

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

2016

Lukáš Fichna

# Zadání bakalářské práce

Student: **Lukáš Fichna**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Railsformers s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radovan Fusek**

Konzultant bakalářské práce: Ing. Richard Lapiš

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016

*Lukáš Faldus*  
.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 14. dubna 2016

**Railsformers s.r.o.**  
IČ: 24704440 DIČ: CZ24704440  
www.railsformers.com  
info@railsformers.com



Děkuji Ing. Jiřímu Kubicovi za možnost vykonávat praxi ve firmě Railsformers s.r.o. a umožnit mi tímto nabýt poznatky při práci na reálném projektu. Dále děkuji Bc. Richardovi Lapišovi za řízení týmu a odborné rady. Taktéž děkuji svému vedoucímu Ing. Radovanovi Fuskovi za pomoc a připomínky k této práci. Za bezmeznou podporu při studiu děkuji všem svým blízkým.

## **Abstrakt**

Cílem této bakalářské práce je popis vykonané individuální odborné praxe ve firmě Railsformers s.r.o. Práce se skládá z 8 kapitol, které popisují zaměření firmy, pracovní zařazení studenta a úlohy, které byly během praxe zadány a vypracovány. V posledních kapitolách jsou popsány získané vědomosti a zhodnocení celkového přínosu absolvování praxe a její výsledky.

**Klíčová slova:** Railsformers s.r.o., Android, odborná praxe

## **Abstract**

The aim of this bachelor thesis is description of executed individual professional practice in the company Railsformers s.r.o. The thesis consists of 8 chapters, that describe the focus of the company, job title and tasks, that were given and done during practice. In the last chapters are described gained knowledge and assessment of the overall contribution of executed practice and its results.

**Key Words:** Railsformers s.r.o., Android, professional practice

# Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam výpisů zdrojového kódu	10
<b>1 Úvod</b>	<b>11</b>
<b>2 Zaměření firmy a pracovní zařazení studenta</b>	<b>12</b>
2.1 O firmě	12
2.2 Pracovní zařazení studenta a pracovní náplň	12
2.3 Pracovní prostředí	12
<b>3 Použité technologie a nástroje</b>	<b>13</b>
3.1 Java	13
3.2 Android SDK	13
3.3 Android Studio	13
3.4 Git	13
<b>4 Zadané úkoly a jejich řešení</b>	<b>14</b>
4.1 Příprava	14
4.2 Struktura aplikace	14
4.3 Multijazyčnost aplikace	15
4.4 Kontrola připojení k internetu	17
4.5 Nákupní seznam	18
4.6 Uživatelské nastavení aplikace	20
<b>5 Znalosti a dovednosti uplatněné v průběhu praxe</b>	<b>21</b>
<b>6 Znalosti a dovednosti scházející v průběhu praxe</b>	<b>22</b>
<b>7 Výsledky absolvované praxe a její zhodnocení</b>	<b>23</b>
<b>8 Závěr</b>	<b>24</b>
<b>Literatura</b>	<b>25</b>

## Seznam použitých zkratek a symbolů

API	– Application Programming Interface
aktivita	– jedna obrazovka aplikace mající svůj životní cyklus
CRUD	– Create Read Update Delete
DAO	– Data Access Object
DTO	– Data Transfer Object
IDE	– Integrated Development Environment
JDK	– Java Development Kit
JVM	– Java Virtual Machine
OS	– operační systém
REST	– Representational State Transfer
SDK	– Software Development Kit
SMS	– Short Message Service
s.r.o.	– společnost s ručením omezeným
VCS	– Version Control System



## Seznam obrázků

1	Aplikace sRecepty v různých jazycích . . . . .	15
2	Kontrola připojení v akci . . . . .	17
3	Funkce nákupního seznamu . . . . .	19

## Seznam výpisů zdrojového kódu

1	Ukázka struktury souboru s překlady textů . . . . .	16
2	Přístup k řetězcům definovaných v XML souborech . . . . .	16
3	Změna jazyka aplikace oproti systémovému jazyku . . . . .	16
4	Způsob získání seznamu ingrediencí patřící pod daný recept z lokální databáze .	18

# 1 Úvod

Tématem mé bakalářské práce bylo absolvování individuální odborné praxe ve firmě Railsformers s.r.o. jako vývojář mobilních aplikací pro operační systém Android. Tuto alternativu bakalářské práce jsem si vybral hlavně z důvodu, že jsem se chtěl naučit tvořit aplikace na OS Android, naučit se pracovat v týmu a zjistit jaký je reálný pracovní postup od zadání zakázky po hotový produkt.

Po celou dobu praxe jsem pracoval na mobilní mnohojazyčné aplikaci s názvem sRecepty, která zprostředkovává data z webů [www.srecepty.cz](http://www.srecepty.cz), [www.srecepty.pl](http://www.srecepty.pl), [www.srecipes.eu](http://www.srecipes.eu) a mnoha dalších spadající pod firmu General recipe s.r.o. Obsahuje tedy materiál s tématem vaření, zdravého životního stylu a všeho okolo[3]. Zároveň slouží částečně i jako sociální síť, díky tomu, že uživatelé mohou reagovat na přidaný obsah i na sebe navzájem. Také se uživatelé mohou přímo podílet na obsáhlosti jednotlivých receptů, jelikož mají možnost přímo z aplikace vyfotit a nahrát hotový recept.

## 2 Zaměření firmy a pracovní zařazení studenta

### 2.1 O firmě

Railsformers s.r.o se již mnoho let zabývá vývojem internetových aplikací a informačních systémů. Jejich vývojáři se zaměřují na různé projekty v prostředí internetu. Budují sociální sítě, komunitní portály, řídicí a různé komplexní systémy. Mezi jejich portfolio patří portál o vaření a zdravém životním stylu sRecepty.cz, internetová aplikace na správu osobních financí sMoneybox nebo jejich hlavní projekt Hedurio, což je inovativní internetová aplikace pro řízení bezpečnostních agentur[4]. Pro vývoj backendu všech jejich současných projektů využívají jejich vývojáři framework Ruby on Rails a odtud pochází i název firmy.

### 2.2 Pracovní zařazení studenta a pracovní náplň

Ve firmě jsem pracoval jako Android vývojář v týmu spolu s jedním dalším vývojářem pro Android pod vedením zkušeného vývojáře serverových aplikací, který pro nás tvořil REST API.

Náplní práce bylo s týmem vytvořit mobilní aplikaci pro OS Android, která bude zobrazovat a zpracovávat data z webového API z portálu sRecepty.cz a jeho jazykových mutací.

### 2.3 Pracovní prostředí

Railsformers s.r.o. sídlí ve Vědecko-technologickém parku Ostrava a také zde má své kanceláře, kde jsem po většinu absolvování praxe pracoval. Zbytek praxe jsem pracoval z domu, ať už to byla neodkladná oprava chyby či samostudium. Celou praxi jsem pracoval na svém vlastním přenosném počítači.

Komunikace ve firmě byla převážně osobního charakteru, ale pokud nebyl možný osobní kontakt, komunikačním prostředkem se stal e-mail, služba Hangouts nebo přímo portál projektového řízení, kde byly zadávány úkoly.

## 3 Použité technologie a nástroje

Praxi by nebylo možno absolvovat bez níže popsaných technologií a nástrojů, proto je zde uveden alespoň jejich krátký popis.

### 3.1 Java

Java je programovací a počítačová platforma, která se v dnešní době těší veliké oblibě díky své jednoduchosti, bezpečnosti a spolehlivosti. Poprvé byla Java vydána v roce 1995 společností Sun Microsystems, Inc. Mezi největší výhody Javy bezesporu patří, že programy v něm vytvořené jsou multiplatformní. To znamená že aplikace napsaná v Javě může být spuštěna na libovolné platformě s nainstalovaným JVM. Další výhodou je například to, že se jedná o objektově orientovaný jazyk nebo že má opravdu široké použití od desktopových, přes serverové, až po mobilní aplikace[5].

### 3.2 Android SDK

Android software development kit je sada vývojových nástrojů umožňující vytváření aplikací pro platformu Android. Android SDK obsahuje ukázkové projekty se zdrojovým kódem, vývojové nástroje, emulátor a požadované knihovny pro tvorbu aplikací na Android. Aplikace jsou napsány pomocí programovacího jazyka Java.

### 3.3 Android Studio

Android Studio je vývojové prostředí založené na IDE IntelliJ IDEA a je speciálně vytvořeno pro potřeby vývoje aplikací na OS Android[6]. Do vydání tohoto IDE v roce 2013 neexistovalo žádné vývojové prostředí, které by bylo primárně určeno pro tyto účely. Vydáním Android Studia firma Google, inc. podpořila Android vývojáře po celém světě a tím podpořila rozmach svého operačního systému na mobilním trhu.

### 3.4 Git

Git je systém pro správu verzí, který vytvořil Linus Torvalds v roce 2005 pro potřeby vývoje linuxového jádra. Tento systém je cílen na rychlost, podporu nelineárního vývoje a schopnost spravovat jak malé, tak opravdu velké projekty. Nejmenší jednotkou systému je commit, tedy revize, a ta by měla v ideálním případě obsahovat jednotlivou logickou část práce.

## 4 Zadané úkoly a jejich řešení

Všechny náležitosti projektu byly předány ve formě projektového řízení přes webovou službu [www.projektove.cz](http://www.projektove.cz) a byly osobně prodiskutovány. Náš tým musel mobilní aplikaci sRecepty vytvořit „od nuly“, do začátku praxe existovalo pouze webové rozhraní.

### 4.1 Příprava

Nezbytným krokem před zahájením práce bylo nainstalovat a zprovoznit na svém přenosném počítači všechny potřebné technologie pro vývoj nativní aplikace pro OS Android. V první řadě jsem musel z oficiálních stránek firmy Oracle stáhnout a nainstalovat JDK 8. Jako IDE jsem zvolil Android Studio, jelikož je oficiálním Android IDE a měl jsem již kladnou zkušenost s vývojovým prostředím IntelliJ IDEA, na které je toto prostředí postaveno.

Veškerá zhotovená práce měla být odesílána na firemní Git, což je VCS. Konkrétně má firma na svém serveru nástroj GitLab, který kromě verzovacího systému nabízí také sledování problémů, wiki a jiné užitečné utility. Proto bylo dalším úkolem zprovoznit na pracovním počítači Git, připojit jej na firemní server a naučit se používat alespoň jeho základní příkazy jak jsou `add`, `commit`, `push` a další[1].

Z důvodu testování celé aplikace na mobilním zařízení, jsem nainstaloval na pracovní počítač i potřebné ovladače.

Když bylo vše potřebné připravené, mohl jsem se začít učit Android SDK. Zkoušel jsem si tutoriály, pročítal dokumentaci a studoval z knihy Vývoj aplikací pro Android[2]. Studium probíhalo po celou dobu praxe, vždy se našel nový problém, který bylo třeba vyřešit novým způsobem. Během zimního ročníku jsem byl také ve škole zapsán do předmětu Tvorba aplikací pro mobilní zařízení II, kde jsem se naučil mnoho užitečného.

Jelikož jsme v týmu neměli žádného grafika, tak jsme si všechny grafické návrhy museli udělat sami. Aby aplikace vypadala moderně, bylo také potřeba nastudovat základy Material designu.

### 4.2 Struktura aplikace

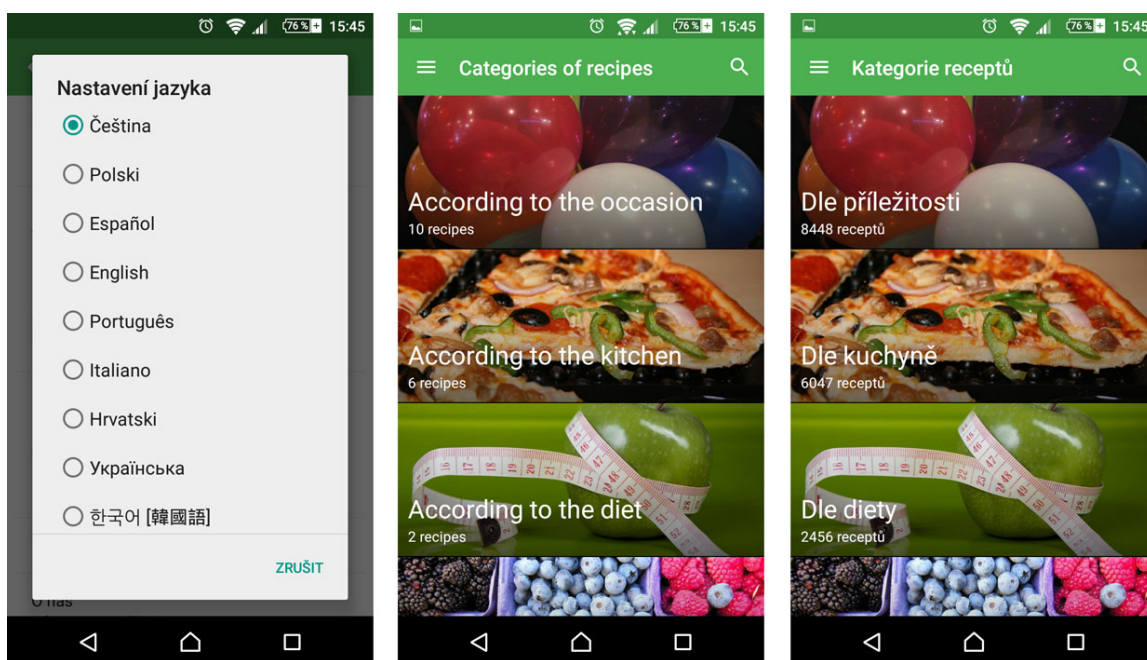
Po naprogramování několika aktivit bylo zjištěno, že budeme potřebovat určitou strukturu abychom se vyhnuli přílišnému kopírování vlastních kódů do různých aktivit a dalším problémům, ale i pro zjednodušení dalších opakujících se činností. Dostal jsem za úkol vymyslet architekturu, která by toto efektivně řešila.

V první řadě jsem vytvořil vlastní třídu, která dědí z třídy `Application` a v níž byl použit návrhový vzor Singleton. Jelikož OS Android udržuje instanci této třídy po celou dobu běhu naší aplikace, je to ideální způsob, jak se vypořádat s činnostmi jako zjištění, zda je zařízení připojeno k internetu, zda se jedná o tablet a spoustu dalších pomocných metod. Především zde ale byla umístěna i inicializace služby Google Analytics, na kterou bylo z aktivit a fragmentů odkazováno právě přes tuto třídu.

Další důležitou částí tohoto úkolu bylo napsání takzvané `BaseActivity`. Tato třída především dědila z `AppCompatActivity` a byla tak základním stavebním prvkem v každé naší aktivitě. Umístil jsem do ní nastavení jazyka a stažení všech domén při prvním spuštění nebo také většinu implementace kontroly připojení. Tyto a další funkce aplikace popíšu podrobněji v dalších kapitolách.

### 4.3 Multijazyčnost aplikace

Jelikož je aplikace klientem pro webovou službu v několika jazykových mutacích na několika doménách, bylo potřeba zajistit multijazyčnost celé aplikace.



Obrázek 1: Aplikace sRecepty v různých jazycích

#### 4.3.1 Překlad definovaných textových řetězců

Mým úkolem bylo zajistit aby každý textový řetězec, který se v aplikaci zobrazuje uživateli byl nahrazen řetězcem v jeho nastaveném jazyce.

Aplikace měla v době absolvování praxe 11 jazykových mutací, tím pádem v projektu existuje 11 souborů s názvem `strings.xml`. Každý z těchto souborů je umístěn ve složce podle jazyka. Například soubor s českým překladem aplikace se nachází ve složce `values-cs`. Výchozím jazykem byla zvolena angličtina a soubor s těmito řetězci byl uložen do složky `values`.

Struktura těchto souborů je v každé složce totožná. Každý identifikátor řetězce je vždy v každém z 11 souborů. Jediné co se mění je vždy hodnota párové značky obsahující atribut s tímto identifikátorem. V aplikaci bylo na konci praxe 139 textových řetězců a téměř všechny byly přeloženy pomocí nástroje Google Překladač.

---

```
<resources>
  <string name="app_name">sRecepty</string>
</resources>
```

---

Výpis 1: Ukázka struktury souboru s překlady textů

V XML kódech jsem na tyto textové řetězce odkazoval přes referenci `@string/` a za lomítko byl umístěn identifikátor řetězce. Systém Android sám při běhu aplikace zjistí který jazyk je v aplikaci nastaven a doplní na místo reference hodnotu podle identifikátoru ze složky, která má v cestě kód jazyka jak jsem zmínil výše. K hodnotám řetězců podle jazyka jsem samozřejmě musel přistupovat i z Java kódu. Tento přístup je znázorněn v následujícím výpisu.

---

```
String appName = getString(R.string.app_name);
```

---

Výpis 2: Přístup k řetězcům definovaných v XML souborech

### 4.3.2 Stahování dat z různých domén podle jazyka

Jelikož měla tato aplikace 11 jazykových mutací, bylo potřeba zajistit dynamickou změnu domény, na kterou se má aplikace dotazovat při získávání či odesílání dat. Domén bylo v době psaní této práce 10 a jmenovitě to byla doména evropská (anglická), česká, německá, španělská, chorvatská, italská, korejská, polská, portugalská a ukrajinská. Slovenská jazyková mutace aplikace odkazuje na českou doménu.

Mým úkolem bylo zajistit aby bylo možné kdykoliv bez změny aplikace přidat ze serveru novou doménu a případně do aplikace pouze přidat soubor s novým překladem. Tato funkcionality byla zajištěna tak, že při úplně prvním spuštění aplikace se z výchozí evropské domény `www.srecipes.eu` stáhnou JSON data obsahující seznam všech domén a pro každou doménu její název, název jazyka, kód jazyka a url. Poté se hledá shoda mezi kódem systémového jazyka zařízení na kterém aplikace běží a kódem jazyka ze staženého seznamu. Jestliže se shoda nenalezne, zvolí se data evropské domény, jinak se zvolí doména, kde proběhla shoda. Dále se data této domény uloží do `SharedPreferences`, což je lokální úložiště naší aplikace v zařízení.

Protože se systémový jazyk může lišit od jazyka naší aplikace, musel jsem také při každém startu aplikace změnit konfiguraci jazyka pomocí `updateConfiguration`. Změna jazyka byla uživatelům umožněna v nastavení aplikace viz obrázek 1.

---

```
protected void setupLanguage() {
    String currentLangCode == this.sharedPrefs.getString(Config.LOCALE_ID, null);
    if (currentLangCode == null) {
        currentLangCode = Locale.getDefault().getLanguage();
        SharedPreferences.Editor editor = sharedPrefs.edit();
        editor.putString(Config.LOCALE_ID, currentLangCode).apply();
    }
}
```



```

}
Locale locale = new Locale(currentLangCode);
Locale.setDefault(locale);
Configuration config = new Configuration();
config.locale = locale;
getBaseContext().getResources().updateConfiguration(config,
    getBaseContext().getResources().getDisplayMetrics());
}

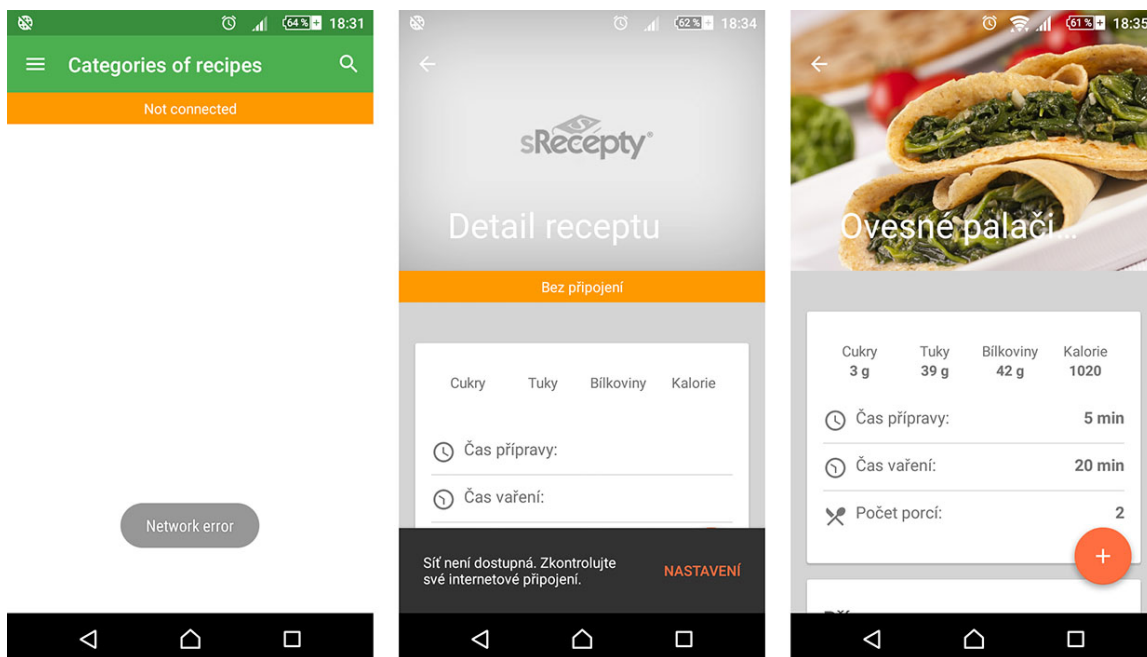
```

Výpis 3: Změna jazyka aplikace oproti systémovému jazyku

#### 4.4 Kontrola připojení k internetu

Dalším mým neméně důležitým úkolem bylo implementovat kontrolu připojení k internetu, jelikož aplikace stahuje téměř všechna data z internetu a až na několik funkcí se bez něj neobejde.

Pokud není zařízení připojeno k internetu, uživatel je na to upozorněn oznamovací lištou umístěnou pod Toolbarem nebo vizuálním prvkem zvaným **Snackbar** s akcí otevření nastavení připojení systému. Při opětovném připojení to aplikace sama rozezná a bez zásahu uživatele načte potřebná data.



Obrázek 2: Kontrola připojení v akci

K této funkčnosti nestačilo zkontrolování připojení při startu aktivity, ale musel jsem napsat vlastní třídu dědicí z třídy **BroadcastReceiver**, který reagoval na systémová volání služby

CONNECTIVITY\_SERVICE. Tento přijímač byl aktivován a deaktivován podle potřeby ve třídě `BaseActivity`, kterou jsem již popsal v jedné z předchozích kapitol.

Layout každé aktivity zobrazující upozornění o nedostupné síti pak zahrnuje i layout s oznamovací lištou, přičemž má tento layout nastavenou viditelnost na neviditelné. Každá taková aktivita pak posílá referenci na toto `View` jako parametr metody `setupNetworkChange`, ve které jsem napsal mechanismus, který zobrazuje a skrývá tuto lištu.

Metodu `setupNetworkChange` jsem umístil do třídy `BaseActivity` a stará se o zobrazení či skrytí oznamovací lišty. Zároveň, pokud je zařízení připojeno, kód zjišťuje zda jsou již data aktuální aktivity stažena či nikoliv. V případě, že stažena nejsou, dá o této skutečnosti metoda vědět aktuální aktivitě a ta provede prvotní načtení potřebných dat.

V případě, že uživatel poprvé spustil aplikaci bez internetové konektivity, znamená to, že nemá ani staženy domény. Proto jsem musel ošetřit i tento problém. Vyřešil jsem to tak, že když tato situace nastane, celá aplikace je restartována a tudíž už bude vše nastaveno a připraveno k dalšímu použití.

## 4.5 Nákupní seznam

Jednou z mála funkcionalit aplikace, která se obejde bez připojení k internetu, je nákupní seznam. Nákupní seznam byl vytvořen z toho důvodu, aby si uživatelé mohli uložit ingredience, které potřebují k danému receptu a jít pohodlně do obchodu, kde si tyto ingredience nakoupí.

### 4.5.1 Uložení seznamu do zařízení

Protože existuje mnoho uživatelů, kteří nemají k dispozici mobilní internet, bylo vhodné aby byla tato funkce naimplementována jako offline. Po zvážení několika různých způsobů uložení seznamu jsem se rozhodl ukládat data do SQLite databáze. K uložení seznamu stačily pouze dvě tabulky o několika attributech. Proto jsem si zvolil cestu, kdy jsem si tabulky vytvořil a spravoval „ručně“.

Pro uložení do SQLite databáze v zařízení byla vytvořena pomocná třída, která rozšiřuje třídu `SQLiteOpenHelper`. Tato pomocná třída poskytuje jednodušší způsob ukládání dat. Obsahuje základní CRUD operace a další jako například získání všech položek nebo počet všech položek vyhovující zadaným parametrům.

Dále jsem zkombinoval výhody návrhových vzorů jako jsou `Továrna`, `DAO` a `DTO` abych zajistil co nejjednodušší přístup k databázi přímo z aktivity. Při manipulaci se seznamem se změny projeví do databáze okamžitě.

---

```
List<IngredientDTO> ingredientsDTO = ingredientDAO.getAll(recipeDTO);
```

---

Výpis 4: Způsob získání seznamu ingrediencí patřící pod daný recept z lokální databáze

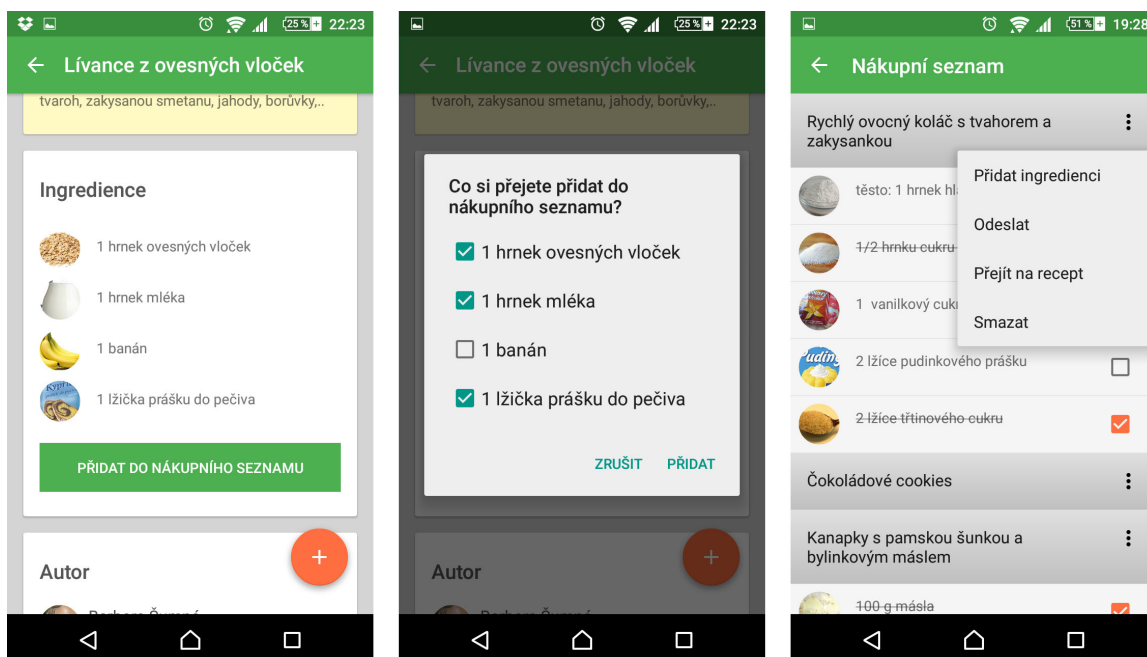
#### 4.5.2 Ovládání a funkce nákupního seznamu

V aktivitě nákupního seznamu se zobrazuje seznam všech receptů jejichž ingredience uživatel přidal. Po rozbalení receptu se zobrazí jednotlivé ingredience. Každá ingredience má vlastní zaškrťovací políčko, které signalizuje zda již tuto ingredienci uživatel má nebo ne.

Zobrazení dynamického seznamu uvnitř jiného dynamického seznamu bylo trochu problematické, ale brzy jsem našel knihovnu, která toto podporovala.

Přidat položku lze dvěma způsoby. Přímou z detailu receptu z karty Ingredience nebo má uživatel možnost přidat vlastní položky pod konkrétní recept přes kontextovou nabídku.

Smazat můžeme ze seznamu celý recept včetně všech jeho položek nebo při dlouhém stisku na ingredienci se zobrazí dialogové okno, které se uživatele ptá, zda chce tuto položku smazat.



Obrázek 3: Funkce nákupního seznamu

Také jsem v seznamu implementoval funkci sdílení seznamu ingrediencí. Tato akce je taktéž dostupná z kontextové nabídky vyvolané kliknutím na tři tečky vedle názvu receptu. Seznam se sdílí jako prostý text, kde byl místo zaškrťovacích políček zvolen buď plný nebo prázdný čtvereček. Díky tomu může uživatel odeslat celý seznam ingrediencí receptu třeba prostřednictvím SMS jiné osobě, která tuto aplikaci vůbec nebude potřebovat. Ze seznamu aplikací, přes které můžeme recept sdílet bylo nutné odstranit aplikaci Facebook, neboť se chová nestandardně.

Poslední zajímavou funkcí seznamu je možnost přejít na konkrétní recept. Tato funkce je taktéž ukryta v kontextové nabídce. Díky této funkci bylo k receptu nutné uložit i doménu, ze které byl recept přidán a zároveň vždy zobrazovat jen seznamy, které byly přidány v aktuálně nastaveném jazyce.

## 4.6 Uživatelské nastavení aplikace

Pro spokojenost všech uživatelů jsem implementoval také uživatelské nastavení. Byly tam vytvořeny tři skupiny nastavení, jmenovitě „Účet uživatele“, „Nastavení zpráv“ a „O aplikaci“.

Nastavení je vytvořeno jako `PreferenceFragment` a díky tomu se ve většině případů ani nemusím starat o uložení volby uživatele. Tato třída to za mne dělá sama když má položka nastavení přiřazený atribut `key`. Tento klíč je identifikátorem položky v `SharedPreferences`.

### 4.6.1 Účet uživatele

V této skupině nastavení se uživateli zobrazují vždy dvě položky. Reálně jsou tam však 3, protože na prvním řádku má uživatel možnost se přihlásit nebo odhlásit a zobrazuje se vždy jen aktuální možnost. Když je uživatel odhlášen, může se tudy dostat na obrazovku přihlášení. A naopak, když je přihlášen, má možnost se zde odhlásit.

Druhou položkou skupiny je již zmíněná změna jazyka aplikace. Tato položka je vytvořena jako `ListPreference` a při tvorbě tohoto nastavení jsem musel vyřešit ten problém, že jednotlivé položky seznamu jazyků se nejdříve musí stáhnout ze serveru aby byly aktuální. Když uživatel vyvolal nabídku jazyků před stažením dat, aplikace spadla. Vyřešil jsem to tak, že při startu aktivity nastavení jsem tuto položku nastavil jako zašedlou a nereagující, asynchronně stáhl data s doménami a jazyky a až poté změnu jazyka odblokoval.

### 4.6.2 Nastavení zpráv

V aplikaci je také implementace přijímání push notifikací (zpráv) odeslaných ze serveru. Pro tuto funkci jsem v nastavení vytvořil vlastní skupinu o dvou položkách. První položka je explicitní intent odkazující na centrum přijatých zpráv.

Druhá položka skupiny je přepínač, kdy má uživatel možnost vypnout či zapnout přijímání push zpráv na toto zařízení. Uživatel může používat s jedním účtem více zařízení a pomocí této nabídky rozlišit na jaká zařízení chce zprávy přijímat a na která nikoliv.

### 4.6.3 O aplikaci

Poslední skupina nastavení je především informativní. Obsahuje tři položky přičemž první z nich je implicitní intent na webový prohlížeč, kdy se uživateli otevře webové rozhraní služby sRecepty. Dále je zde položka zobrazující provozní podmínky. Poslední položka otevře okno s informacemi o službě sRecepty a aktuální verzi aplikace.

## 5 Znalosti a dovednosti uplatněné v průběhu praxe

Při absolvování praxe jsem využil znalosti objektivě orientovaného programování, které jsem se naučil především v předmětech Algoritmy I a II a v předmětech Programování I a II. Také jsem ocenil, že jsem měl dobré základy programování v jazyce Java z předmětu Programovací jazyky I.

Dále jsem řešil návrh, implementaci a komunikaci s databází, přičemž jsem uplatnil znalosti nabyté v předmětech Úvod do databázových systémů a Databázové a informační systémy.

Jelikož jsem programoval aplikaci na OS Android, byl pro mne obrovským přínosem předmět Tvorba aplikací pro mobilní zařízení II, který jsem absolvoval až během praxe. Proto mne mrzelo, že jsem ho neměl dřív, protože by mi to na praxi ušetřilo mnoho času.

Na praxi jsem také navrhoval uživatelské rozhraní, kdy jsem čerpal z předmětu Uživatelská rozhraní. Využil jsem také návrhové vzory naučené v předmětu Vývoj informačních systémů.

Nezanedbatelnou část znalostí uplatněných na praxi jsem však musel nabýt samostudiem. To ale vnímám jako velký přínos.

## 6 Znalosti a dovednosti scházející v průběhu praxe

Co mi opravdu na praxi chybělo byla znalost systému Git či jiného verzovacího systému. Až na praxi jsem si uvědomil jak silný a nepostradatelný je tento nástroj a na škole bych určitě přivítal předmět „Verzovací systémy“. Tuto scházející znalost jsem však doplnil samostudiem.

## 7 Výsledky absolvované praxe a její zhodnocení

Produktem mé praxe je Android aplikace, kterou si kdokoliv může stáhnout a nainstalovat z Google Play obchodu. Aplikace je zcela zdarma a v současné době (během dokončování této práce) ji má na svých zařízeních nainstalováno přes 3 300 lidí. Zároveň je docela oblíbená, to dokazuje i fakt, že při hledání fráze „recepty“, v obchodě Google Play, se aplikace aktuálně zobrazí hned na druhém místě. Dlouhodobě si aplikace drží velmi pozitivní hodnocení.

Jsem moc rád za to, že jsem se tohoto projektu mohl účastnit. Během praxe jsem také musel najít a opravit mnoho chyb, kterým se v budoucnu mohu rovnou vyhýbat. Celou praxi tedy hodnotím jako velké pozitivum a ucelení dovedností, které jsem nabýval po celé tři roky bakalářského studia.

## 8 Závěr

Hlavním cílem této práce bylo popsat činnosti, které jsem vykonával na 50 denní odborné praxi u firmy Railsformers s.r.o. V této práci jsem zmínil 6 nejzajímavějších úloh, které jsem dostal zadány a vyřešil je. V posledních kapitolách popisuji uplatněné vědomosti z univerzitního prostředí, chybějící znalosti a nově nabyté dovednosti z této praxe.

V prvních měsících šla práce pozvolna, ale čím více jsem měl zkušeností, tím rychleji jsem dokázal zpracovávat jednotlivé úkoly. Již na konci října byla vydána první verze naší aplikace, kterou jsem do konce zimního semestru ještě ladil. Před každým vydáním nové verze jsem aplikaci intenzivně testoval.

V letním semestru jsem dostal zadány úkoly s novými funkcemi, které jsem měl zapracovat do aplikace. Požadavky byly postupně splněny a implementovány do aplikace. Druhá majoritní aktualizace nese označení 2 a byla vydána na konci letního semestru.

Absolvování této individuální odborné praxe bylo pro mne velkým přínosem. Naučil jsem se spoustu nových vědomostí, poznal jsem týmovou práci a zjistil jaký je průběh reálného projektu od nápadu až po zpětnou vazbu od zákazníka používajícího výsledný produkt. Navíc mne na této praxi začal vývoj aplikací pro OS Android bavit a věřím, že se tomuto směru budu věnovat i ve své budoucí profesní kariéře.



## Literatura

- [1] Git: Dokumentace [online]. [cit. 2016-04-21]. Dostupné z: <https://git-scm.com/docs>
- [2] LACKO, Luboslav. Vývoj aplikací pro Android. 1. vyd. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.
- [3] General recipe s.r.o.: Sites about cooking [online]. [cit. 2016-04-21]. Dostupné z: <http://generalrecipe.com/>
- [4] Railsformers s.r.o. [online]. [cit. 2016-04-21]. Dostupné z: <http://railsformers.com/>
- [5] Java [online]. [cit. 2016-04-21]. Dostupné z: <https://www.java.com/en/about/>
- [6] Android Studio [online]. [cit. 2016-04-21]. Dostupné z: <http://developer.android.com/sdk/index.html>