Středoškolská odborná činnost

Obor SOČ: 18. Informatika

Baby-friendly Map

Stanislav Kinzl

Jakub Kortus

kraj Královéhradecký

Náchod 2017

Středoškolská odborná činnost

Obor SOČ: 18. Informatika

Baby-friendly Map

Autoři:	Stanislav Kinzl
	Jakub Kortus
Škola:	Jiráskovo gymnázium Náchod
Kraj:	Královéhradecký
Konzultant:	RNDr. Jan Preclík, Ph.D.

Náchod 2017

Prohlášení

Prohlašujeme, že jsme svou práci SOČ vypracovali samostatně a použili jsme pouze podklady (literaturu, projekty, SW atd.) uvedené v seznamu vloženém v práci SOČ.

Prohlašujeme, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemáme závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Náchodě

Poděkování

Tímto vyjadřujeme své upřímné poděkování panu profesorovi Janu Preclíkovi za odbornou pomoc při vypracování této práce, slečně Lence Pošepné za zastřešení a investice do projektu, Jindře Nývltové za jazykovou korekturu práce. Děkujeme také všem, co se na projektu jakkoliv podíleli.

Anotace

Naším cílem bylo vytvořit aplikaci pro mobilní zařízení se systémem Android (chytré telefony a tablety), která pomocí intuitivní technologie Google Maps pomůže rodinám s dětmi vyhledat ve svém okolí zařízení, jež jsou „baby-friendly“ – „přátelská k dětem“

Jedná se o různé podniky (hotely, restaurace…), mezi jejichž základní vybavení patří dětský koutek, přebalovací pult, dětská postýlka, bezbariérový přístup, káva bez kofeinu atd., tedy významná usnadnění pro rodiče, kteří taková zařízení aktivně, nejen na dovolených, využívají.

Aplikace získala stabilní a použitelnou podobu s příjemným uživatelským prostředím a stabilní databází. Přestože je již volně ke stažení ve své finální podobě, je stále doplňována novými užitečnými vylepšeními.

Klíčová slova: aplikace, Android, Google Maps, baby-friendly

Annotation

Our goal was to develop an application for mobile devices with operating system Android (smartphones and tablets), which would help families with children to search for institutions in their surroundings, which are „baby-friendly“. This all is supported by intuitive technology called Google Maps.

Baby-friendly institutions are different businesses (hotels, restaurants…), parameters of which include: play area in restaurants, nappy changing unit, cot, coffee without caffeine, wheelchair and so on. Thus, it is a real help for parents, who actively look for such institutions.

The Application was developed into stable and user friendly form with pleasant user interface and stable database. Although it is available at Google Play, it is still being updated.

Keywords: application, Android, Google Maps, baby-friendly

Obsah

Úvod

Inspirací pro aplikaci Baby-friendly Map, později i tuto práci, byl nápad, který se zrodil ze skutečné potřeby jedné slečny, která cestovala s malým dítětem. Občas potřebovala najít nějaké zařízení, které nabízí výše zmíněné služby. Google Maps tuto funkci nemají, proto se rozhodla vyhledat někoho, kdo by byl schopný a ochotný potřebnou „kapesní mapu“ vytvořit.

Nejprve oslovila mě, Stanislava Kinzla, coby designéra a programátora, a já později uvedl do projektu Jakuba Kortuse jako grafika. Hned na začátku jsme se rozhodli, že finální produkt musí splňovat požadavek intuitivnosti, kvality a dostupnosti. Intuitivnost znamená, že když rodič poprvé otevře aplikaci, hned by mu mělo být jasné, jak postupovat. Kvalita zastřešuje funkčnost a dobře odvedenou práci. Dostupností míníme co nejmenší nároky kladené na uživatele, co se týká přístupu k aplikaci a jejího aktivního využívání.

2 Teoretický úvod

2.1 Mobilní aplikace

Termín **aplikace** skrývá spoustu významů. V této práci jí ale přiřazujeme význam poslední dobou širokou veřejností nejčastěji používaný, tedy **mobilní aplikace**.

„Mobilní aplikace (mobile app) je softwarová aplikace vytvořená speciálně pro **chytré telefony** (smartphones), **tablety**; a další mobilní zařízení. Tvůrci takových aplikací se obvykle snaží co nejvíce využít možností intuitivního uživatelského rozhraní a dotykového ovládání, které mobilní zařízení nabízí.“ [1]

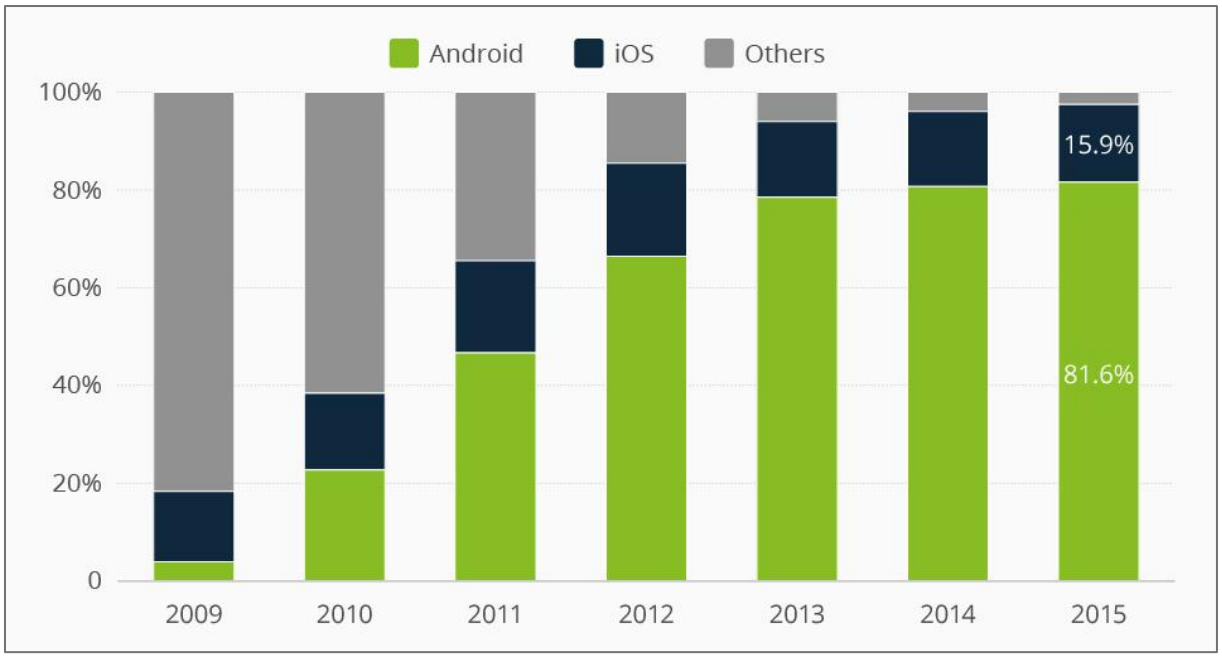
Protože aplikace hýbou současným světem moderních technologií a ovlivňují tak naše každodenní činnosti, od jednoduché navigace přes navazování sociálních vztahů až k přístupu k takřka neomezenému množství informací stiskem tlačítka, vznikl enormní trh se skoro nekonečným potenciálem. Trh se rozděluje na hned několik platforem/marketů, které často závisí na operačním systému daného mobilního zařízení. Android má **Google Play**, iOSu od společnosti Apple náleží App Store, Windows Phone má Microsoft Store a tak dále.

Důsledkem výše uvedeného je vznik nové profese s názvem vývojář mobilních aplikací. Dobří vývojáři se nehledají snadno, jelikož způsob vývoje a aktuální praktiky se každou chvílí mění a přizpůsobují trendu. To, co bylo normou před třemi lety, je v současném světě mobilních aplikací považováno za zastaralé.

2.2 Android

V rámci tohoto projektu je největší pozornost kladena na operační systém **Android**, pro nějž byla naše aplikace primárně vytvořena. Nevylučujeme tím ale možnost, že bude v budoucnu přizpůsobena jiným operačním systémům, jako je například iOS.

Android je mobilní operační systém založený společností **Google Inc.** Cílem Androidu je snažit se o progresivní rozvoj mobilních technologií, které budou mít zásadně nižší náklady jak na vývoj, tak na distribuci. Jádro Androidu, založené na jádru **Linuxu**, bylo navrženo tak, aby běželo na různém hardwaru.



Obrázek 1: Graf podílu mobilních operačních systému na trhu (na základě prodaných kusů). Zelená znázorňuje Android, tmavě modrá iOS a šedá ostatní OS. [online]. [cit. 2017-01-21]. Dostupné z: https://www.statista.com

Z grafu můžeme vyčíst, že se Androidu v posledních šesti letech povedlo vytlačit z trhu jakékoliv ostatní operační systém vyjma iOS, jehož uživatelská základna zůstává loajální.

Android je tedy jeden z nejrychleji rostoucích OS, což dokazuje i rychlost, s jakou vycházejí nové verze tohoto systému (aktuální Nougat 7.0 – 7.1.1).

Rozmach Androidu se dá vysvětlit svojí přístupností pro nezávislé vývojáře, dostupností a jednoduchostí. Používá ho pro svá mobilní zařízení drtivá většina výrobců (**Samsung**, **Sony**, **Huawei**, **LG** a k roku 2017 i **Microsoft**).

2.3 Struktura mobilní aplikace

2.3.1 Material Design

Material Design je současný trend udávaný Googlem. Je to vizuální „styl“ pro Android, Chrome OS a webové aplikace, který klade důraz na **přehlednost a přístupnost** uživateli.

„Mezi cíle Material Designu patří:

- Sestavit vizuální jazyk, který syntetizuje klasické principy dobrého designu s inovací a možností propojit technologie a vědu.
- Vytvořit jednotný systém umožňující zformátovat stejné postupy napříč platformami, bez ohledu na typ zařízení.
- Důraz na důležitost vstupních metod: dotyk, hlas, myš a klávesnice.“[2]

2.3.2 Ikona aplikace

Že by aplikace měly být po vzhledové stránce atraktivní, je všem vývojářům bezpochyb zřejmé. Pokud chtějí, aby se jejich aplikace umístily v top žebříčku Google Play, měly by mít „hezký kabátek“, který uživatelé ocení. Ještě předtím, než se uživatel rozhodne aplikaci nainstalovat, uvidí její ikonu.

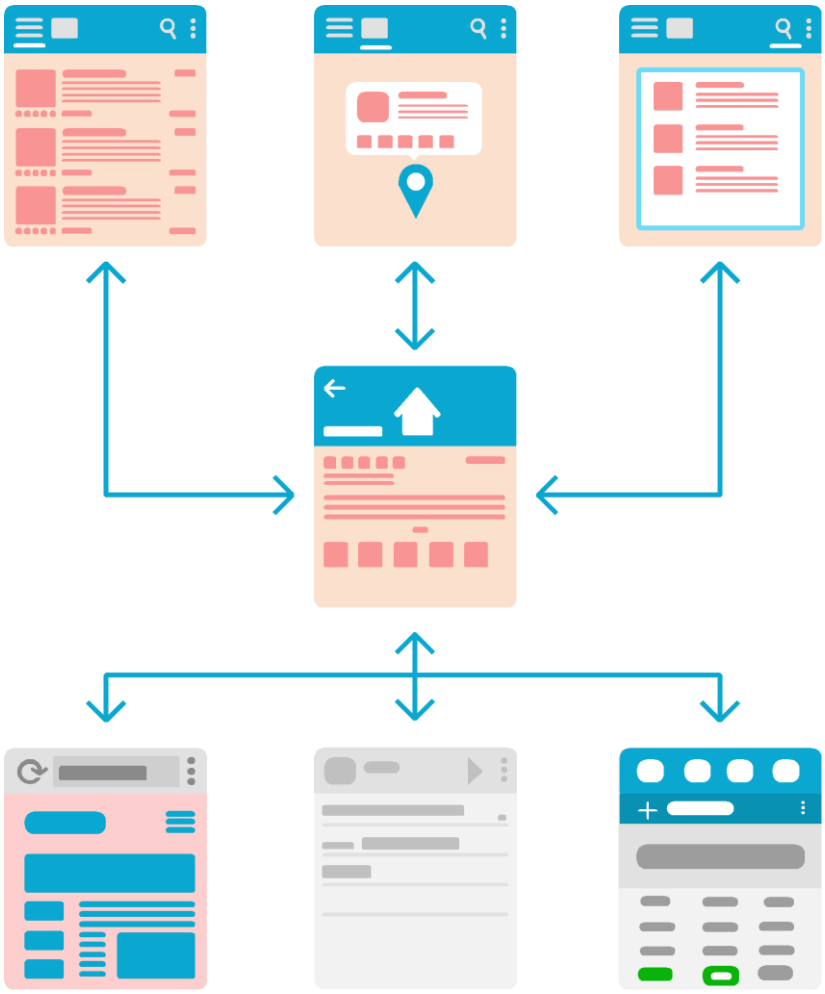
K pochopení důležitosti ikon je potřeba znát jejich účel. V první řadě tvoří jeden z nástrojů marketingu, za druhé, ikona by se měla lišit od ostatních a zároveň prezentovat klíčové prvky aplikace. Podle mého názoru se ale kreativitě meze nekladou, a proto porušení některého z designových pravidel může být pro celkový výsledek výhodou.

Obrázek 2: Ikona naší aplikace Baby-friendly Map

2.3.3 Architektura aplikace

Typická Android aplikace se skládá z vrchní úrovně (Top view) a detailních views. Pokud je hiearchie komplexní, vrchní úrovně mohou navazovat na úrovně kategorií.

Na následujícím obrázku vztahujícím se k naší aplikaci demonstrujeme, jak taková architektura vypadá.



Obrázek 3: Reprezentace architektury mobilní aplikace pro Android. Obrázek znázorňuje zjednodušenou architekturu naší aplikace Baby-friendly Map.

2.3.3.1 Top view

Top view aplikace se typicky skládá z různých views, které daná aplikace podporuje. Views jsou často různé reprezentace jedněch a těch samých dat. (Seznam videí atd.)

2.3.3.2 Detail/Edit view

Detail/Edit view je zobrazením, ve kterém uživatel konzumuje nebo vytváří data. (Například popis restaurace nebo přidání poznámky do blogu.)

2.3.3.3 Externí aplikace

Aplikace navazující na uživatelovu činnost v detailní aktivitě. Například při stisku URL webové stránky se otevře externí aplikace, v tomto případě prohlížeč, nebo například číselník, anebo emailový klient.

Při vývoji moderní aplikace se tedy musí dbát na uživatelské prostředí. To by mělo být navrženo takovým způsobem, aby byl pohyb uživatele v aplikaci naprosto přirozený a intuitivní.

2.4 Java

„Java je programovací jazyk nezávislý na platformě. Vyvíjí ho společnost SUN, Microsystems a je zdarma dostupný pro různé operační systémy (Windows, Linux, Solaris).“[3]

Mezi hlavní vlastnosti Javy patří:

- **Jednoduchost** – její syntaxe je zjednodušenou (a drobně upravenou) verzí syntaxe jazyka C a C++. Odpadá většina nízkoúrovňových konstrukcí.
- **Objektová orientovanost** – což značí programovací paradigma, které se odlišilo od původního procedurálního. Výkonný kód je v objektovém programování přidružen datům, což umožňuje snadnější přenos kódu mezi různými projekty. Propojení umožnilo zavést například takové užitečnosti, jakými jsou objekty, dědičnost a polymorfizmus.
- **Interpretovanost** – místo skutečného strojového kódu se vytváří pouze tzv. mezikód (bajtkód), který je nezávislý na architektuře počítače nebo zařízení. Program poté může pracovat na libovolném počítači, který má dostupný interpret Javy, tzv. virtuální stroj Javy – Java Virtual Machine (JVM).
- **Robustnost** – Java je určená pro psaní vysoce spolehlivého softwaru – z toho důvodu neumožňuje některé programátorské konstrukce, které bývají často příčinou chyb.
- **Bezpečnost** – má vlastnosti, které chrání počítač v síťovém prostředí, na kterém je program zpracován, před nebezpečnými operacemi nebo napadením vlastního operačního systému nepřátelským kódem.
- **Výkonnost** – přestože je to jazyk interpretovaný, není ztráta výkonu významná, neboť překladače mohou pracovat v režimu „just-in-time“ a do strojového kódu se překládá jen ten kód, který je opravdu za potřebí.
- **Elegantnost** – velice pěkně se v něm pracuje, je snadno čitelný, což je výhodou při publikaci a šíření algoritmů. Přímo vyžaduje ošetření chyb.

*HelloWorld.java

```
1
2  public class HelloWorld {
3      public static void main(String[] args) {
4          System.out.println("Hello, World");
5      }
6  }
```

Obrázek 4: Ukázka tradičního algoritmu programu napsaném v Javě, jehož výstupem je „Hello, World!“. V horní liště si můžeme všimnout, že třídy jazyka Javy se ukládají ve formátu .java

2.4.1 Použití Javy pro Android

„Společnosti Google a Android si zvolily použití Javy jako klíčový pilíř při tvorbě stejnojmenného open source mobilního operačního systému určeného pro chytré telefony a tablety. I přesto, že je Android postaven na Linuxovém jádře a byl napsán převážně v programovacím jazyce C, tak SDK Androidu používá jazyk Java jako základ pro svoje aplikace. Java je však používána pouze pro syntaxi, nikoliv pro svoji knihovnu tříd.“[4]

```
package com.example.kinzl.socukazka;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {


    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
    }
}
```

Ukázka fundamentální prázdné Aktivity.

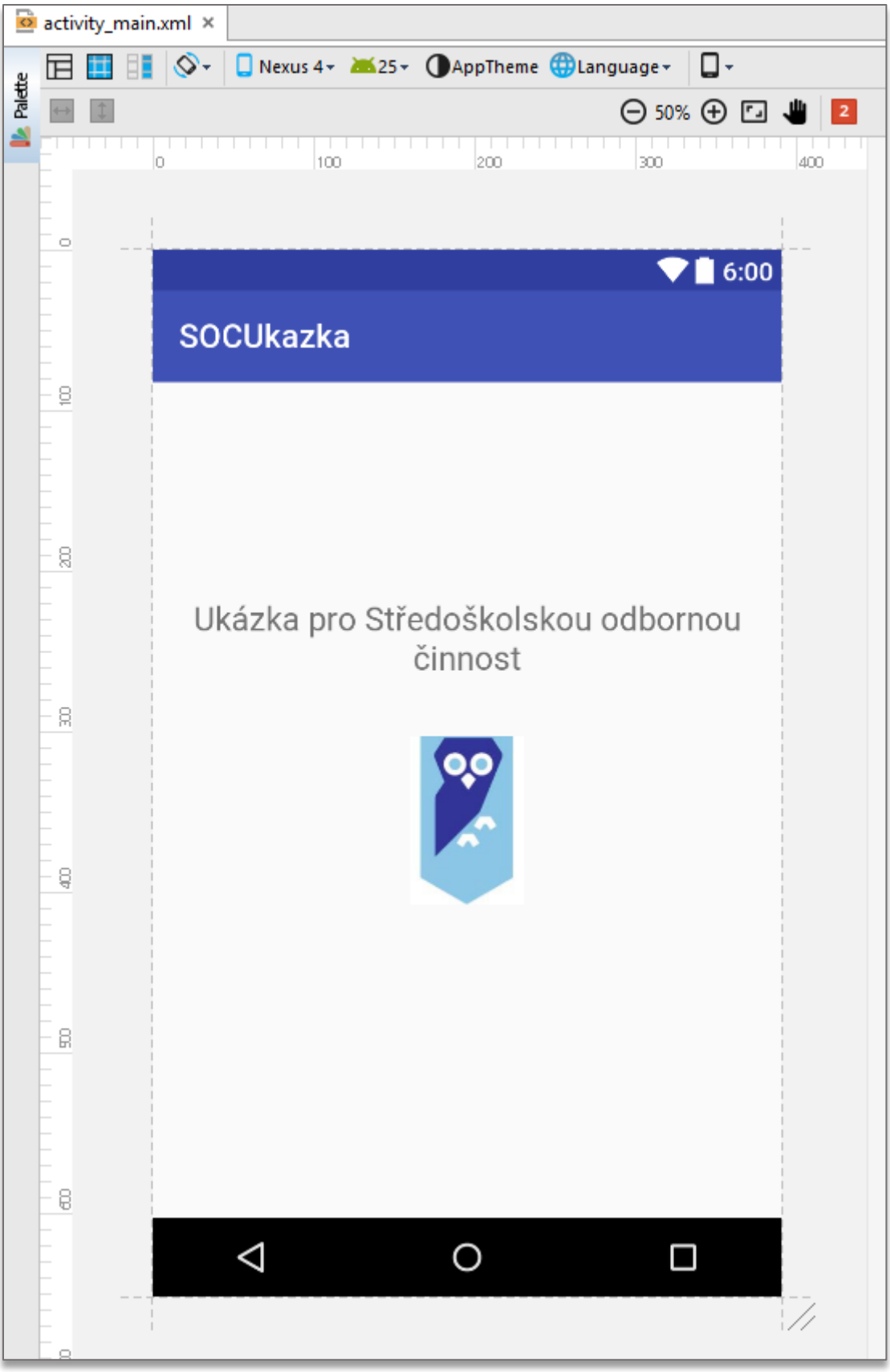
2.5 Android Studio

Android Studio je IDE – vývojové prostředí založené na IntelliJ IDEA – bylo oficiálně představeno firmou Google v roce 2013. Nejvíce se jeho kvalitám blíží IDE Eclipse, v němž jsem vyvíjel dříve, avšak Android Studio svými možnostmi, funkcemi a oficiální podporou Googlu drtivě poráží jakákoliv jiná vývojová prostředí cílená pro vývoj aplikací na Android a je k dispozici ke stažení pro Windows, Mac OS X a Linux.

Mezi funkce, které nejvíce oceňuji, patří: propracovaný editor s možností navrhovat uživatelské prostředí metodou drag-and-drop, jednoduchá a praktická lokalizace aplikací do cizích jazyků, design mód, Gradle – automatický build systém, Firebase databáze a mnohá další.

```
<?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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.kinzl.socukazka.MainActivity">

    <TextView
        android:gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ukázka pro Středoškolskou
         odbornou činnost"
        android:layout_marginTop="117dp"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textSize="20sp"/>

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/soc2016"
        android:id="@+id/imageView"
        android:scaleX="0.5"
        android:scaleY="0.5"
        android:layout_alignTop="@+id/textView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp" />
</RelativeLayout>
```

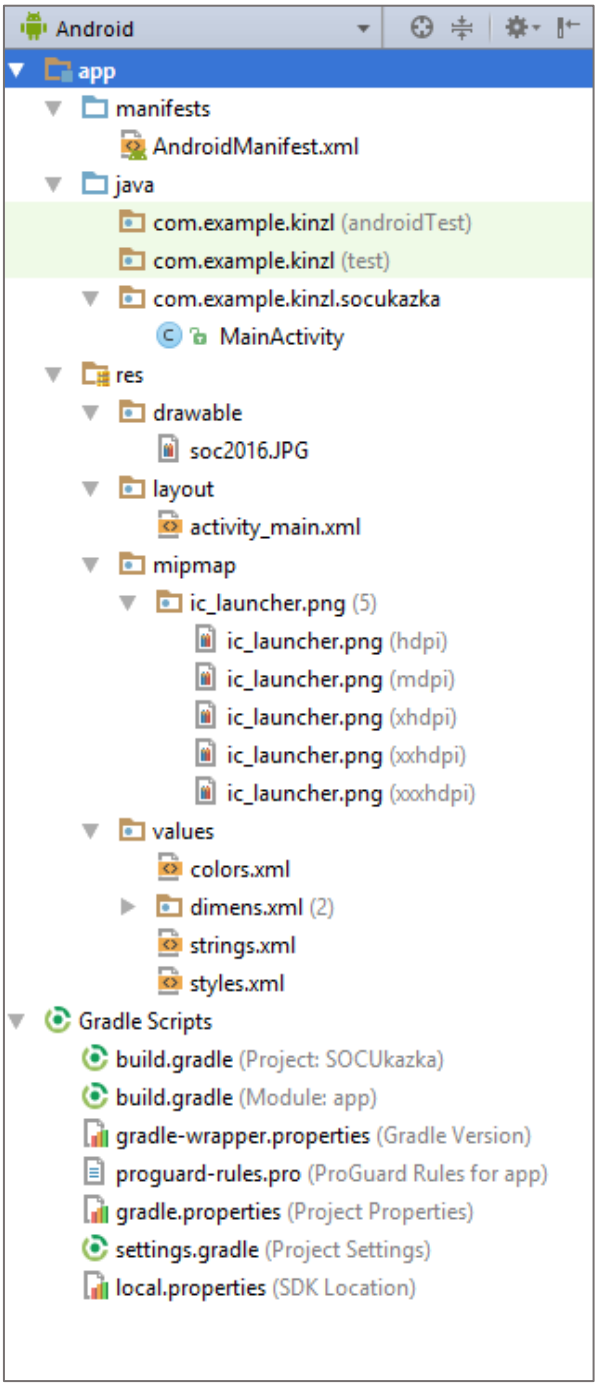
Demonstrace programování jednotlivých komponentů v jednotlivých .xml layout souborech v Android Studiu.



Obrázek 5: Zobrazení rozložení komponent TextView a ImageView daného layoutu pomocí design módu. Kód .xml na ukázce kódu č. 2 promítající se do tohoto zobrazení.

2.5.1 Přehled projektu v Android Studiu

Po celou dobu vývoje jsme pracovali v rámci přehledu projektu. Podobně jako u her nebo Windows aplikací obsahuje projekt vše, co definuje náš finální produkt; od kódu, obrázků a assetů až po testovací kód a build konfiguraci.



Obrázek 7: Přehled nově založeného Android projektu.

2.5.2 Stručný rozbor projektu

V následujícím rozdělení krátce popísujeme, k čemu jsou jednotlivé části projektu.

- **AndroidManifext.xml** uvádí systému Android podstatné informace o aplikaci. Běžně obsahuje povolení (permissions). Například povolení k použití GPS, nebo přístupu k internetu. Dále určuje, která aktivita se spustí po zapnutí aplikace jako první a tak dále.
- **java:** Složka java obsahuje veškeré třídy definující funkce aktivit. Zde se odehrává převážná část programování.
- **res:** Složka res (resources) doslova znamená zdroje. Obsahuje další čtyři podsložky: drawable, layout, mipmap a values. (dále můžeme přidávat složky jako menu a assets).

2.5.2.1 Složka drawable

Složka drawable se v hierarchii nachází pod složkou res. Slouží jako uložiště bitmapových obrázků nebo **.xml** souborů reprezentujících vektorové obrázky. Ke konverzi do `.xml` můžeme použít formáty: **.SVG** a **.PSD.** Souborům v této složce se automaticky po vložení přiřadí funkce „drawable“ a můžeme je volat z layout souborů v `.xml` kódu pomocí příkazu „`@drawable/nazev_obrazku`“ (například když chceme přiřadit komponentě obrázek, viz obrázek č. 6).

2.5.2.2 Složka layout

Složka layout obsahuje `.xml` soubory, které definují vnější vzhled aplikace. Vytváříme v nich uživatelské rozhraní UI a vnější vzhled aktivit, dialogů atd. Převážná část práce se odehrává právě zde.

2.5.2.3 Složka mipmap

Složka mipmap obsahuje bitmapové a vektorové obrázky. Nezahrnuje `.xml` soubory. Pro podporu rozdílných rozlišení displejů zařízení obsahuje více verzí toho samého obrázku. Mezi rozdílná rozlišení patří: `Idpi, mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi…` V této složce se nachází například ikona aplikace. Voláme je pomocí příkazu „`@mipmap/nazev_mipmapy`“.

2.5.2.4 Složka values

Složka values se používá primárně k definování proměnných dimenzí, barev, textů, ale také pro definování celkové barevné kompozice aplikace. Probíhá zde například překlad aplikace do cizích jazyků. Hodnoty v této složce voláme různě; např. textové řetězce příkazem „`@string/nazev_textoveho_retezce`”.

3 Metodologie

V této části práce se budeme stručně věnovat tomu, jaké metody jsme pro dosažení svého cíle využili a z jakých důvodů.

Některé podstatné informace k osvětlení jsme již uvedli v teoretické části práce.

První platformou, na které se aplikace bude vyskytovat a je pro ni primárně vyvinuta, je platforma **Android**, a to už jen pro její oblíbenost a rozšíření mezi cílovými uživateli.

Jelikož se uživatelské rozhraní Androidu programuje pomocí jazyka **Java**, zvolili jsme stejný programovací jazyk.

Vývojovým prostředím je **Android Studio.** Důvodem je to, že v současnosti vystupuje nad všechna ostatní vývojová prostředí určená pro vývoj na platformu Android a je určené k profesionálnímu vývoji aplikací.

3.1 Google Maps API

Google svým vývojářům poskytuje spoustu **API**. API jsou „frameworky“ určené k co nejsnazší implementaci dané funkcionality, jsou tedy často základem dané aplikace. Jelikož je naše aplikace primárně mapa, na které se zobrazují markery jednotlivých baby-friendly zařízení, bylo použití Google Maps API nejrozumnějším řešením. Toto API zároveň poskytuje mnoho užitečných funkcí, včetně využití GPS lokalizace uživatele.

3.2 Firebase databáze

Obrázek 8: Logo Firebase.

Firebase je technologií umožňující vývojářům pro různé platformy vývoj produktů nezávisle na programování a vývoji serverové části. Tvorba je díky tomu rychlejší a jednodušší. Firebase poskytuje funkce, jako jsou ukládání dat, ověřování uživatelů, real-time databází a další.

Pro náš projekt je esenciální právě real-time databáze, do které jsou uložena všechna zařízení. Ta se pomocí následujícího algoritmu promítají do naší aplikace:

```
private DatabaseReference institutionsReference;

public ArrayList<FirebaseInstitution> firebaseInstitutions = new
ArrayList<>();

@Override
public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        ...

      institutionsReference =
      FirebaseDatabase.getInstance().getReference().child("institutions");
      institutionsReference.keepSynced(true);

      institutionsReference.addValueEventListener(new ValueEventListener()
      {
          @Override
          public void onDataChange(DataSnapshot dataSnapshot) {

              firebaseInstitutions.clear();

              for (DataSnapshot institutonSnapshot :
      dataSnapshot.getChildren()) {
                  FirebaseInstitution institution =
      institutonSnapshot.getValue(FirebaseInstitution.class);
                  institution.setKeyID(institutonSnapshot.getKey());
                  firebaseInstitutions.add(institution);
              }
              sortOnCreate();
          }
          @Override
          public void onCancelled(DatabaseError databaseError) {
              Log.e("The read failed: ", databaseError.getMessage());
          }
      });
}
```

Algoritmus načítání zařízení z databáze a jejich ukládání do lokální proměnné.

4 Aplikace

Aplikace **Baby-friendly Map** byla vyvíjena za účelem pomoci rodinám s dětmi vyhledat nejen v jejich okolí zařízení, která jsou přátelská k dětem, a to všechno v intuitivním a jednoduchém uživatelském prostředí **material designu**.

Nic z toho by nebylo možné bez předchozích znalostí pokročilé Javy a vývoje pro mobilní platformy, což znamená čtyřletou praxi, zahrnující počáteční programování příkazového řádku, vývoj hry pro Windows, celoroční práci na hře pro Android s názvem Fizardy, dostupné na Google Play, a půlroční práci ve frameworku LibGDX, určeném pro vývoj multiplatformní hry v Javě.

Aplikace byla vyvíjena s vědomím, že je určena pro masu uživatelů, proto jsme se od začátku snažili o profesionalitu finálního produktu.

4.1 Menu aplikace

Aplikace obsahuje základní „menu“ reprezentující jednotlivé funkce, mezi kterými může uživatel plynule přecházet. Tato menu jsou vyzobrazena na horním toolbaru aplikace.

Mapa

Mapa, která zobrazuje veškerá zařízení z naší databáze pomocí markerů.

Seznam

Seznam zařízení s jejich hodnocením a popisem. Po rozkliknutí jednotlivé položky v seznamu se zobrazí detailnější informace (adresa, telefon, komentáře uživatelů) s možností ohodnotit zařízení. Seznam lze také podle potřeby seřadit.

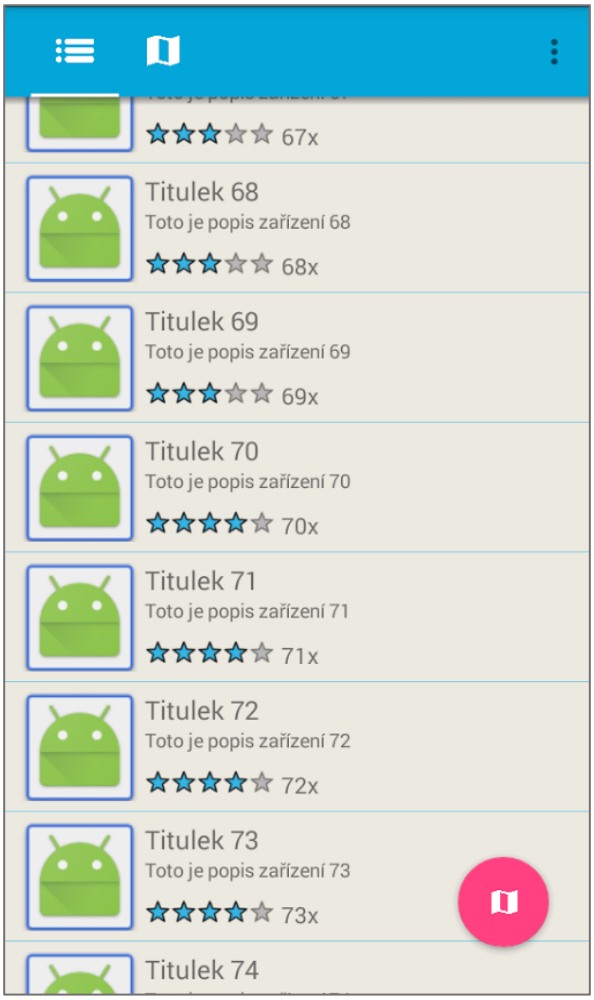
Vyhledávání

Funkce vyhledat dané zařízení na základě jeho názvu nebo lokality, ve které se nachází (funkce přidána později ve vývoji).

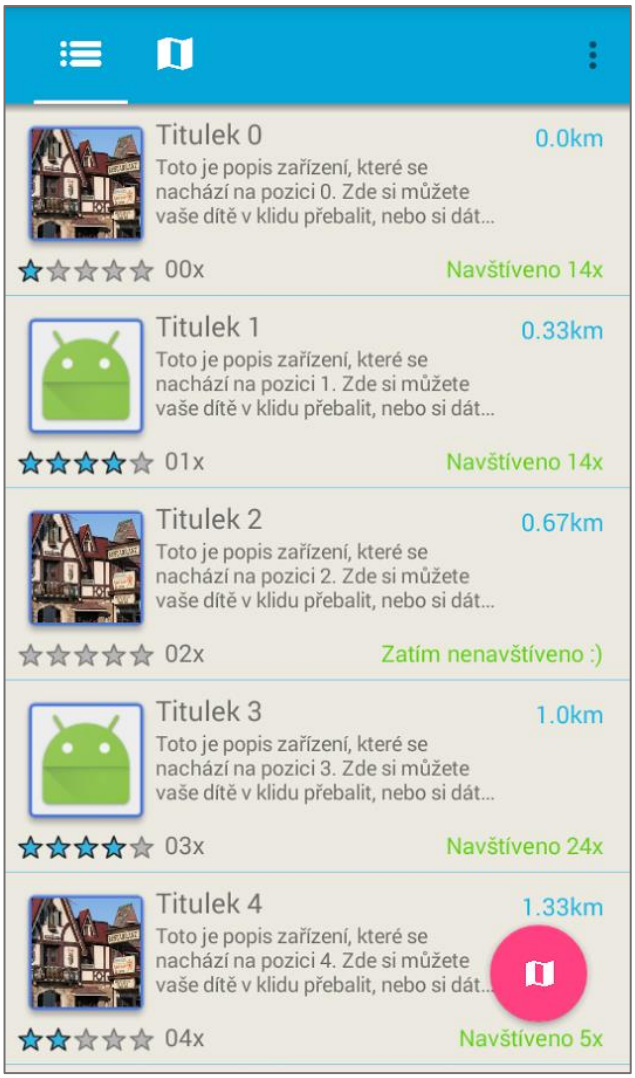
4.2 Evoluce a ukázky aplikace

Aplikace během vývoje prošla několika změnami. Pár z nich bych rád demonstroval v této části práce.

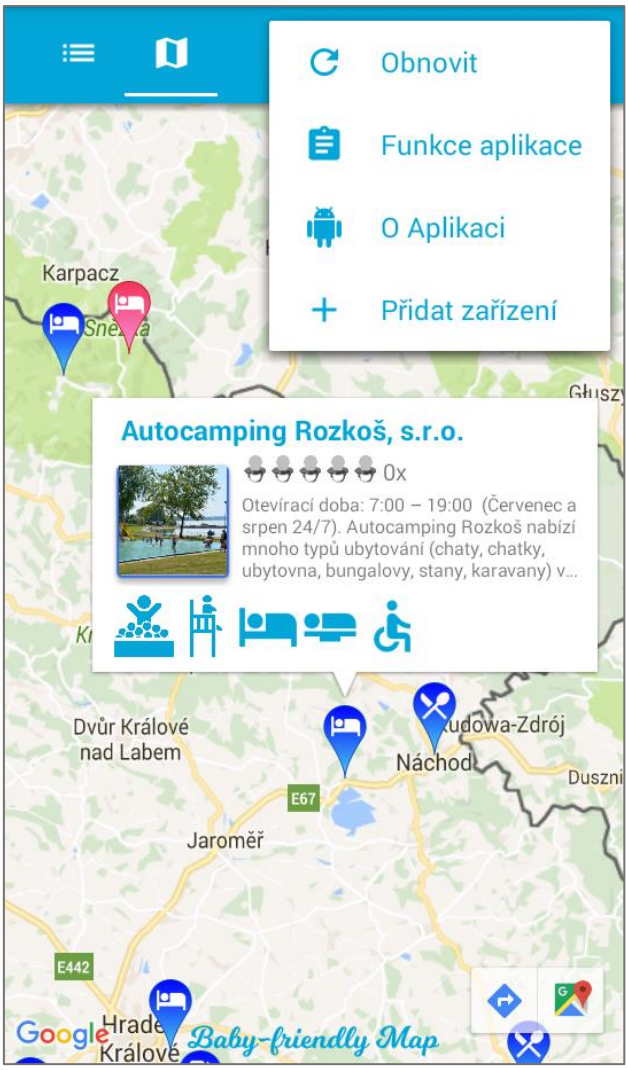
Obrázek 9: Verze aplikace a0.1: Tmavě modrý toolbar. Prázdná mapa bez zařízení..



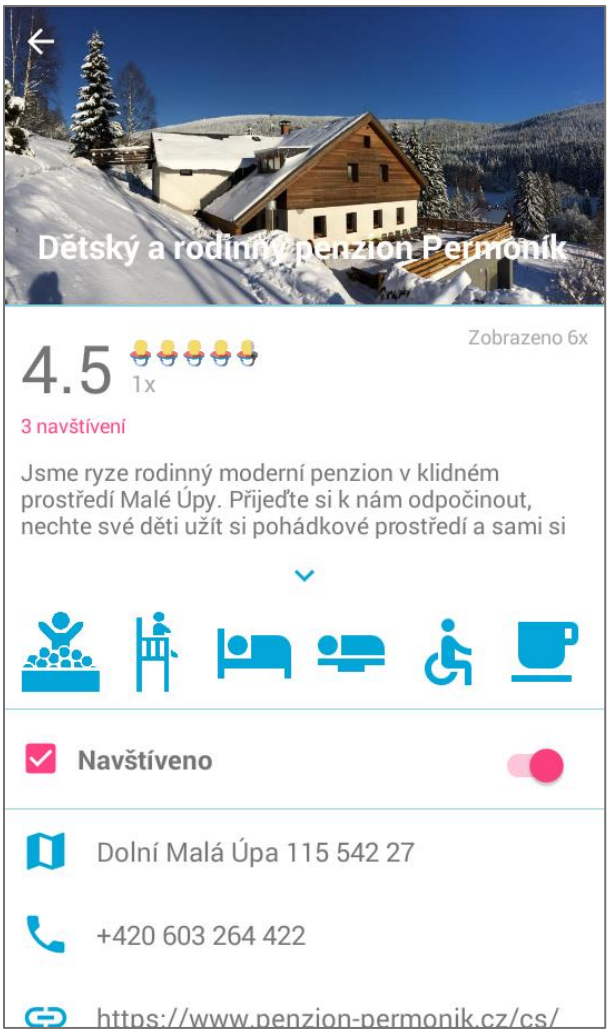
Obrázek 10: Verze aplikace a0.2: Již zvolená finální barva toolbaru. Funkční plynulé přecházení mezi jednotlivými menu (seznam, mapa). Umělý seznam zařízení. Přidáno „Floating Action Button“ do fragmentu listu.



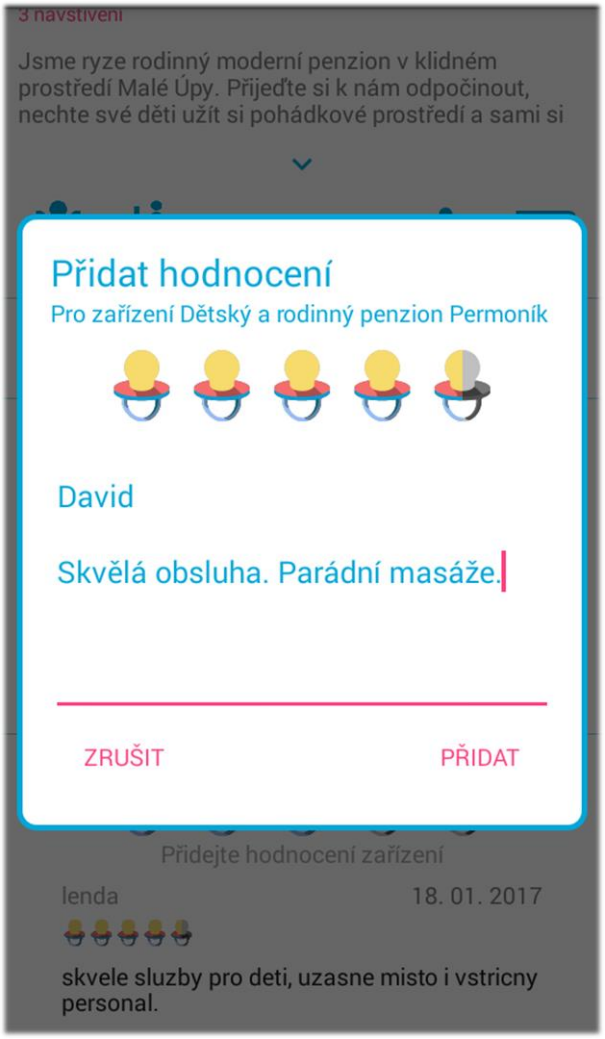
Obrázek 11: Verze aplikace a0.3: Úprava seznamu. Prostornější „řádek“ zařízení. Hodnocení přesunuto pod ikonu zařízení. Přidána vzdálenost mezi polohou uživatele a baby-friendly zařízením.



Obrázek 12: Současná verze aplikace: Jednotlivá baby-friendly zařízení na mapě, rozdělená na restaurace, benzinky a hotely. Vyskakovací okénko baby-friendly zařízení, na kterém se zobrazují symboly (vlevo dětský koutek, dětská židlička, dětská postýlka…). Ve vyskakovacím okénku baby-friendly zařízení si lze také povšimnout nového typu hodnocení reprezentující pět symbolů prázdných dudlíku. Růžový marker vlevo nahoře mapy představuje již uživatelem navštívené zařízení. Vpravo nahoře nové „overflow menu“ s funkcemi: Obnovit, Funkce aplikace, O Aplikaci a Přidat zařízení. Vpravo dole přidány dvě tlačítka na navigaci, odkazující uživatele na Google Maps aplikaci. Dole na mapě je nové logo aplikace „Baby-friendly Map“, zobrazující se pouze na fragmentu mapy.



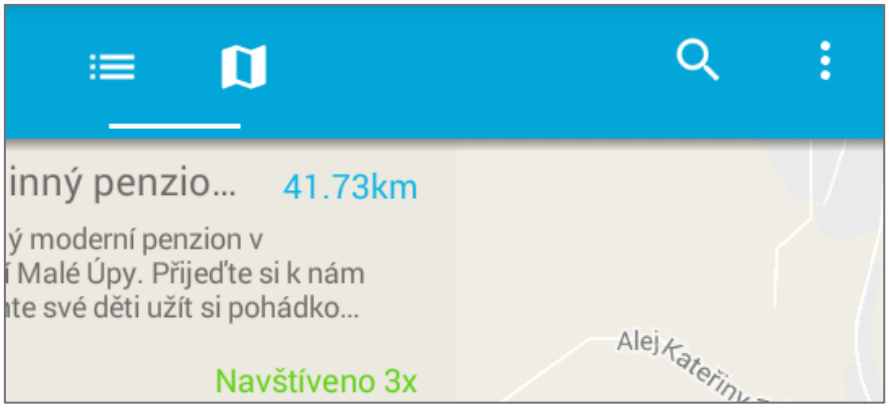
Obrázek 13: Současná verze aplikace: Detailní aktivita baby-friendly zařízení. Nahoře ikona zařízení (obrázek). Pod ikonou nalevo celkové hodnocení, od něj vpravo toto hodnocení demonstrované počtem barevných dudlíků, pod kterými je počet hodnocení. Vpravo pod ikonou počet zobrazení aplikace (přičítající se na základě prvotního otevření detailní aktivity jednotlivého uživatele). Pod hodnocením jen třířádkový popisek zařízení, který lze plynule expandovat dle délky. Pod popisem se nachází lišta se symboly vlastností zařízení. Pod touto lištou „páčka“ pro navštívení uživatelem. Následně textové odkazy ve formě adresy, telefonního čísla, webové stránky a emailu baby-friendly zařízení.



Obrázek 14: Současná verze aplikace: Přidávání hodnocení baby-friendly zařízení uživatelem v detailní aktivitě.

4.3 Hlavní toolbar

Toolbar je tou nejdůležitější navigační součástí v aplikaci. Pomocí komponenty **PagerSlidingBarStrip** můžeme spravovat fragmenty mapy a seznamu tak, že jsou vedle sebe a uživatel mezi nimi může přecházet velmi intuitivním způsobem, buď přetáhnutím prstu na vedlejší fragment, nebo volbou menu na hlavním toolbaru. Na hlavním toolbaru se také nachází funkce **vyhledání baby-friendly zařízení** (lupa) a samotné **overflow menu** (tři tečky nad sebou), po jehož rozkliknutí se zobrazí více sekundárních možností, které lze vidět na obr. 12 (Obnovit, Seznam funkcí, O aplikaci, …).



Obrázek 15: Hlavní toolbar. Uživatel právě přechází mezi fragmentem seznamu a mapy.

```
<android.support.v7.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#02a6d8"
    android:minHeight="?android:attr/actionBarSize"
    app:theme="@style/MyDarkToolbarStyle">

    <com.astuetz.PagerSlidingTabStrip
        android:id="@+id/tabs"
        android:layout_width="wrap_content"
        android:layout_height="?android:attr/actionBarSize"
        android:layout_alignParentRight="true"
        android:layout_marginLeft="8dp"
        android:background="#02a6d8"
        app:pstsDividerColor="#02a6d8"
        app:pstsIndicatorColor="#fff"
        app:pstsIndicatorHeight="2dp"
        app:pstsShouldExpand="false"
        app:pstsUnderlineHeight="0dp"/>
</android.support.v7.widget.Toolbar>
```

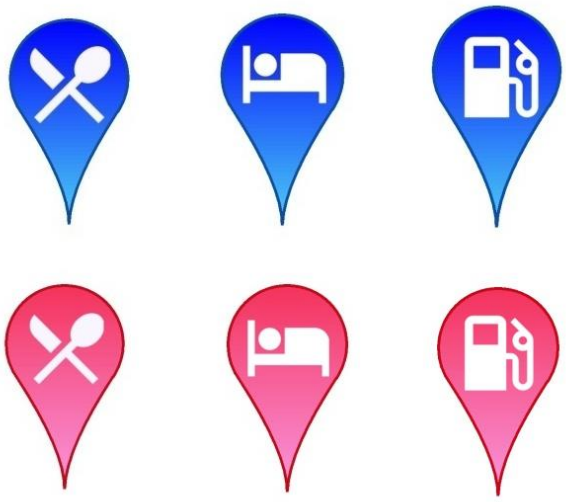
Komponenty Toolbaru a jeho součást PagerSlidingTabStrip.

4.4 Fragment mapy

Nejdůležitější součástí aplikace je mapa. V této části práce jsou podrobně demostrovány její funkce, mezi které patří **markery** baby-friendly zařízení a **GPS lokalizace**.

4.4.1 Markery

Marker je kapička ukazující na dané místo zeměpisné délky a šířky (latitude a longitude) na mapě. V naší aplikaci reprezentuje marker jedno baby-friendly zařízení, své vlastní markery rozdělujeme podle typu baby-friendly zařízení. Liší se symbolem uvnitř kapičky; příbor pro **restaurace**, postel pro **hotely** a nádrž pro **benzínové pumpy**. Markery se liší také barvou; vámi nenavštívené jsou modré, navštívené červenorůžové.



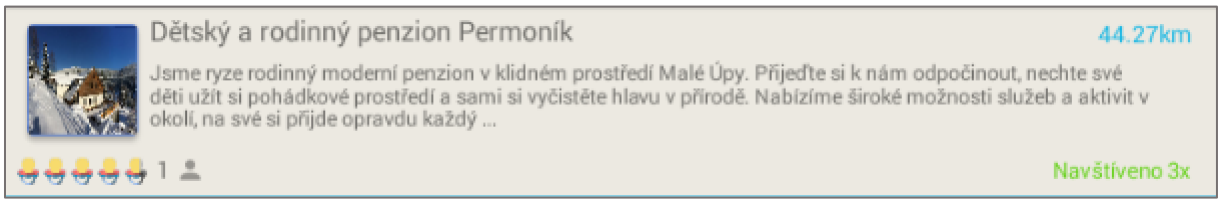
Obrázek 16: Druhy markerů baby-friendly zařízení: restaurace, hotely, benzínové pumpy. Modře nenavštívené a červeno-růžové navštívené.

4.4.2 GPS lokalizace

GPS lokalizace je funkční součástí aplikace. Pokud uživatel otevře aplikaci již se zapnutou GPS, aplikace na něj plynule zamíří. Pokud má GPS vypnutou a rozhodne se zamířit na sebe, může tak učinit pomocí „my location button“, po jehož stisku se aplikace zeptá, jestli chce uživatel spustit GPS. Frekvence dotazu na uživatelovu polohu je nastavena na pět sekund.

4.5 Fragment seznamu

Fragment seznamu zobrazuje pod sebou seřazená baby-friendly zařízení. Seznam funguje na bázi načítání zařízení z databáze (realtime) a jejich následného ukládání do lokální proměnné. Uživatel si zařízení může seřadit podle daných parametrů.



Obrázek 17: Jedna položka v seznamu. Nalevo ikona zařízení, název, popis, hodnocení ve formě pěti dudlíků, a napravo vzdálenost od zařízení a navštívenost.

Seřazení

- Abecedně
- Navštívenosti
- Hodnocení
- Vzdálenosti

```
public class CustomComparator implements Comparator<FirebaseInstitution> {
    @Override
    public int compare(FirebaseInstitution obj1, FirebaseInstitution obj2){
        try {
            switch (orderBy) {
                case ALPHABETICALLY:
                    return obj1.getTitle().compareTo(obj2.getTitle());
                case VISITS:
                    return ((Integer)
obj2.getNumberOfVisits()).compareTo(obj1.getNumberOfVisits());
                case RATING:
                    return ((Float)
obj2.getCurrentRating()).compareTo(obj1.getCurrentRating());
                case DISTANCE:
                    return sortByDistance(obj2, obj1);
                default:
                    return obj1.getTitle().compareTo(obj2.getTitle());
            }
        } catch (Exception e) {
            return 0;
        }
    }
}
```

Třída Comparator „porovnávač“ určená k seřazování jednotlivých zařízení.

```
public void sortOnCreate(){
    if(this.orderBy != DEFAULT && this.orderBy != DISTANCE) {
        Collections.sort(firebaseInstitutions, new CustomComparator());
    }
    else if(this.orderBy == DISTANCE){
        if(currentLatitude != 0 && currentLongitude != 0){
            Collections.sort(firebaseInstitutions, new CustomComparator());
        }
        else{
            if(MainActivity.babyfriendlyMapFragment.mGoogleApiClient != null)
                MainActivity.babyfriendlyMapFragment.showEnableLocationDialog();
            else{
                Toast.makeText(getActivity(), "Nepodařilo se seřadit zařízení podle
vzdálenosti.", Toast.LENGTH_SHORT).show();
            }
        }
    }
    try {
        if(firebaseInstitutionListAdapter == null && getActivity() != null){

            firebaseInstitutionListAdapter = new
FirebaseInstitutionListAdapter(getActivity(),
                    firebaseInstitutions);
            setListAdapter(firebaseInstitutionListAdapter);}
        else
            try {
                firebaseInstitutionListAdapter.notifyDataSetChanged();}
            catch (Exception e){}
    } catch (Exception e){}

    if(index!=-1){
        try {
            this.getListView().setSelectionFromTop(index, top);
        } catch (Exception e) {}
    }
}
```

Metoda sortOnCreate() prvotního seřazení zařízení v seznamu.

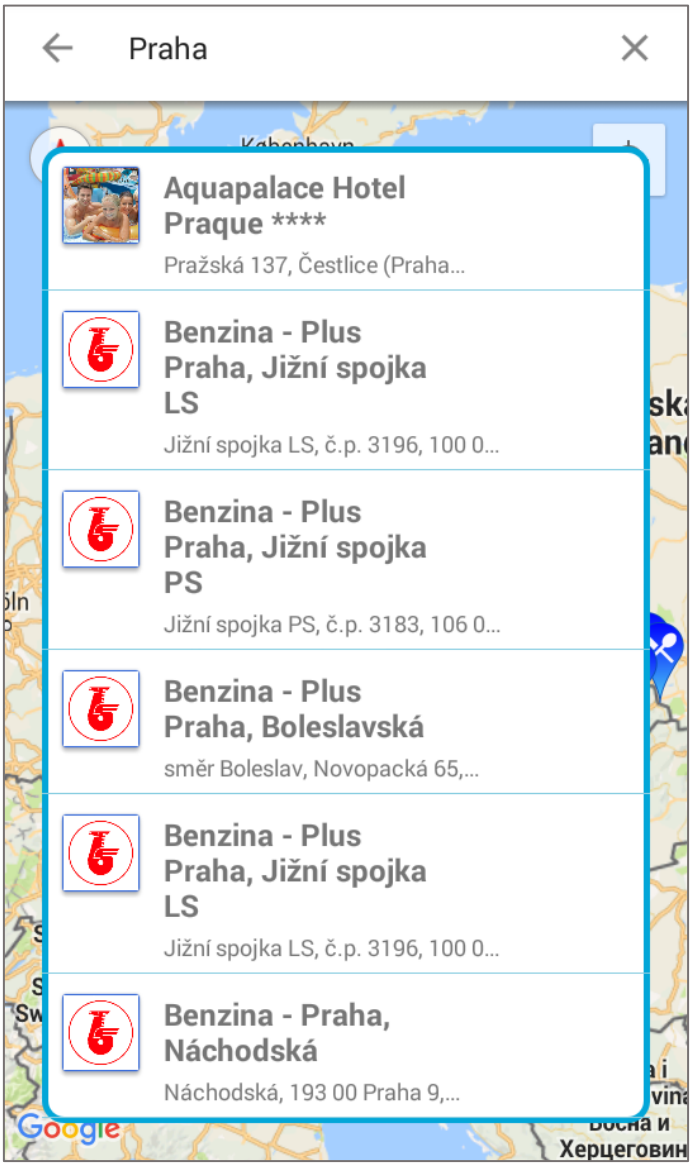
```
private int sortByDistance(FirebaseInstitution obj1, FirebaseInstitution obj2){
        float[] resultsObj1 = new float[1];

        Location.distanceBetween(
                currentLatitude, currentLongitude, obj1.getLatitude(),
obj1.getLongitude(), resultsObj1);

        float distanceBetweenObj1 = resultsObj1[0];
        distanceBetweenObj1 /= 1000;
        distanceBetweenObj1 = round(distanceBetweenObj1, 2);
        float[] resultsObj2 = new float[1];

        Location.distanceBetween(
                currentLatitude, currentLongitude, obj2.getLatitude(),
obj2.getLongitude(), resultsObj2);

        float distanceBetweenObj2 = resultsObj2[0];
        distanceBetweenObj2 /= 1000;
        distanceBetweenObj2 = round(distanceBetweenObj2, 2);

        return ((Float) distanceBetweenObj2).compareTo(distanceBetweenObj1);
}
```

Metoda sortByDistance() k porovnávání zařízení ve vzdálenosti od současné polohy uživatele.

4.6 Fragment vyhledávání

V naší aplikaci má uživatel možnost dané zařízení vyhledat pomocí funkce **vyhledávání.** Po stisku lupy se plynulou animací zobrazí **fragment vyhledávání,** který má určitou velikost a po jeho zobrazení je stále vidět mapa nebo seznam.

Zařízení lze vyhledat na základě názvu nebo lokality. To znamená, že pokud uživatel napíše do vyhledávání „Praha“, zobrazí se veškerá zařízení nacházející se v Praze.



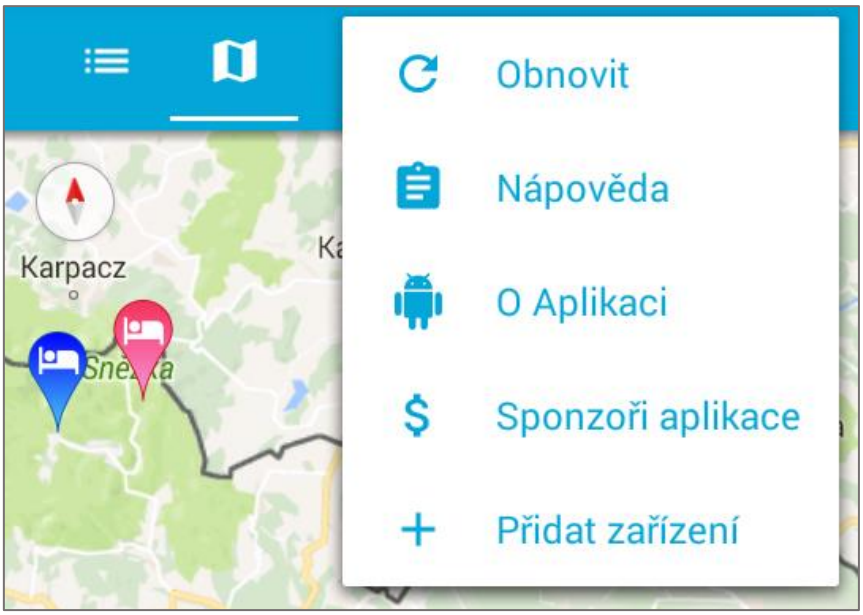
Obrázek 18: Fragment vyhledávání.

```
private class InstitutionsFilter extends Filter{

    @Override
    protected FilterResults performFiltering(CharSequence charSequence) {

        String filterString = charSequence.toString().toLowerCase();

        FilterResults results = new FilterResults();

        final List<FirebaseInstitution> list = originalData;

        int count = list.size();

        final ArrayList<FirebaseInstitution> nList = new
ArrayList<>(count);

        for(int i = 0; i < count; i++){

            /**inicializace nových informací*/
            String filterableTitle = list.get(i).getTitle();
            String filterableAdress = list.get(i).getDetail_adress();

            FirebaseInstitution firebaseInstitution = list.get(i);

            //filtr pracuje mezi tímto Stringem a inputem uživatele
(filterString):
            String filterableData = filterableTitle + filterableAdress;

            if(filterableData.toLowerCase().contains(filterString)){;

                FirebaseInstitution institution = firebaseInstitution;

                nList.add(institution);
            }
        }

        results.values = nList;
        results.count = nList.size();

        return results;
    }

    @Override
    protected void publishResults(CharSequence charSequence, FilterResults
filterResults) {
        filteredData = (ArrayList<FirebaseInstitution>)
filterResults.values;
        notifyDataSetChanged();
    }
```

Třída InstitutionsFilter funguje jako filtr zařízení skrz uživatelem zadaný textový řežezec. Výsledkem jsou pouze ta zařízení, která ve svém názvu nebo adrese obsahují pouze takový text, který je zadaný uživatelem.

4.7 Overflow menu

Overflow menu slouží jako menu zobrazení dalších užitečných funkcí. Zobrazí se po rozkliknutí třech bílých teček vpravo na hlavním toolbaru.



Obrázek 19: Overflow menu

Overflow menu obsahuje následující funkce:

- **Obnovit** – znovu načte zařízení z databáze
- **Nápověda** – otevře aktivitu obsahující popisky jednotlivých funkcí aplikace.
- **O Aplikaci** – zobrazí logo, ikonu, tvůrce aplikace a jejich kontakty.
- **Sponzoři aplikace** – zobrazí seznam všech sponzorů, kteří na aplikaci přispěli. Převážně z kampaně na Startovači.
- **Přidat zařízení** – odkáže uživatele na webovou stránku, na které se nachází formulář k vyplnění požadovaných informací o zařízení atd.

4.8 Detailní aktivita

Detailní aktivita je velmi podstatnou součástí aplikace. Poskytuje totiž uživateli detailnější informace o baby-friendly zařízení, například zvětšenou ikonu v podobě hlavičky, detailní popis zařízení, symboly vlastností, různé hyperodkazy (telefon, email, webová adresa, …), další statistiky, nebo třeba přidávání hodnocení.

Na detailní aktivitu se lze dostat třemi intuitivními způsoby: kliknutím na vyskakovací okénko markeru, kliknutím na položku v fragmentu seznamu a kliknutím na položku v fragmentu vyhledávání.

Detailní aktivita se skládá ze dvou layoutů, hlavičky a těla. Hlavička obsahuje ikonu baby-friendly zařízení a nadpis, tělo pak řadu dalších informací.

4.8.1 Hlavička detailní aktivity

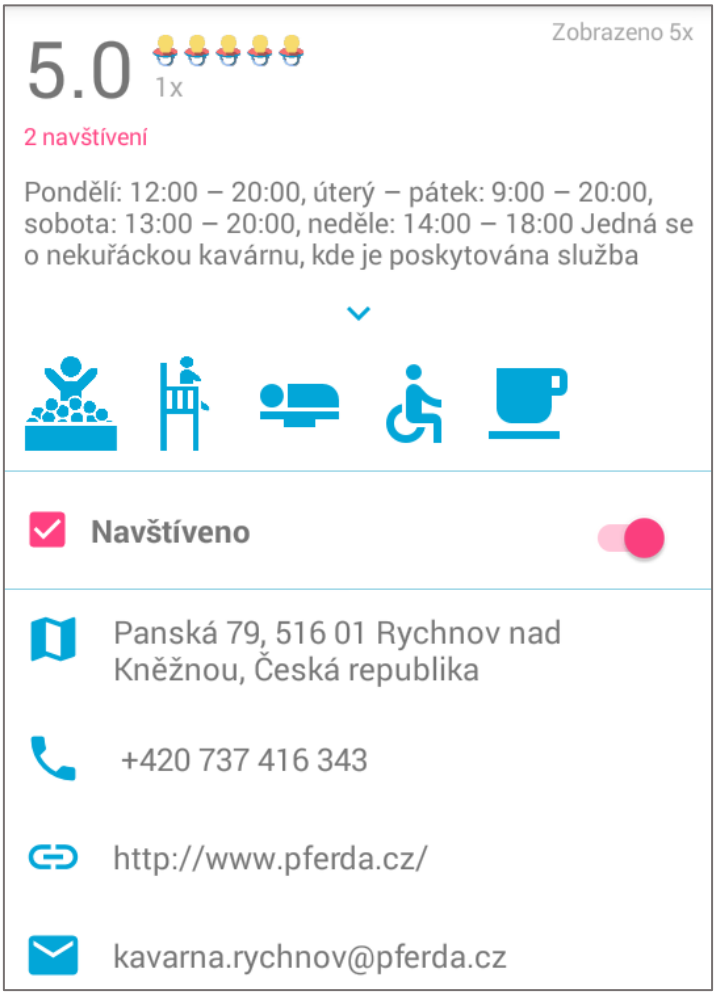
Obrázek 20: Hlavička baby-friendly zařízení v detailní aktivitě

Hlavička je poměrně jednoduchá. Obsahuje seříznutou ikonu zařízení, název a šipku zpět.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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"
    android:fitsSystemWindows="true"
 tools:context="com.kinzlstanislav.babyfriendly.activities.DetailActivity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="@dimen/app_bar_height"
        android:fitsSystemWindows="true"
        android:theme="@style/Babyfriendly.AppBarOverlay">

        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/toolbar_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed">

            <ImageView
                android:id="@+id/detail_headerimage"
                android:layout_width="fill_parent"
                android:layout_height="match_parent"
                android:scaleType="centerCrop"
                app:layout_collapseMode="pin"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Název zařízení"
                android:textSize="20sp"
                android:layout_gravity="center_vertical"
                android:paddingLeft="20dp"
                android:paddingTop="110dp"
                android:textColor="#ffff"
                android:textStyle="bold"
                android:ellipsize="end"
                android:maxLines="1"
                android:id="@+id/detail_headertext"/>

            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/back_arrow_white"
                android:padding="10dp"
                android:id="@+id/detail_back_arrow"/>

        </android.support.design.widget.CollapsingToolbarLayout>
    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/content_detail" />
</android.support.design.widget.CoordinatorLayout>
```

Layout hlavičky baby-friendly zařízení. První ImageView zastupuje ikonu baby-friendly zařízení, TextView je název zařízení a druhý ImageView reprezentuje šipku zpět.

4.8.2 Tělo detailní aktivity



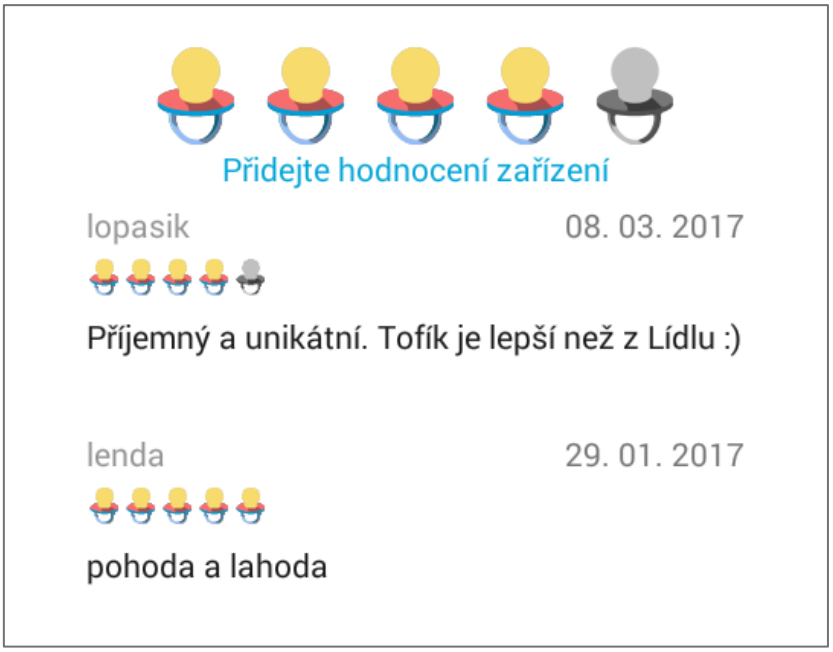
Obrázek 21: Tělo detailní aktivity.

Tělo obsahuje následující prvky jdoucí odshora dolů:

- **Hodnocení** – TextView
- **Hodnocení demonstrované pomocí pěti dudlíků** – RatingBar
- **Počet zobrazení detailní aktivity** – TextView
- **Počet hodnocení** - TextView
- **Počet navštívení baby-friendly zařízení** – TextView
- **Detailní informace** – vlastní view jménem ExpandablePanel obsahující TextView a ImageButton k zobrazení dalších informací.
- **Obrázky reprezentující vlastnosti zařízení** – šest ImageView
- **Navštíveno** – ImageView, TextView a SwitchCompat
- **Hyperodkazy**: adresa, telefon, webová adresa, emailová adresa – čtyři ImageView a čtyři TextView, po jejichž stisknutí se otevře daná aktivita.

4.8.3 Komentáře a hodnocení

Uživatel Baby-friendly Map má možnost jednotlivá zařízení po návštěvě ohodnotit. Toto hodnocení se přidá do seznamu dalších a vytvoří se komentář.



Obrázek 22: Přidávání hodnocení a list uživatelů, kteří zařízení již ohodnotili.

Naše aplikace disponuje vlastním RatingBarem, hodnocení totiž reprezentují dudlíky. Po vybrání počtu dudlíků se zobrazí následující vyskakovací okénko:

Obrázek 23: Vyskakovací okénko k přidávání hodnocení.

4.8.3.1 Model komentáře

Model komentáře stanoví dané proměnné a následně je po stisku tlačítka „přidat“ (viz obrázek 23) nahraje do databáze. Pokud je uživatel offline, komentář se nahraje, jakmile se připojí k síti. K tomu slouží model komentáře **FirebaseComment**.

```
public class FirebaseComment {

    String prezdivka;
    String comment;
    HashMap<String, Object> timestampCreated;
    float rating;

    public FirebaseComment(){}

    public FirebaseComment(String comment, String prezdivka, float rating)
{
        this.comment = comment;
        this.prezdivka = prezdivka;
        this.rating = rating;

        timestampCreated = new HashMap<>();
        timestampCreated.put("timestamp", ServerValue.TIMESTAMP);

    }
    //getters and setters...

      @Exclude
      public long getTimestampCreatedLong(){
          return (long)timestampCreated.get("timestamp");
      }

}
```

Model FirebaseComment.

`HasmMap<String, Object> timestampCreated;` je proměnná, ve které je uložený čas přidání komentáře/hodnocení.

Přidá-li uživatel komentář, zadané parametry modelu FirebaseComment se převedou do formátu `.json` a nahrají do databáze. To se promítne v přidání komentáře do listu příspěvků. Příspěvek se následně zařadí nad ostatní, a to díky vlastnímu CustomComparatoru, viz níže. Ten porovnává hodnotu timestamp (unikátní serverová hodnota času, kterou lze převést na rok, den, hodinu...).

```
public class CustomComparator implements Comparator<FirebaseComment> {
    @Override
    public int compare(FirebaseComment obj1, FirebaseComment obj2) {
        return ((Long)
obj2.getTimestampCreatedLong()).compareTo(obj1.getTimestampCreatedLong());
    }
}
```

4.8.3.2 Nahrávání komentáře do databáze

Firebase určuje, jakým způsobem se mají dané „child“ (komentář je child parentu „comments“ a ten je zas child parentu zařízení), nahrávat do databáze.

Následující metoda createNewComment nahrání komentáře zajistí. Ověří, zda uživatel již zařízení hodnotil, pokud ano, pouze přepíše jeho staré hodnocení, pokud ne, přidá jeho nové hodnocení do databáze:

```
private void createNewComment(String name, String comment, float rating){

    //creates new unique user comment id key
    if(!alreadyAddedComment){
        userKey = databaseReference.child("comments").push().getKey();

        SharedPreferences settings = this.getSharedPreferences(
                keyId, 0);
        SharedPreferences.Editor editor = settings.edit();

        editor.putString(IDUserKey, userKey);

        editor.commit();

        if(!visited){
            materialSwitch.setChecked(true);
            materialSwitch.callOnClick();
        }

        alreadyAddedComment = true;
        save();
    }
    else{
        SharedPreferences settings = this.getSharedPreferences(keyId, 0);

        userKey = settings.getString(IDUserKey, "user_comment");
    }

    //creating the comment
    FirebaseComment firebaseComment = new FirebaseComment(comment, name,
rating);

    //adding comment into the database/editing

databaseReference.child("comments").child(userKey).setValue(firebaseComment
);
}
```

Metoda createNewComment - nahrávání komentáře do databáze.

4.9 Databáze zařízení

Databáze je alfou a omegou Baby-friendly Map. Bez ní by byla mapa nefunkční, jelikož by nemohla zobrazovat jakákoliv opravdová zařízení. Tento úkon lze provést manuálně přímo v kódu, například načítat zařízení z .txt dokumentů nebo sql databáze. To ale znamená, že aplikace nemůže disponovat žádným systémem hodnocení a načítat zařízení real-time.

Z toho důvodu byly vyhledávány nejvýhodnější způsoby řešení. Prostřednictvím vyhledávání jsme se dozvěděli o **real-time databázi**, jež patří mezi jednu z funkcí platformy **Firebase**, kterou zaštiťuje samotná společnost Google – současný vývojář Androidu (viz kapitola 3.2 Firebase Databáze).

Jako nejlepší možné řešení ukázala se databáze Firebase, proto jsme se rozhodli právě pro ni.

Pro tento účel byla vytvořena třída **FirebaseInstitution**, jež je modelem baby-friendly zařízení a obsahuje všechny potřebné proměnné.

```
public class FirebaseInstitution implements Serializable {

    String title;
    String description;
    String image_url;

    float currentRating;
    int numberOfRatings;
    int numberOfVisits;
    int numberOfViews;

    double latitude;
    double longitude;

    boolean vlastnost_bezbarierovyPristup;
    boolean vlastnost_detskyKoutek;
    boolean vlastnost_prebalovaciPult;
    boolean vlastnost_detskaPostylka;
    boolean vlastnost_kavaBezKofeinu;
    boolean vlastnost_detskaZidlicka;

    String detail_adress;
    String detail_phoneNumber;
    String detail_webAdress;
    String detail_emailAdress;

    String keyID;

    boolean marker_restaurant;
    boolean marker_pumpa;
    boolean marker_hotel;

    /* Inicializace, konstruktor, gettery a settery... */
}
```

Model FirebaseInstitution

4.9.1 Přidávání zařízení do databáze

Stejně jako komentáře jsou baby-friendly zařízení v databázi ve formátu .json. Ale narozdíl od komentářů, které se do databáze nahrají sami, se baby-friendly zařízení vkládají ručně.

```
{
  "title" : "Koala Café",
  "description" : "Otevírací doba: Út – Ne od 10:00 do 19:00. Dětské zá-
bavní centrum s kavárnou a občerstvením. Moře zábavy pro kluky i holky.
Velké bludiště s tobogány a skluzavkami. Míčkové bazínky, šlapací káry a
prostorný hrací koutek pro nejmenší. Kavárna s čerstvě praženou kávou, Wi-
Fi připojení zdarma!",
  "image_url" : "http://www.koalacafe.cz/plogger/index.php?level=al-
bum&id=18",
  "detail_adress" : "Dolnoměcholupská 209/17, 10200, Praha 10, Hostivař,
Česká republika",
  "detail_emailAdress" : "info@koalacafe.cz",
  "detail_phoneNumber" : "+420 212 241 171",
  "detail_webAdress" : "www.koalacafe.cz",
  "latitude" : 50.0561431,
  "longitude" : 14.5429808,
  "vlastnost_bezbarierovyPristup" : false,
  "vlastnost_detskaPostylka" : true,
  "vlastnost_detskaZidlicka" : true,
  "vlastnost_detskyKoutek" : true,
  "vlastnost_kavaBezKofeinu" : true,
  "vlastnost_prebalovaciPult" : true,
  "marker_restaurant" : true,
  "marker_pumpa" : false,
  "marker_hotel" : false
}
```

Ukázka zařízení uloženého v databázi ve formátu json.

Výše uvedená ukázka se skládá z řady proměnných, které ve finále tvoří dané baby-friendly zařízení. Pro nahrání takového zařízení stačí mít zaplacen vývojářský účet a aktivovanou databázi, do které se následně importují jednotlivá zařízení. Zapsaný kód se následně postará o to, aby se zařízení okamžitě nahrálo do aplikace. Pokud je uživatel offline, zařízení se ukáže po připojení k síti.

5 Závěr

Celková tvorba aplikace trvala půl roku. Ze začátku byly nejdůležitější společné schůzky, na kterých jsme se dohadovali o fundamentálních principech aplikace. Následovala kontrola terénu, tedy debata o tom, co je a není možné udělat, jak to udělat a co je potřeba se naučit. Jelikož jsme mohli využít znalostí z předešlých projektů, stačilo se naučit strukturu aplikace (callbacky, layouty, aktivity, …). V tomto období jsme se naučili, jak postupovat při návrhu a tvorbě profesionální aplikace pro Android a jak vést konzultace s klienty.

Museli jsme se také naučit, jak postupovat při tvorbě projektu v malém týmu, jak odlišit dobrou a špatnou grafiku, jak práci dobře zvládnout jak z kreativního, tak technologického úhlu pohledu (oběma přisuzujeme stejnou důležitost). Ukázalo se, k naší radosti, že jsme schopni dlouhodobě spolupracovat.

Přestože je aplikace již hotová, teprve po kampani na Startovači se chystá prorazit na Google Play. V blízké budoucnosti ji také čekají mnohé aktualizace, v nichž budeme přidávat další symboly pro jednotlivá zařízení a databázi s trasami pro kočárky.

Nyní je již aplikace dostupná na Google Play.

Použitá literatura

1. Mobilní aplikace. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2015- [cit. 2017-02-05]. Dostupné z: https://cs.wikipedia.org/wiki/Mobiln%C3%AD_aplikace
2. Material Design. In: *Gizchina* [online]. Praha: Gizchina, 2014 [cit. 2017-02-05]. Dostupné z: http://gizchina.cz/2015/03/02/okenko-strycka-googla-material-design/
3. Java. In: *Interval.cz* [online]. Česká Republika: Jiří Semecký, 2002 [cit. 2017-02-05]. Dostupné z: https://www.interval.cz/clanky/naucte-se-javu-uvod/
4. Java a Android. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-02-05]. Dostupné z: https://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk)

Obrázky