

Aplikace – RugbyTrainingApp

Kryštof Uchytíl

1. Popis ovládání

Moje aplikace s názvem Rugbytraining, je pro jakéhokoliv uživatele, graficky nenáročná a přehledná.

Po spuštění aplikace se uživatel musí přihlásit přes login, pokud ještě účet nemá, musí se registrovat. K tomu mu stačí email ve správném formátu a heslo, tyto informace o účtech se ukládají do databáze.

Po přihlášení se objeví hlavní obrazovka, která pomocí bottonů odkazuje na další obrazovky, uživatel jako první musí do profilu, aby si nastavil výchozí hodnoty tělesné tak i informativní. Po vyplnění údajů se profil uloží do databáze.

Poté když uživatel přijde na obrazovku tréninkový plán, tak pomocí vyplněných údajů z profilu, se mu vygeneruje plán. Plus přes GPS lokaci určí nejbližší cvičiště.

V poslední obrazovce statistik, uživatel vloží údaje a podle těchto údajů porovnaných s předchozími údaji v profilu, se vytvoří statistika.

Toto je celá funkčnost aplikace a popis jejího použití.

2. Layouty a design

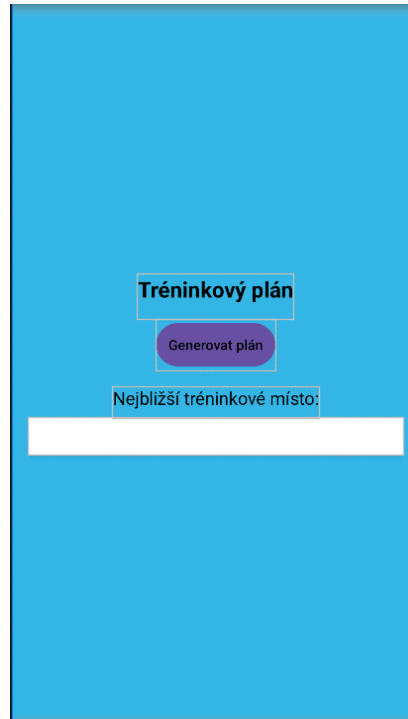
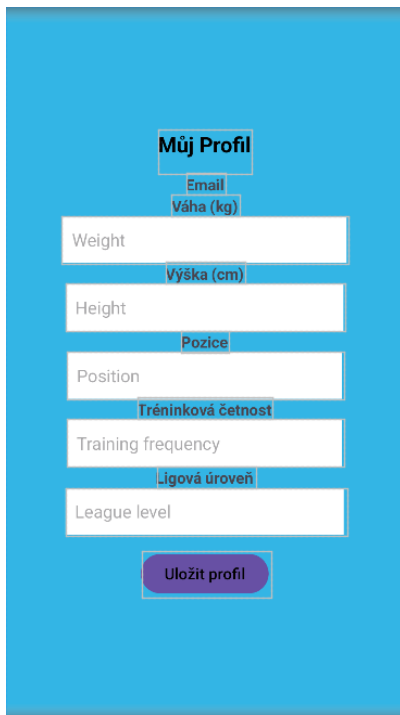
Celou aplikaci vedeme v modré barvě propojenou s fialovými tlačítky a v některých částech i bílé textové pole. Všechny komponenty jsou přes wrap content aby obsadili daný text. První dvě obrazovky Login a Registr, email, password a button

```
<EditText
    android:id="@+id/et_email"
    android:layout_width="340dp"
    android:layout_height="wrap_content"
    android:hint="@string/hint_email"
    android:inputType="textEmailAddress"
    android:autofillHints="emailAddress"
    android:textColorHint="@color/black"
    android:minHeight="48dp"
    android:padding="10dp" />
<EditText
    android:id="@+id/et_password"
    android:layout_width="345dp"
    android:layout_height="wrap_content"
    android:hint="@string/hint_password"
    android:inputType="textPassword"
    android:textColorHint="@color/black"
    android:autofillHints="textPassword"
    android:minHeight="48dp"
    android:padding="10dp" />
```

```
<Button
    android:id="@+id/btn_registr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/btn_register"
    android:textColor="@android:color/black"
    android:layout_marginTop="10dp"/>
```

Po přihlášení nebo registraci se objeví hlavní obrazovka, která je pouze přechodná pro další obrazovky.

Z hlavní obrazovky se dá přejít do osobního profilu, tréninkového plánu a statistik



V obrazovce můj profil vidíme jednotlivá pole, které vyplní uživatel. Pole jsou brány jako string.

```
<EditText
    android:id="@+id/et_profile_weight"
    android:layout_width="293dp"
    android:layout_height="wrap_content"
    android:background="@android:color/white"
    android:hint="@string/hint_weight"
    android:inputType="number"
    android:minHeight="48dp"
    android:autofillHints="profileWeight"
    android:padding="10dp" />
```

Zde se vypíše tréninkový plán

```
<TextView
    android:id="@+id/tv_training_plan_result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="16sp"
    android:textColor="@android:color/black"
    android:padding="8dp"
    android:background="@android:color/white"
    android:elevation="2dp"
    android:paddingHorizontal="12dp"
    android:paddingVertical="8dp"
    android:gravity="center"
```

Statistiky jsou vypisovány textView.

```
<TextView
    android:id="@+id/tv_distance"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:text="@string/tv_distance"
    android:textColor="@android:color/black"
    android:textSize="18sp" />
```

3) Funkce

Zde je ukázka MainActivity.java a přesměrování na ostatní obrazovky, přes funkci new Intent

```
public class MainActivity extends AppCompatActivity {

    2 usages
    private Button btnProfile, btnTrainingPlan, btnStatistics;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnProfile = findViewById(R.id.btn_profile);
        btnTrainingPlan = findViewById(R.id.btn_training_plan);
        btnStatistics = findViewById(R.id.btn_statistics);

        btnProfile.setOnClickListener(v -> startActivity(new Intent( packageContext: MainActivity.this, ProfileActivity.class)));
        btnTrainingPlan.setOnClickListener(v -> startActivity(new Intent( packageContext: MainActivity.this, TrainingPlanActivity.class)));
        btnStatistics.setOnClickListener(v -> startActivity(new Intent( packageContext: MainActivity.this, StatisticsActivity.class)));
    }
}
```

Tady ukázaná logika v přihlášení, která přes databázi kontroluje users, zda-li už mají účet

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    dbHelper = new DatabaseHelper( context: this);
    etEmail = findViewById(R.id.et_email);
    etPassword = findViewById(R.id.et_password);
    btnLogin = findViewById(R.id.btn_login);
    btnRegister = findViewById(R.id.btn_register);

    btnLogin.setOnClickListener(v -> {
        String email = etEmail.getText().toString().trim();
        String password = etPassword.getText().toString().trim();

        if (dbHelper.checkUser(email, password)) {
            Toast.makeText( context: LoginActivity.this, text: "Přihlášení úspěšné!", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent( packageContext: LoginActivity.this, MainActivity.class);
            intent.putExtra( name: "USER_EMAIL", email);
            startActivity(intent);
            finish();
        } else {
            Toast.makeText( context: LoginActivity.this, text: "Špatné údaje!", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Další obrazovkou je profile kde se ukládají údaje o uživateli

```
private void loadUserProfile() {
    Cursor cursor = dbHelper.getUserProfile(userEmail);
    if (cursor != null && cursor.moveToFirst()) {
        etHeight.setText(cursor.getString(cursor.getColumnIndexOrThrow( columnName: "height")));
        etWeight.setText(cursor.getString(cursor.getColumnIndexOrThrow( columnName: "weight")));
        etPosition.setText(cursor.getString(cursor.getColumnIndexOrThrow( columnName: "position")));
        etFrequency.setText(cursor.getString(cursor.getColumnIndexOrThrow( columnName: "training_frequency")));
        etLevel.setText(cursor.getString(cursor.getColumnIndexOrThrow( columnName: "league_level")));
        cursor.close();
    }
}

1 usage
private void saveUserProfile() {
    String heightStr = etHeight.getText().toString().trim();
    String weightStr = etWeight.getText().toString().trim();
    String position = etPosition.getText().toString().trim();
    String frequencyStr = etFrequency.getText().toString().trim();
    String level = etLevel.getText().toString().trim();

    if (TextUtils.isEmpty(heightStr) || TextUtils.isEmpty(weightStr) || TextUtils.isEmpty(position) || TextUtils.isEmpty(frequencyStr) || TextUtils.isEmpty(level)) {
        Toast.makeText( context: this, text: " ! Vyplňte všechna pole!", Toast.LENGTH_SHORT).show();
        return;
    }

    try {
        double height = Double.parseDouble(heightStr);
        double weight = Double.parseDouble(weightStr);
        int frequency = Integer.parseInt(frequencyStr);
    }
```

Obrazovka tréninkového plánu a statistik funguje na podobném principu, kdy bere údaje a poté něco vypíše.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_training_plan);

    // Inicializace UI prvků
    tvNearestTrainingSpot = findViewById(R.id.tv_nearest_training_spot);
    tvTrainingPlanResult = findViewById(R.id.tv_training_plan_result);
    btnGeneratePlan = findViewById(R.id.btn_generate_plan);

    // Inicializace databáze a GPS klienta
    dbHelper = new DatabaseHelper( context: this);
    fusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this);

    // Načtení e-mailu uživatele ze SharedPreferences
    SharedPreferences prefs = getSharedPreferences( name: "UserPrefs", Context.MODE_PRIVATE);
    userEmail = prefs.getString( key: "USER_EMAIL", defValue: null);

    if (userEmail == null || userEmail.isEmpty()) {
        Toast.makeText( context: this, text: "✗ Chyba: E-mail uživatele není dostupný.", Toast.LENGTH_LONG).show();
        finish(); // Ukončí aktivitu, protože bez e-mailu nemůžeme pokračovat
        return;
    }
}
```

Kde u tréninkového plánu je ještě lokace pomocí GPS

```
1 usage
private String findNearestTrainingSpot(Location location)
    // Tady by bylo reálné vyhledávání na základě databáze
    return "Městský stadion"; // Pevná hodnota pro testování
}

// Generování tréninkového plánu podle lokality
1 usage
private String generatePlan(String nearestSpot) {
    return "Váš tréninkový plán u místa: " + nearestSpot;
}
```

4) Problémy, nedostatky a pocity

Aplikace má nedostatky, absolutně bezchybně funguje login s registrem a obrazovka s profilem. Tréninkový plán se generuje, ale není to zcela podle hodnot uživatele, lokace není plně funkční a statistiky kvůli špatnému plánu nemohou být generovány. Jinak jsem rád, že aplikace má aspoň základní funkce a lehčí úprava by jí celou dokončila.