

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
PREŠOV

IV.SB - 15

ŠK. R. 2019 – 2020

PČOZ – HUDOBNÝ PREHRÁVAČ

SEBASTIÁN MARCIN

Konzultant: Ing. Peter Vargovčík

Anotácia v slovenskom jazyku

Zamerali sme sa na naprogramovanie PC aplikácie hudobného prehrávača v programovacom jazyku Java, ktorý bude využívať databázu užívateľov a hudobných skladieb (databáza v jazyku SQL-lite). Prehrávač bude priebežne zobrazovať text skladby (titulky). Naprogramovanie funkcionality administrátora pre správu užívateľov a skladieb, možnosť ohodnotiť jednotlivé skladby užívateľom.

Anotácia v anglickom jazyku

We focused on programming the PC application of the music player in the Java programming language, which will use a database of users and music tracks (database in SQL-lite language). The player will continuously display the text of the song (subtitles). the ability to rate individual songs by users.

Čestné vyhlásenie

Vyhlasujem, že celú prácu stredoškolskej odbornej činnosti na tému „PC aplikácia Hudobný prehrávač“ som vypracoval/a samostatne, s použitím uvedenej literatúry.

Vyhlasujem, že danú prácu som neprihlásil a neprezentoval v žiadnej inej súťaži, ktorá je schválená Ministerstvom školstva, vedy, výskumu a športu SR.

Som si vedomý zákonných dôsledkov, ak v nej uvedené údaje nie sú pravdivé.

Prešov, 15. február 2020

.....

vlastnoručný podpis

Pod'akovanie

Radi by sme poďakovali pánovi Ing. Peter Vargovčík. Za všetky rady a odbornú pomoc pri tvorbe projektu ako aj pomoc pri výbere témy.

Obsah

Úvod	5
1 Cieľ práce a metodika práce	6
2 Hudobný prehrávač Pulse – popis funkcionality	7
3 Využívané technológie	8
3.1 Java	8
3.2 Scene Builder	8
3.3 SQL Lite.....	8
3.4 Firebase	8
3.5 CSS	8
4 JavaFX.....	9
4.1 Model-View-Controller (MVC)	10
5 Použité knižnice	11
5.1 SQLite-JDBC	11
5.2 JFoenix.....	11
6 Návrh grafického dizajnu	12
7 Tvorba databázy	14
7.1 Návrh databázy	14
7.2 Tvorba tabuliek databázy	14
7.3 Vytvorenie úložiska – FireBase	15
16	
7.4 Použitie úložiska v databáze	16
8 Funkčnosť aplikácie	17
8.1 Hudobný prehrávač	17
8.2 Live vyhľadávanie	18
8.3 Priebežne zobrazovanie textu skladby	18
8.4 Tvorba recenzie	21
8.5 Administrátorské rozhranie.....	22
9 Výsledky práce a diskusia.....	23
10 Závery práce	24
Zhrnutie.....	25
Resumé.....	26
Použitá literatúra	27

Úvod

Hudba je súčasťou každodenného života každého z nás. Dnešná doba ľuďom umožňuje počúvať svoje obľúbené kapely alebo spevákov hocikedy a hocikde. Populárne sú hlavne streamovacie služby, ktoré umožňujú jednoduchý prístup k hudbe.

Rozhodli sme sa vytvoriť vlastnú počítačovú aplikáciu, ktorá by mohla konkurovať komerčným platformám. Naša aplikácia obsahuje taktiež zaujímavú funkciu, ktorá zobrazuje priebežne text piesne. Vďaka tejto funkcionalite naša aplikácia dokáže konkurovať iným hudobným aplikáciám.

Počítačová aplikácia umožňuje prístup k hudbe, albumom a interpretom, ktorý si uloží v databáze. Užívateľ si môže vyhľadávať jednotlivé albumy alebo konkrétne piesne či interpretov. Jednotlivé piesne môžu byť ohodnotené slovnou recenziou ale rovnako aj číselným ohodnotením, ktoré je reprezentované prostredníctvom obrázka hviezd. Slovné recenzie podliehajú kontrole administrátora. Administrátor môže užívateľovi, ktorý píše nevhodné recenzie zakázať prístup do aplikácie. Každá pieseň sa môže stať obľúbenou, čo znamená že sa pridá do zoznamu obľúbených piesní konkrétného užívateľa.

Dôvody vytvorenia práce :

- Vytvorenie konkurencie schopného hudobného prehrávača
- Vytvorenie zaujímavých funkcionalít

1 Cieľ práce a metodika práce

Cieľom práce bolo vytvoriť počítačovú aplikáciu – hudobný prehrávač, ktorá bude určená širokej verejnosti.

Užívateľ bude vedieť prehrávať skladby uložené v databáze, uvidí taktiež priebežne text skladby. Bude mať možnosť ohodnotiť skladby slovne ale aj číselne. Administrátor bude vedieť blokovať alebo povoliť prístup k aplikácii registrovaným užívateľom.

Po vybratí konzultanta a schválení témy sme začali premýšľať nad jednotlivými funkcionalitami a ich prevedením. Ďalším krokom bolo vybratie vhodného dizajnu pre jednotlivé scény aplikácie. Pre našu aplikáciu sú základom hudobné skladby. Skladby sme potrebovali uložiť. Vytvorili sme úložisko pomocou FireBase. Následne sme vytvorili komplexnú databázu s potrebnými tabuľkami údajov, ktorú sme prepojili s úložiskom FireBase.

V ďalšom kroku sme vytvorili priečinkovú štruktúru pre náš projekt. Priečinkovú štruktúru sme vytvorili podľa MVC architektúry. Vytvorili sme podpriečinky „Controllers“, ktorý bude slúžiť pre kontrolér ku jednotlivým scénam, „Models“, kde budú uložené entity a „FXMLs“, podpriečinek pre grafické súbory scén.

Následne sme naplnili úložisko skladbami, databázu potrebnými údajmi. Postupne sme riešili každú požadovanú funkcionalitu až sme našu aplikáciu zdarne dokončili.

2 Hudobný prehrávač Pulse – popis funkcionality

Po zapnutí sa užívateľovi zobrazí prihlasovacia scéna, kde si môže vybrať z možností prihlásenie alebo registrácia. Pri vybratí registrácie je potrebné správne vyplniť registračný formulár. Následne je užívateľ úspešne registrovaný a môže sa prihlásiť do aplikácie.

Po úspešnom prihlásení sa užívateľovi zobrazí scéna, kde sa na ľavej strane nachádzajú tlačidlá na prepínanie medzi zobrazovaním albumov, interpretov alebo konkrétnych skladieb. V hornej časti scény je vyhľadávací panel, ktorý slúži na vyhľadávanie skladieb, albumov a interpretov. Výsledky vyhľadávania sa priebežne zobrazujú v hlavnej časti scény.

Po kliknutí tlačidla „Artists“ sa užívateľovi zobrazia interpreti a vybratí interpreta sa zobrazia interpretove albumy a následne skladby. Rovnako to funguje aj s tlačidlami „Albums“ a „Songs“.

Po dvojkliku na konkrétnu skladbu sa skladba spustí. V dolnej časti scény sa zobrazujú informácie o skladbe a dva bežce. Jeden slúži na pretáčanie skladby a druhý na ovládanie hlasitosti. Ďalej sa tam nachádzajú tlačidlá start/pause a stop. Každú skladbu vie užívateľ ohodnotiť v pravej časti scény. Po potvrdení sa recenzie zobrazia. Recenzie môže písať každý užívateľ.

Jednotlivé recenzie kontroluje administrátor, ktorý po kontrole môže užívateľovi zakázať prístup do aplikácie. Taktiež administrátor vidí kompletný zoznam užívateľov, ktorým taktiež vie zakázať alebo povoliť prístup do aplikácie.

3 Využívané technológie

V tomto projekte sme využili overené technológie a postupy určené na vytváranie aplikácií v programovacom jazyku Java. Základom našej aplikácie je platforma JavaFX.

3.1 Java

Java je objektovo orientovaný programovací jazyk. Je vyvíjaný spoločnosťou Oracle. Jeho syntax vychádza z jazykov C a C++. Zdrojové programy sa nekompilujú do strojového kódu, ale do medzistupňa, tzv. „byte-code“, ktorý nie je závislý od konkrétnej platformy. Tento byte-code neskôr vykonáva a spracováva interpreter, Java Virtual Machine.[1]

3.2 Scene Builder

Scene Builder je vizuálny rozvrhovací nástroj, ktorý umožňuje používateľom rýchlo navrhovať používateľské rozhrania aplikácií JavaFX bez kódovania. Výsledkom je súbor FXML, ktorý je možné skombinovať s projektom Java tak, že sa používateľské rozhranie prepojí s logikou aplikácie.[2]

3.3 SQL Lite

Databázová platforma SQL. SQLite je najpoužívanejší databázový stroj na svete. SQLite je zabudovaný do všetkých mobilných telefónov a väčšiny počítačov a je dodávaný v nespočetných ďalších aplikáciách, ktoré ľudia používajú každý deň.[3]

3.4 Firebase

Firebase je služba typu Backend-as-a-Service (BaaS). Firebase Storage poskytuje jednoduchý spôsob ukladania binárnych súborov – v našom prípade mp3 súbory a obrázky - do služby Google Cloud Storage priamo z klienta.[4]

3.5 CSS

Kaskádové šablóny štýlov (CSS) je jednoduchý mechanizmus na pridávanie štýlov (napr. Písma, farby, medzery) do webových dokumentov. CSS je taktiež podporované v JavaFX s menšou úpravou syntaxu.[5]

4 JavaFX

JavaFX je softvérová platforma na vytváranie a poskytovanie desktopových aplikácií, ako aj bohaté internetové aplikácie, ktoré môžu bežať na širokom spektre zariadení. JavaFX má podporu pre stolové počítače a webové prehliadače v systémoch Microsoft Windows, Linux a MacOS. JavaFX je vyvíjaná spoločnosťou Oracle, prvá verzia bola vydaná 4.12.2008 a je postavená na architektúre Model-View-Controller. JavaFX patrí medzi najpopulárnejšie softvérové platformy, čo sa týka vývoja aplikácií.[6]

Výhody JavaFX :

- Výkon platformy
- Virtualizácia
- Overená technológia

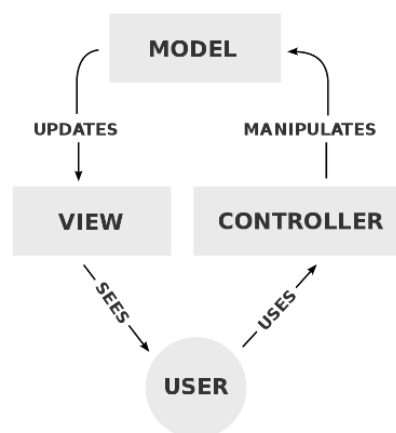


Obr. 1 JavaFX (https://en.wikipedia.org/wiki/JavaFX#/media/File:JavaFX_Logo.png)

4.1 Model-View-Controller (MVC)

Model-View-Controller je vzor návrhu softvéru bežne používaný na vývoj užívateľských rozhraní, ktorý rozdeľuje súvisiacu logiku programu na tri vzájomne prepojené prvky. Toto sa vykonáva na oddelenie vnútorných zobrazení informácií od spôsobu, akým sú informácie predkladané a prijímané používateľom. Tento druh vzoru sa používa na navrhovanie rozloženia stránky. Tento model, ktorý sa tradične používa pre grafické používateľské rozhrania (GUI) pre stolné počítače.[7]

- **Model** - Centrálna súčasť vzoru. Je to dynamická dátová štruktúra aplikácie, nezávislá od používateľského rozhrania. Priamo riadi údaje, logiku a pravidlá aplikácie.
- **View** - Akákoľvek reprezentácia informácií, ako je graf, diagram alebo tabuľka. Je možné viacero pohľadov na rovnaké informácie, ako napríklad stĺpcový graf pre správu a tabuľkové zobrazenie pre účtovníkov.
- **Controller** - Prijíma vstup a prevádza ho na príkazy pre model alebo zobrazenie.



Obr. 2 Diagram of inter 1 (

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller#/media/File:MVC-Process.svg>)

5 Použité knižnice

V našom projekte sme použili dve knižnice. Prvá slúži na prepojenie SQL lite databázy s Java kódom. Druhá knižnica obsahuje grafické elementy, ktoré sme použili v našej aplikácii.

5.1 SQLite-JDBC

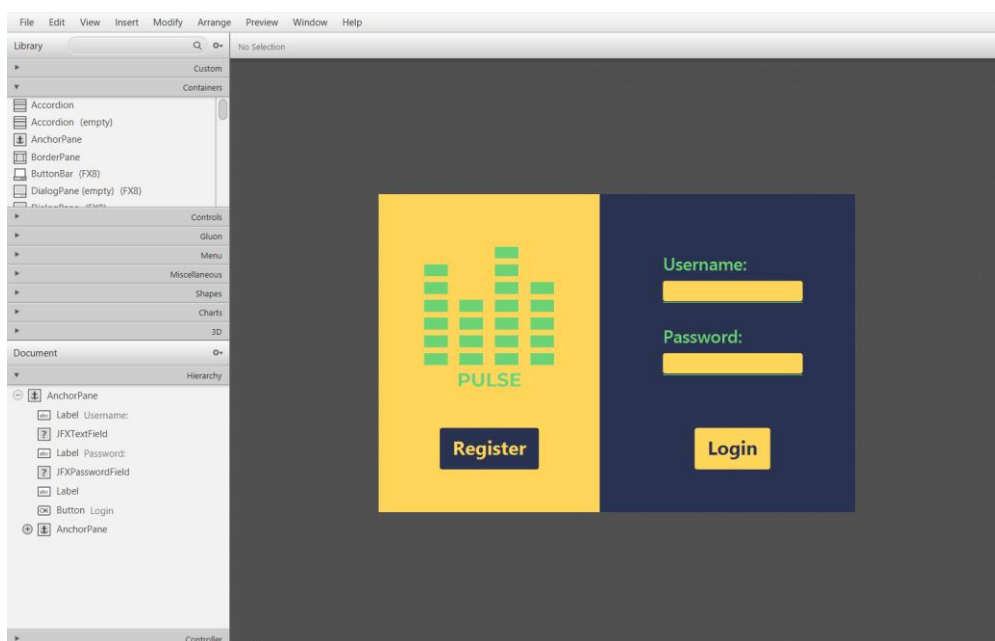
SQLite JDBC, vyvinutý Tarom L. Saitom, je knižnica na prístup a vytváranie databázových súborov SQLite v Jave.[\[8\]](#)

5.2 JFoenix

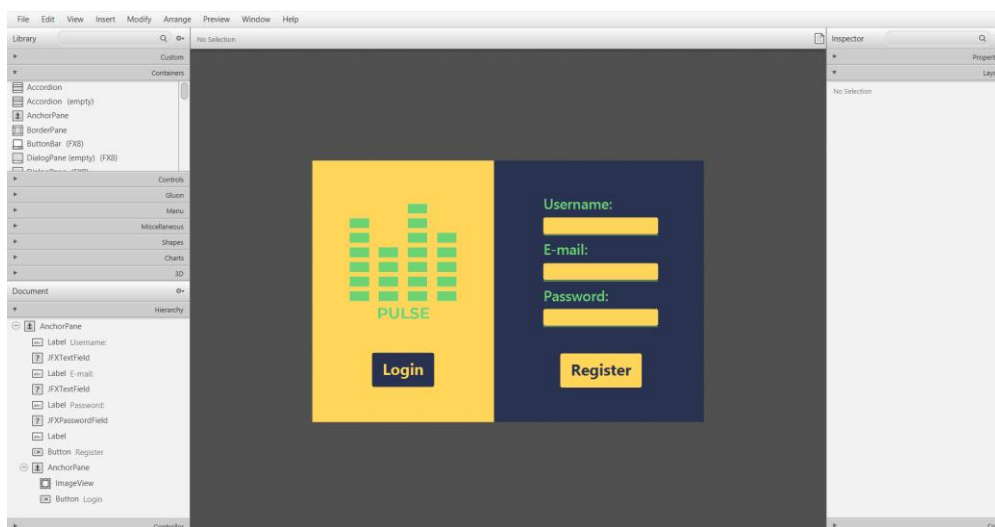
JFoenix je open source Java knižnica, ktorá implementuje Google Material Design pomocou komponentov programovacieho jazyku Java.[\[9\]](#)

6 Návrh grafického dizajnu

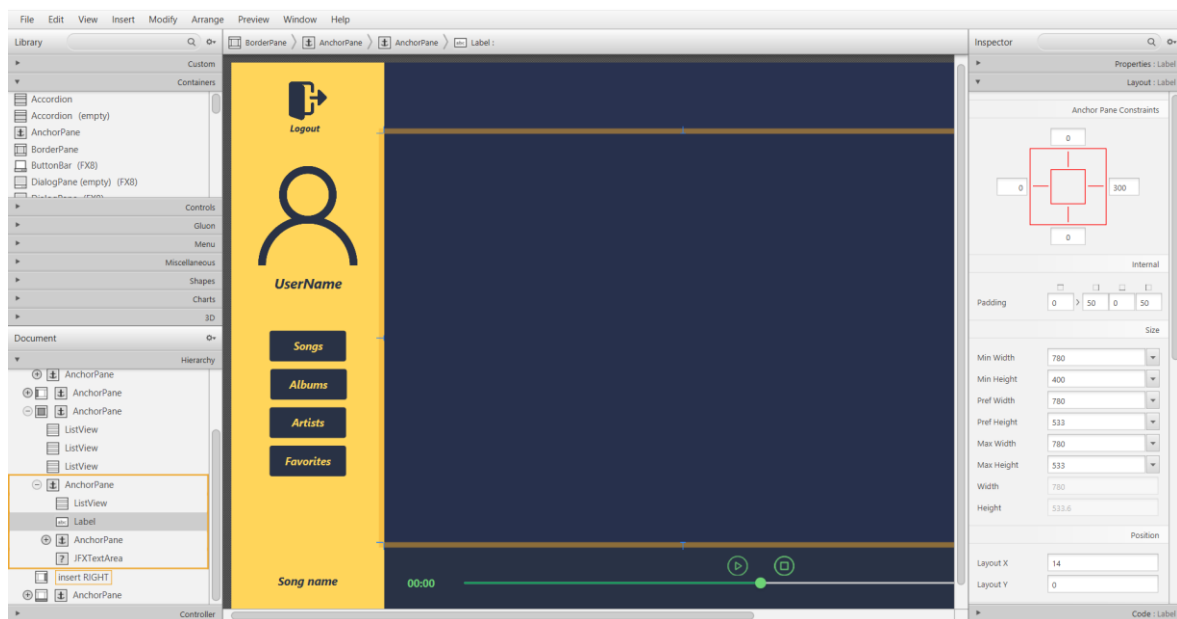
Pri vytváraní grafického dizajnu sme si naše predstavy a návrhy načrtli na papier a hľadali sme správnu kombináciu farieb. Následne si vybrali rozloženie jednotlivých elementov a farebnú schému. V ďalšom kroku sme náš návrh implementovali do súboru FXML, pomocou programu SceneBuilder. Všetky ikony použité v našej aplikácii boli použité z webovej stránky : <https://www.flaticon.com/>.



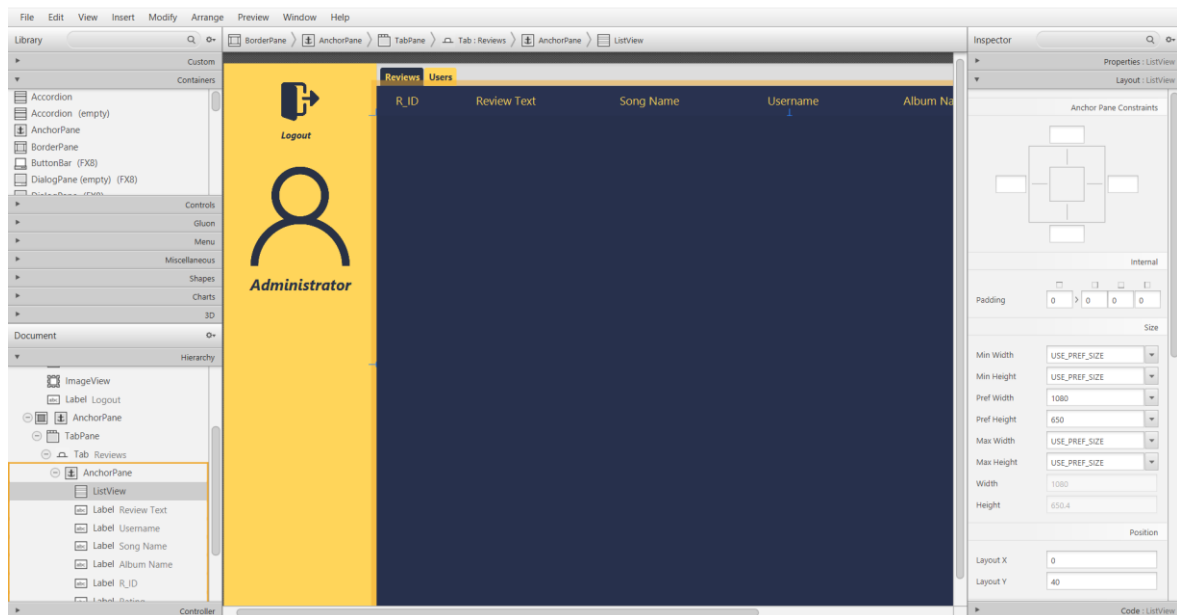
Obr. 3 Implementácia grafického návrhu 1 (Marcin, 2020)



Obr. 4 Implementácia grafického návrhu 2 (Marcin, 2020)



Obr.5 Implementácia grafického návrhu 3 (Marcin,2020)



Obr. 6 Implementácia grafického návrhu 4 (Marcin,2020)

7 Tvorba databázy

V projekte sme využili databázu na ukladanie dôležitých údajov o užívateľoch, interpretoch, albumoch, skladbách.

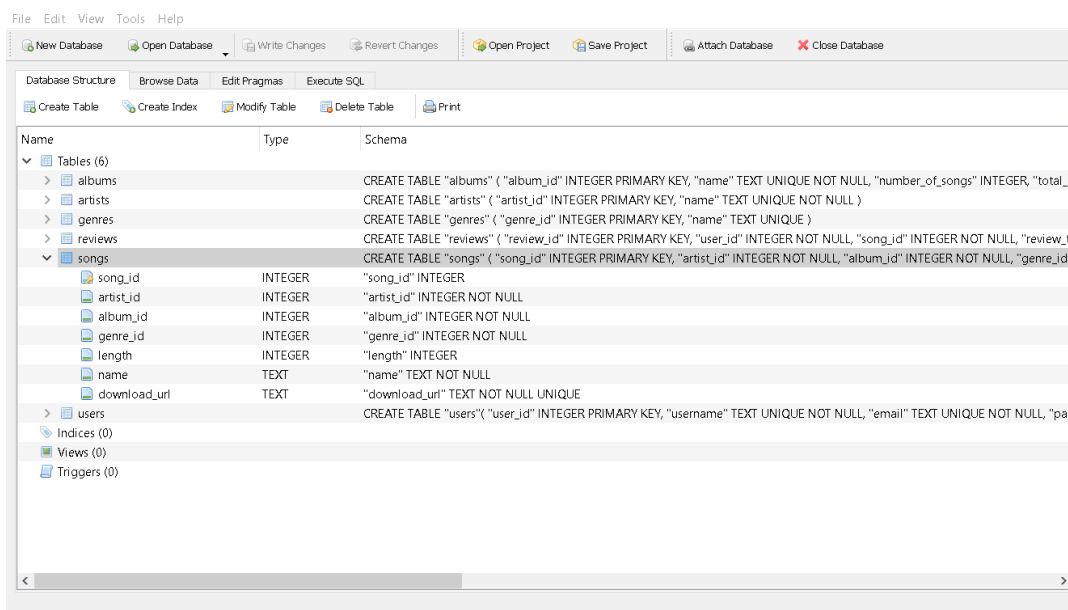
7.1 Návrh databázy

Rozhodli sme sa ako by mali tabuľky našej databázy vyzerieť, aké vzťahy medzi jednotlivými tabuľkami nastaviť.

7.2 Tvorba tabuliek databázy

Celú databázu a tabuľky sme vytvorili pomocou programu DB Browser (SQL lite). Pre ukážku tabuľku Songs sme vytvorili pomocou príkazu:

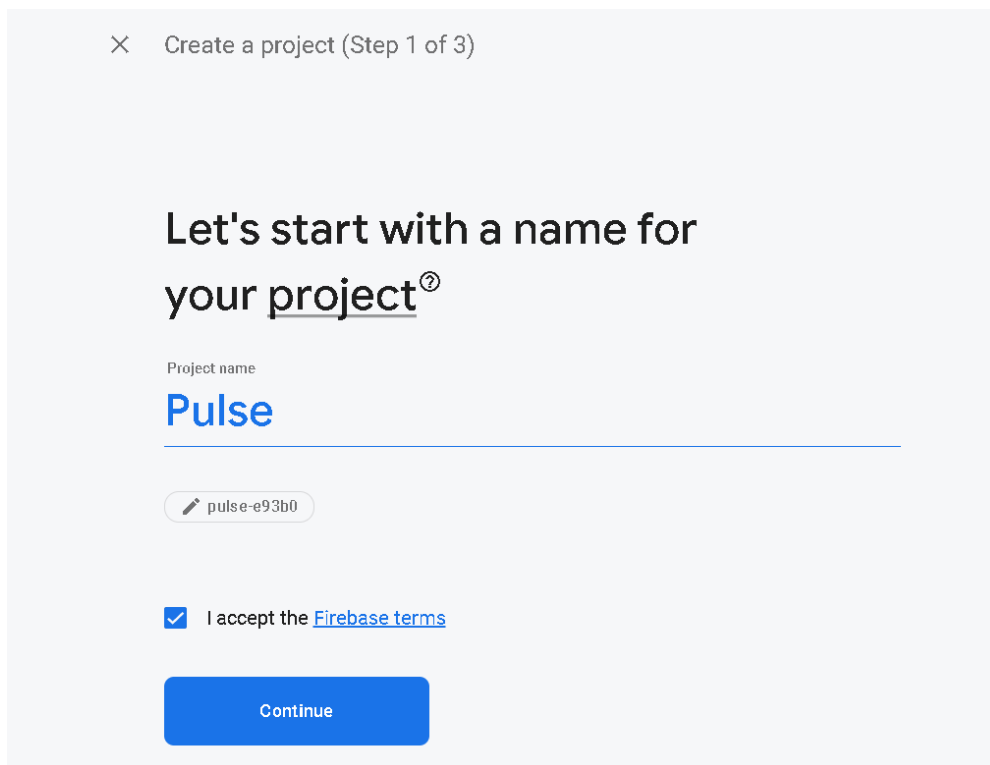
```
CREATE TABLE "songs" (  
    "song_id" INTEGER PRIMARY KEY,  
    "artist_id" INTEGER NOT NULL,  
    "album_id" INTEGER NOT NULL,  
    "genre_id" INTEGER NOT NULL,  
    "length" INTEGER,  
    "name" TEXT NOT NULL,  
    "download_url" TEXT NOT NULL UNIQUE,  
    FOREIGN KEY(artist_id) REFERENCES artists(artist_id),  
    FOREIGN KEY(album_id) REFERENCES albums(album_id),  
    FOREIGN KEY(genre_id) REFERENCES genres(genre_id))
```



Obr. 7 Vytvorenie databázy (Marcin,2020)

7.3 Vytvorenie úložiska – FireBase

Po registrácii na platforme Firebase : <https://firebase.google.com/> sme si vytvorili projekt s názvom Pulse.



× Create a project (Step 1 of 3)

Let's start with a name for your project[?]

Project name

Pulse

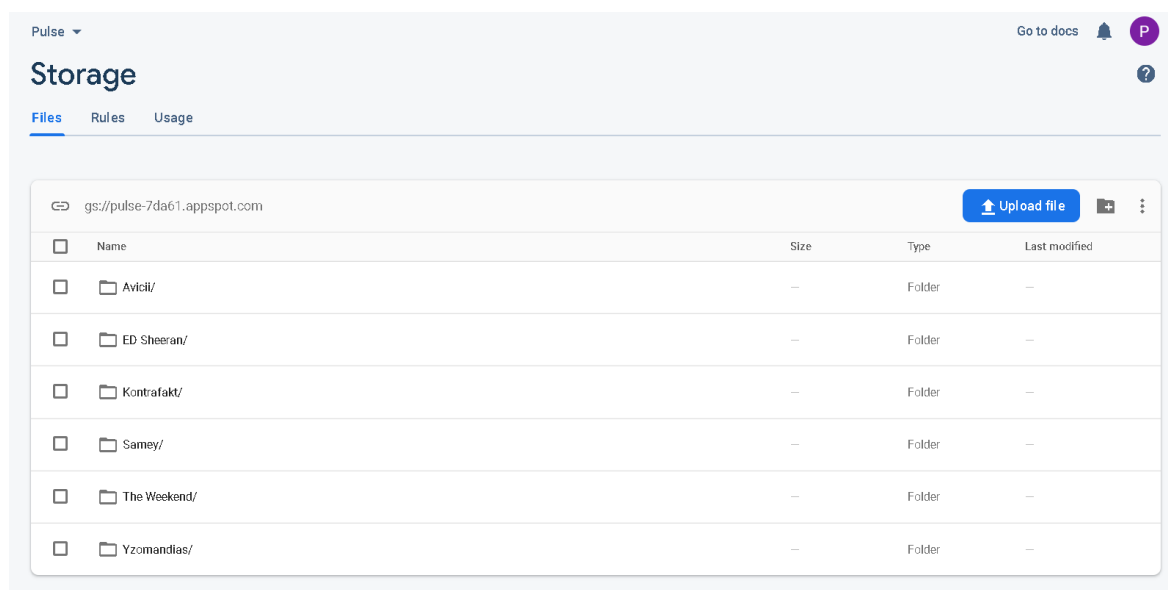
pulse-e93b0

☒ I accept the [Firebase terms](#)

Continue

Obr. 8 Tvorba FireBase projektu 1 (Marcin,2020)

Po úspešnom vytvorení projektu nasledovalo vytvorenie úložiska s dohodnutou štruktúrou jednotlivých adresárov a následne naplnenie tohto úložiska. Štruktúra úložiska funguje nasledovne: *INTERPRET* → *ALBUM INTERPRETA* → *SKLADBY ALBUMU*.



Obr. 9 Tvorba FireBase projektu 1 (Marcin,2020)

gs://pulse-7da61.appspot.com > Yzomandias				Upload file		
<input type="checkbox"/>	Name	Size	Type	Last modified		
<input type="checkbox"/>	📁 Sbohem Roxano/	—	Folder	—		

Obr. 10 Tvorba FireBase projektu 2 (Marcin,2020)

gs://pulse-7da61.appspot.com > Yzomandias > Sbohem Roxan...				Upload file		
<input type="checkbox"/>	Name	Size	Type	Last modified		
<input type="checkbox"/>	📁 Andez.mp3	3.51 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 Halo ft. Karlo.mp3	8.09 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 KML ft. Milion+.mp3	10.2 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 Kokalero.mp3	6.68 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 Kordylery ft. Decky, Zmrd.mp3	6.28 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 Nevzláajte.mp3	6.56 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 Opozice.mp3	7.04 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 Oujeje ft. Jickson, Nik Tendo, Zm.mp3	9.12 MB	audio/mp3	Jan 14, 2020		
<input type="checkbox"/>	📁 Pod' blíž ft. Čis T, Smack.mp3	8.62 MB	audio/mp3	Jan 14, 2020		

Obr. 11 Tvorba FireBase projektu 3 (Marcin,2020)

7.4 Použitie úložiska v databáze

Každá skladba v našej databáze vlastní URL pomocou, ktorej sa vieme dostať ku konkrétnej skladbe v úložisku a následne ju prehrať v našom hudobnom prehrávači.

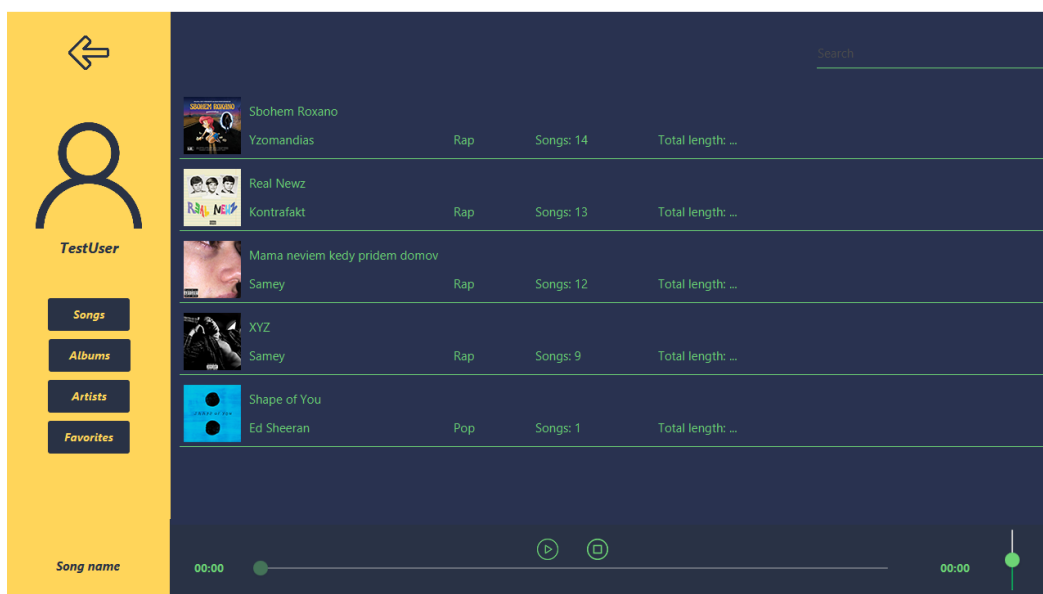
name	download_url
Filter	Filter
Andez	https://firebasestorage.googleapis.com/v0/b/p...
Halo	https://firebasestorage.googleapis.com/v0/b/p...
KML	https://firebasestorage.googleapis.com/v0/b/p...
Kokalero	https://firebasestorage.googleapis.com/v0/b/p...
Kordylery	https://firebasestorage.googleapis.com/v0/b/p...
Nevzláajte	https://firebasestorage.googleapis.com/v0/b/p...
Opozice	https://firebasestorage.googleapis.com/v0/b/p...
Oujeje	https://firebasestorage.googleapis.com/v0/b/p...
Pod' blíž	https://firebasestorage.googleapis.com/v0/b/p...
Rehab	https://firebasestorage.googleapis.com/v0/b/p...
Spolu	https://firebasestorage.googleapis.com/v0/b/p...
TTBB	https://firebasestorage.googleapis.com/v0/b/p...
Ukážu ti jak	https://firebasestorage.googleapis.com/v0/b/p...
WWW	https://firebasestorage.googleapis.com/v0/b/p...

Obr. 12 Tvorba FireBase projektu 3 (Marcin,2020)

8 Funkčnosť aplikácie

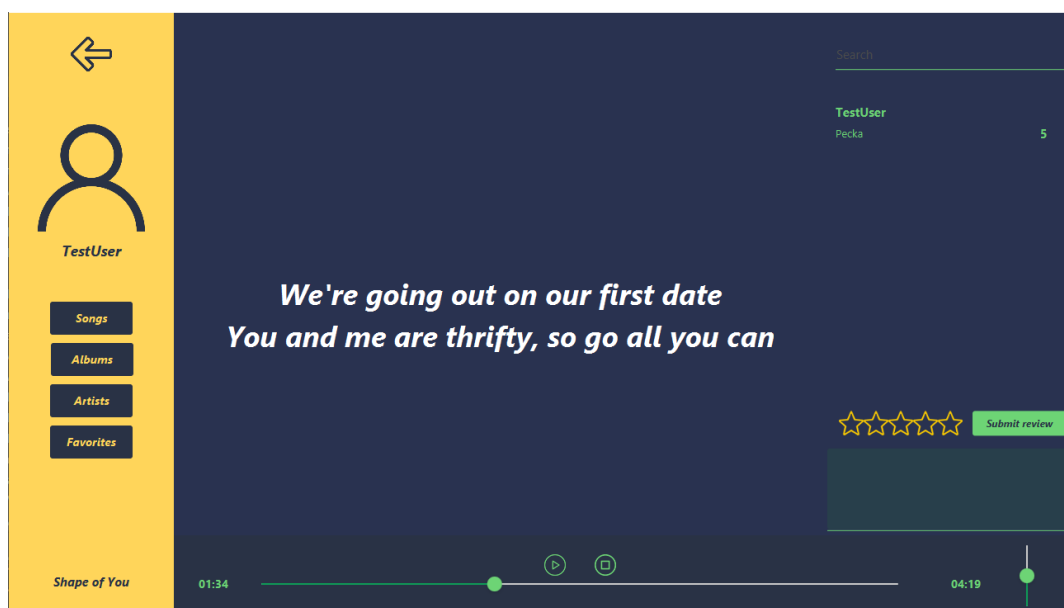
8.1 Hudobný prehrávač

Po prihlásení užívateľovi zobrazí hudobný prehrávač, kde užívateľ vidí všetky dostupné albumy. V hornej časti sa nachádza vyhľadávací panel, ktorý užívateľovi uľahčuje vyhľadávanie albumov, skladieb či interpretov. V ľavej časti sa nachádza prehľadné menu. Jednotlivé položky reprezentujú základne elementy našej aplikácie.



Obr. 13 Hlavná scéna prehrávača (Marcin, 2020)

Po spustení konkrétnej skladby sa užívateľovi zobrazujú informácie o skladbe. Taktiež sa priebežne zobrazuje text skladby ak taký text existuje. V pravej časti prehrávača sa nachádza časť, ktorá slúži na recenzie.



Obr. 14 Prehrávanie skladby (Marcin, 2020)

8.2 Live vyhľadavanie

V našom prehrávači sme použili už vytvorený kód určený na vyhľadavanie v TableView: https://stackoverflow.com/questions/44317837/create-search-textfield-field-to-search-in-a-javafxtableview?fbclid=IwAR2xWr0_q1prWutTihdGh7tYBokKhYCGznyfr-4yQOOFrnGvIcxdhSscdXc, ktorý sme upravili aby bol použiteľný v našom prehrávači.

```
searchField.setOnKeyPressed(e -> {
    songsListView.toFront();
    searchField.textProperty().addListener((observableValue, oldValue, newValue) -> {
        String LC = newValue.toLowerCase();
        // if one of these conditions is true we want to show song
        filteredData.setPredicate((Predicate<? super Song>) song->{
            if(newValue == null || newValue.isEmpty())
                return true;
            if(null != song.getName() && song.getName().toLowerCase().contains(LC))
                return true;
            if(null != song.getAlbumName() && song.getAlbumName().toLowerCase().contains(LC))
                return true;
            if(null != song.getArtistName() && song.getArtistName().toLowerCase().contains(LC))
                return true;
            if(null != song.getGenreName() && song.getGenreName().toLowerCase().contains(LC))
                return true;
            return false;
        });
    });
});
```

Obr. 15 Vyhľadavanie v prehrávači (Marcin,2020)

8.3 Priebežne zobrazovanie textu skladby

Najdôležitejšou funkciou prehrávača je priebežné zobrazovanie textu skladby. Po načítaní textu z úložiska FireBase vytvoríme TreeMap, ktorý obsahuje text ako hodnotu a čas prepočítaný na milisekundy ako kľúč. Následne sa vlákno každú milisekundu dopytuje na daný TreeMap, či existuje pár pre daný čas skladby prepočítaný na milisekundy. Daný text sa zobrazuje užívateľovi.

Formát textu: 00:00:00,000 → 00:00:00,000



Riadky textu



Prázdny riadok

```

private void startLyrics(){
    if (actualSong.getLyricsUrl().isEmpty()){
        lyricsText.setText("No lyrics for this song!");
        return;
    }

    URL url;
    URLConnection connection;
    BufferedReader bufferedReader;
    String line;

    try {
        // get lyrics from firebase storage
        url = new URL(actualSong.getLyricsUrl());
        connection = url.openConnection();
        bufferedReader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
        lyrics = new TreeMap<>();

        // create map with lyrics for every millisecond
        // lyrics format
        // 00:00:00,000 --> 00:00:00,000
        // Few rows of text
        // Empty row
        while ((line = bufferedReader.readLine()) != null){
            if (line.contains("-->")){
                int from = ConvertToMillis(line.split( regex: "-->")[0]);
                int to = ConvertToMillis(line.split( regex: "-->")[1]);

                // get text after time
                String text = "";
                while (!(line = bufferedReader.readLine()).isEmpty()){
                    text = text + line + "\n";
                }

                // insert same text for every millisecond during show time
                // for case when user change playing time of song
                for (int i = from; i < to; i++)
                    lyrics.put(i, text);
            }
        }
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Obr. 16 Priebežné zobrazovanie textu 1(Marcin,2020)

```

// crate thread that will run every millisecond and show lyrics
Thread thread = new Thread() -> {
    while (true){
        if (!playing)
            return;

        try {
            Thread.sleep( millis: 1);
            // show lyrics if we have some for current time
            if (lyrics.containsKey((int) mediaPlayer.getCurrentTime().toMillis()))
                Platform.runLater(() -> lyricsText.setText(lyrics.get((int) mediaPlayer.getCurrentTime().toMillis())));
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (NullPointerException e) {
            playing = false;
        }
    }
});

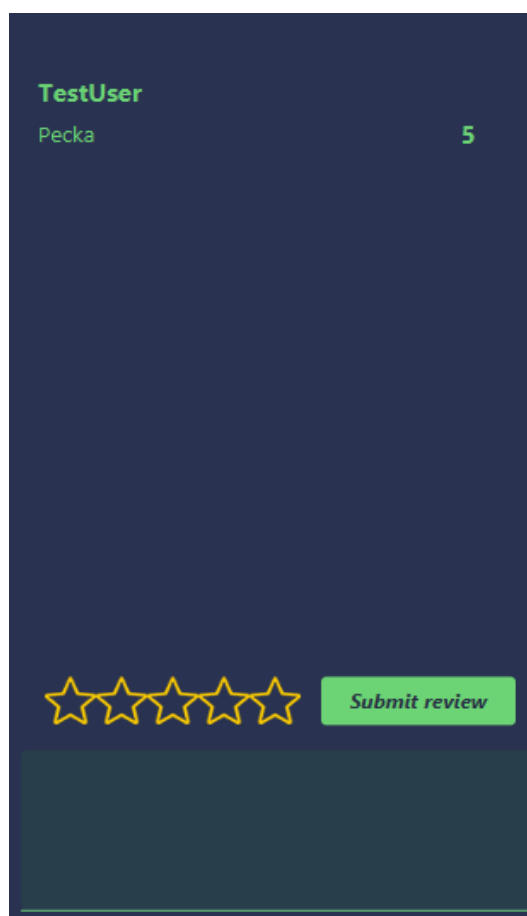
thread.start();

```

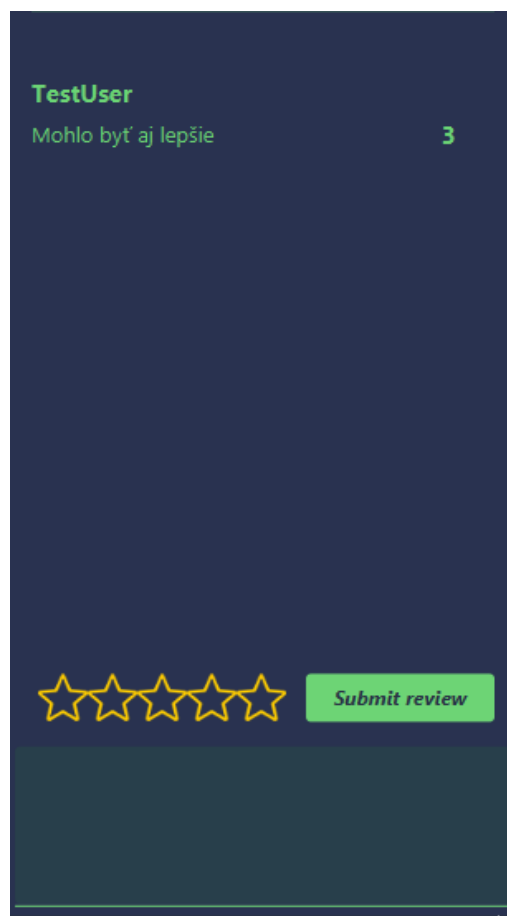
Obr. 17 Priebežné zobrazovanie textu 2(Marcin,2020)

8.4 Tvorba recenzie

Užívateľovi je umožnené ohodnotiť skladby. Môže tak urobiť slovne alebo číselne pomocou počtu hviezd. Každú recenziu vidí administrátor, ktorý na základe nevhodnej recenzie môže užívateľovi zakázať prístup do aplikácie.



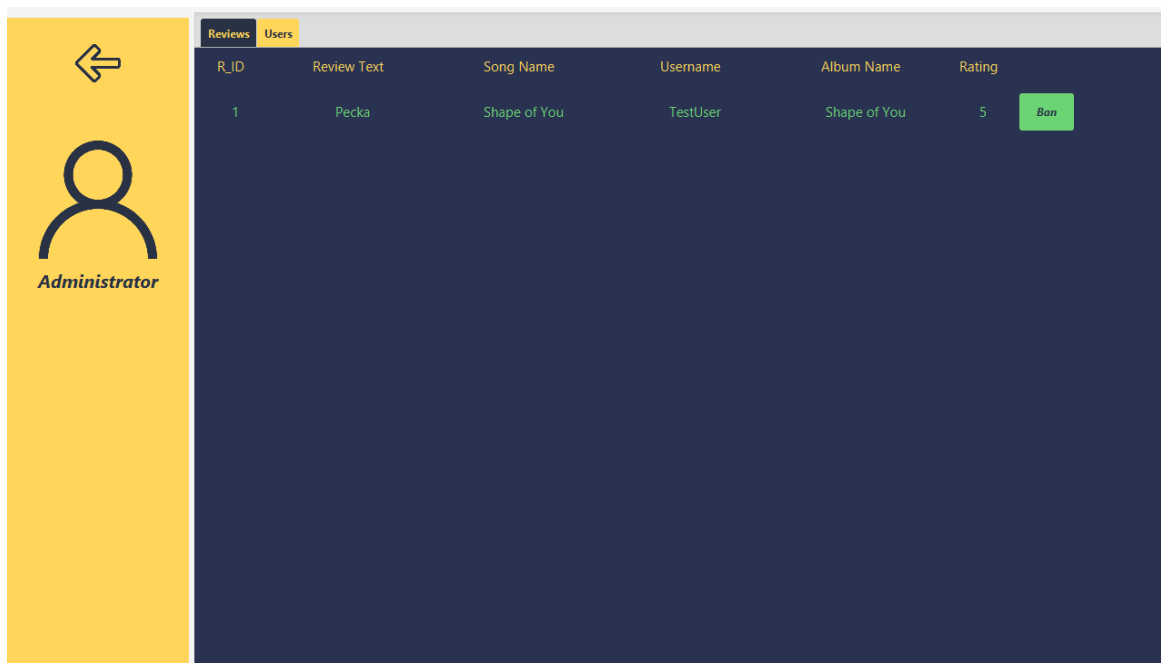
Obr. 18 Recenzie (Marcin, 2020)



Obr. 19 Recenzie 2 (Marcin,2020)

8.5 Administrátorské rozhranie

Administrátorské rozhranie slúži na kontrolu recenzií, zakazovanie a povoľovanie prístupu užívateľom do aplikácie.



Obr. 20 Administrátor-recenzie (Marcin, 2020)



Obr.21 Administrátor-užívateľa (Marcin,2020)

9 Výsledky práce a diskusia

Vytvorili sme atraktívny hudobný prehrávač, ktorý dokáže konkurovať aj komerčným hudobným prehrávačom. Obsahuje atraktívne funkcionality ako priebežné zobrazovanie textu, hodnotenie skladieb. Užívateľ si vie vyberať medzi uloženými interpretmi, albumami a skladbami. Na otestovanie aplikácie boli vytvorení skúšobní užívatelia, ktorí testovali aplikáciu.

10 Závery práce

Vytvorili sme aplikáciu, ktorá umožňuje prehrávať skladby, zobrazovať text skladby a hodnotiť jednotlivé skladby. Všetky skladby, albumy a interpretov , ktorí sú uložení na našom úložisku má užívateľ sprístupnených na jednom mieste. Všetky stanovené ciele aplikácie boli splnené.

Zhrnutie

Ako hlavný cieľ práce sme si stanovili vytvoriť prehrávač, ktorý bude konkurencie schopný. Práca začala návrhom prehrávača .Pokračovali sme návrhom databázy, grafickým návrhom jednotlivých scén a konkrétnych funkcionalít. Databáza bola prepojená s projektom. Nasledovalo aplikovanie funkcionalít do projektu a vytvorenie kompletnej aplikácie. Aplikácia, ktorú sme vytvorili spĺňa kritéria, ktoré boli stanovené v úvode projektu.

Resumé

Work began with the design of the player. Our main goal was to create a player that is competitive. We continued with database design, graphic design of individual scenes and specific functionalities. The database was linked to the project. This was followed by the application of functionalities to the project and the creation of a complete application. The application we created meets the criteria that were set in the project introduction.

Použitá literatúra

[1] Java : odsek 1, odsek 2

[https://sk.wikipedia.org/wiki/Java_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk))

[2] SceneBuilder : odsek 1

<https://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html>

[3] SQL lite: odsek 1

<https://www.sqlite.org/index.html>

[4] Chris Esplin What is FireBase? [online] Publikované 24.8.2016 [citované 5.4.2020] Dostupné z

<https://howtofirebase.com/what-is-firebase-fcb8614ba442>

[5] CSS : odsek 1

<https://www.w3.org/Style/CSS/Overview.en.html>

[6] JavaFX : odsek 1

<https://en.wikipedia.org/wiki/JavaFX>

[7] Tiffany Peña Model-View-Controller (MVC) Architecture & Java/JavaFX [online] Publikované 10.5.2019 [citované 5.4.2020] Dostupné z

<https://medium.com/@tiff.sage/model-view-controller-mvc-architecture-java-javafx-8540a575b523>

[8] SQL lite JDBC : Readme.md odsek 1

<https://github.com/xerial/sqlite-jdbc>

[9] JFoenix :odsek 1

<http://www.jfoenix.com/documentation.html>