/wyloguj2000x500" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### 4.5.4 activity\_moi\_znajomi.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MoiZnajomi">  
  
 <ImageView  
 android:id="@+id/TopFill"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="50dp"  
 android:scaleType="fitXY"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/golfimage"  
 tools:ignore="MissingConstraints" />  
  
 <TextView  
 android:id="@+id/tytulnaglowek"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/score\_play"  
 android:textColor="#FFFFFF"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="@+id/TopFill"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/RecViewMoiZnajomi"  
 android:layout\_marginTop="60dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="1dp"  
 android:layout\_marginLeft="20dp"  
 />  
  
 <Button  
 android:id="@+id/btnPrzyciskPowrotuMoiZnajomi"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="Wróc"  
 android:backgroundTint="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="@+id/RecViewMoiZnajomi"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill"  
 app:layout\_constraintVertical\_bias="1.0" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### 4.5.5 activity\_mojewyniki.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".mojewyniki">  
  
 <ImageView  
 android:id="@+id/TopFill"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="50dp"  
 android:scaleType="fitXY"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/golfimage"  
 tools:ignore="MissingConstraints" />  
  
 <TextView  
 android:id="@+id/tytulnaglowek"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/score\_play"  
 android:textColor="#FFFFFF"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="@+id/TopFill"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill" />  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/poleRecView"  
 android:layout\_marginTop="60dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="1dp"  
 android:layout\_marginLeft="20dp"  
 />  
  
 <Button  
 android:id="@+id/btnPrzyciskPowrotu"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="Wróc"  
 android:backgroundTint="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="@+id/poleRecView"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill"  
 app:layout\_constraintVertical\_bias="1.0" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### 4.5.6 activity\_pokaz\_dolkow\_wynikowych.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".PokazDolkowWynikowych">  
  
 <ImageView  
 android:id="@+id/TopFill"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="50dp"  
 android:scaleType="fitXY"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/golfimage"  
 tools:ignore="MissingConstraints" />  
  
 <TextView  
 android:id="@+id/tytulnaglowek"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/score\_play"  
 android:textColor="#FFFFFF"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="@+id/TopFill"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill" />  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/poleRecViewDolki"  
 android:layout\_marginTop="60dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="1dp"  
 android:layout\_marginLeft="20dp"  
 />  
  
 <Button  
 android:id="@+id/btnPrzyciskPowrotuDolki"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="Wróc"  
 android:backgroundTint="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="@+id/poleRecViewDolki"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill"  
 app:layout\_constraintVertical\_bias="1.0" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### 4.5.7 activity\_pokaz\_wynikow\_uzytkownika.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".PokazWynikowUzytkownika">  
 <ImageView  
 android:id="@+id/TopFill"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="50dp"  
 android:scaleType="fitXY"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@mipmap/golfimage"  
 tools:ignore="MissingConstraints" />  
  
 <TextView  
 android:id="@+id/tytulnaglowek"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/score\_play"  
 android:textColor="#FFFFFF"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="@+id/TopFill"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill" />  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/RecViewWynikiUzytkownika"  
 android:layout\_marginTop="60dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="1dp"  
 android:layout\_marginLeft="20dp"  
 />  
  
 <Button  
 android:id="@+id/btnPrzyciskPowrotuWynikiUzytkownika"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="Wróc"  
 android:backgroundTint="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="@+id/RecViewWynikiUzytkownika"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/TopFill"  
 app:layout\_constraintVertical\_bias="1.0" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### 4.5.8 activity\_register.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".RegisterActivity">  
  
  
 <EditText  
 android:id="@+id/editTextEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:hint="Adres e-mail"  
 android:inputType="textEmailAddress"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
  
 <EditText  
 android:id="@+id/editTextPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editTextEmail"  
 android:layout\_marginTop="16dp"  
 android:hint="Hasło"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
  
 <EditText  
 android:id="@+id/editTextImie"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editTextPassword"  
 android:layout\_marginTop="16dp"  
 android:hint="Imie"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
  
 <EditText  
 android:id="@+id/editTextNazwisko"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editTextImie"  
 android:layout\_marginTop="16dp"  
 android:hint="Nazwisko"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
  
  
 <Button  
 android:id="@+id/buttonRegister"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editTextNazwisko"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="24dp"  
 android:backgroundTint="@color/black"  
 android:text="Zarejestruj" />  
 <Button  
 android:id="@+id/buttonwroczrejestracji"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/buttonRegister"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="24dp"  
 android:backgroundTint="@color/black"  
 android:text="Wróc do Logowania" />  
  
</RelativeLayout>

## 4.6 Pliki Layout Karty

### 4.6.1 kartadolkawynikowego

<?xml version="1.0" encoding="utf-8"?>  
<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/parentkarta2"  
 app:cardCornerRadius="7dp"  
 app:cardElevation="7dp"  
 android:layout\_margin="15dp"  
 >  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <ImageView  
 android:layout\_width="130dp"  
 android:layout\_height="150dp"  
 android:id="@+id/imgBook"  
 android:src="@mipmap/ic\_launcher"  
 />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Tytul Pola"  
 android:textStyle="bold"  
 android:textSize="16sp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 android:id="@+id/txtDolekTytulPola"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textStyle="bold"  
 android:textSize="12sp"  
 android:id="@+id/txtNazwaDolka"  
 android:layout\_below="@+id/txtDolekTytulPola"  
 android:text="Miejscowosc"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="12sp"  
 android:id="@+id/txtDolki"  
 android:layout\_below="@+id/txtNazwaDolka"  
 android:text="Numer Dolka"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"/>  
  
  
  
  
 </RelativeLayout>  
  
</com.google.android.material.card.MaterialCardView>

### 4.6.2 kartapolagolfowego.xml

<?xml version="1.0" encoding="utf-8"?>  
<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/parentkarta"  
 app:cardCornerRadius="7dp"  
 app:cardElevation="7dp"  
 android:layout\_margin="15dp"  
 >  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <ImageView  
 android:layout\_width="130dp"  
 android:layout\_height="150dp"  
 android:id="@+id/imgBook"  
 android:src="@mipmap/ic\_launcher"  
 />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Tytul Pola"  
 android:textStyle="bold"  
 android:textSize="16sp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 android:id="@+id/txtTytulPola"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textStyle="bold"  
 android:textSize="12sp"  
 android:id="@+id/txtMiejscowoscPola"  
 android:layout\_below="@+id/txtTytulPola"  
 android:text="Miejscowosc"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="12sp"  
 android:id="@+id/txtDolki"  
 android:layout\_below="@+id/txtMiejscowoscPola"  
 android:text="Liczba Dolkow"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"/>  
  
  
  
  
 </RelativeLayout>  
  
</com.google.android.material.card.MaterialCardView>

### 4.6.3 kartawynikuuzytkownika.xml

<?xml version="1.0" encoding="utf-8"?>  
<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/parentkartaWynikuUzytk"  
 app:cardCornerRadius="7dp"  
 app:cardElevation="7dp"  
 android:layout\_margin="15dp"  
 >  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <ImageView  
 android:layout\_width="130dp"  
 android:layout\_height="150dp"  
 android:id="@+id/imgBook"  
 android:src="@mipmap/ic\_launcher"  
 />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Data Wyniku"  
 android:textStyle="bold"  
 android:textSize="16sp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 android:id="@+id/txtDataWyniku"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textStyle="bold"  
 android:textSize="12sp"  
 android:id="@+id/txtIloscUderzen"  
 android:layout\_below="@+id/txtDataWyniku"  
 android:text="Ilosc Uderzen"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="12sp"  
 android:id="@+id/txtWynikNazwaDolka"  
 android:layout\_below="@+id/txtIloscUderzen"  
 android:text="Dolek"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"/>  
  
 </RelativeLayout>  
  
</com.google.android.material.card.MaterialCardView>

### 4.6.4 kartaznajomego.xml

<?xml version="1.0" encoding="utf-8"?>  
<com.google.android.material.card.MaterialCardView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/parentkartaZnajomy"  
 app:cardCornerRadius="7dp"  
 app:cardElevation="7dp"  
 android:layout\_margin="15dp"  
 >  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <ImageView  
 android:layout\_width="130dp"  
 android:layout\_height="150dp"  
 android:id="@+id/imgBook"  
 android:src="@mipmap/ic\_launcher"  
 />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Imie"  
 android:textStyle="bold"  
 android:textSize="16sp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 android:id="@+id/txtImieUzytkownika"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textStyle="bold"  
 android:textSize="12sp"  
 android:id="@+id/txtNazwiskoUzytkownika"  
 android:layout\_below="@+id/txtImieUzytkownika"  
 android:text="Nazwisko"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="7dp"  
 />  
  
  
  
  
  
 </RelativeLayout>  
  
</com.google.android.material.card.MaterialCardView>

## 4.7 Testowanie i znane problemy

Program został przetestowanych na różnych wartościach. Gdy wartości są puste nie powoduje to błędu w programie. W przypadku wystąpienia problemu program się nie „wysypuje”, ale informuje użytkownika dzięki np. funkcji catch() lub onfailure(), że coś poszło nie tak i próbuje ponownie wykonać akcję. Gdy użytkownik stara się wyświetlić wyniki dla danego użytkownika, jednak on nie ma wyników to nie powoduje to błędu, jednak wyświetlona jest informacja o braku wyników.

## 4.8 Problemy związane z projektem

Jednym z największych problemów okazało się wybranie odpowiedniej bazy danych dla projektu wraz z serwerem go obsługującym. Próby połączenia z bazą danych lokalną na danym komputerze było niemożliwe czego dokładny powód nie jest dokładnie znany do dzisiaj. Jedynie przypuszczenia jakie występują to blokowanie na poziomie procesu przesyłania z jednej aplikacji do drugiej między jedną aplikacją, a drugą. Być może właśnie na jakimś etapie przesłania nie posiadał ten proces uprawnień odpowiednich i nie było bezpośrednio możliwości nadania. Sprawdzono też dokładnie porty czy dany port nie był zablokowany. W tym uruchamianie ręczne usług z poziomu wiersza poleceń z uprawnieniami administratora.

Przetestowane aplikacje z lokalnymi bazami danych:

* MySQL Workbench
* Xampp
* sql server management studio
* SQL server

Końcowo najlepszym rozwiązaniem okazało się rozwiązanie chmurowe od Firebase w postaci Realtime Database oraz Authentication

## 4.9 Pokaz aplikacji z wyświetleniem wyniku

Menu Aplikacji

Obraz zawierający tekst, Telefon komórkowy, Urządzenie przenośne, gadżet

Opis wygenerowany automatycznie

Wybór Moje Wyniki

Obraz zawierający tekst, Telefon komórkowy, zrzut ekranu, gadżet

Opis wygenerowany automatycznie

Wybór Machał i Wymachał pola golfowego

Obraz zawierający tekst, Telefon komórkowy, zrzut ekranu, gadżet

Opis wygenerowany automatycznie

Wybranie Pod Górką

Obraz zawierający Telefon komórkowy, gadżet, Urządzenie przenośne, Komunikator

Opis wygenerowany automatycznie

# 5.Źródła

<https://firebase.google.com/docs?hl=pl>

<https://www.youtube.com/@android_knowledge>

<https://www.youtube.com/watch?v=fis26HvvDII>

<https://www.youtube.com/watch?v=RcSHAkpwXAQ&t=0s>

<https://www.youtube.com/watch?v=HK515-8-Q_w&list=PLS1QulWo1RIbVgr0GRuQW5q_K0zb3rrct>