

Gymnázium, Praha 6, Arabská 14



Předmět Programování, vyučující Mgr. Jan Lána

Dokumentace ročníkového projektu

Zupl

Tomáš Černý

I.E 2016/2017

Květen 2017



Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

---

## Anotace

Tento ročníkový projekt má za cíl zjednodušit a zrychlit získávání rozvrhu na gymnáziu Arabská bez nutnosti přihlašovacích údajů do systému Bakaláři. Aplikace je pro mobilní zařízení s operačním systémem Android 4.1 Jelly Bean a vyšší. Součástí projektu je i volba skupin předmětů a aplikace informuje uživatele, když se objeví nové suplování a jaká je následující hodina včetně jejího umístěním ve škole. To se hodí zejména pro nové studenty. Aplikace podporuje i vyučující.

Součástí projektu je i knihovna, která zprostředkovává rozvrhy a suplování, a je použitelná i v jiných projektech. Knihovna v současné době podporuje webové stránky rozvrhů a suplování gymnázia Arabská.

## Abstract

The objective of this project is to make gathering time table information on grammar school Arabská easier and faster without requiring of login to the Bakaláři system. The app is for mobile devices running Android 4.1 Jelly Bean and higher. It includes personalization of subject groups and the app informs the user when new substitutions are available and what is the next lesson including its location in the school. This is especially useful for new students. The app also supports teachers.

Part of this project is a library which provides time and substitution tables and is also applicable in other projects. The library currently supports the web pages of time tables and substitutions of grammar school Arabská.

## Zadání práce

- Vytvořit knihovnu, která bude zpracovávat rozvrhy a suplování pomocí knihovny JSOUP z webových stránek školy
- Vytvořit Android aplikaci, co využívá knihovnu rozvrhů a suplování, která bude
  - Umožňovat volbu předmětových skupin
  - Podporovat rozvrhy a suplování studentů i vyučujících
  - Upozorňovat uživatele na aktualizace suplování a před následující hodinou

## Úvod

Motivací pro tento projekt byly pomalé rozvrhy v mobilní aplikaci Bakaláři. Čekat více než pět vteřin na aktuální rozvrh není úplně ideální stav aplikace tohoto typu. Dále jsem se do své aplikace pokusil zahrnout funkce, které aplikaci Bakaláři určitě chybí, a které budou hojně využívány. Druhou motivací bylo se naučit vytvářet aplikace pro platformu Android.

V této práci nejdříve najdete odborné termíny, zkratky a vysvětlivky, nezbytné k pochopení některých kapitol, použité technologie, jaké části má vytvořený program a dále vysvětluji, jak tyto jednotlivé části fungují a jejich případné použití, klíčové problémy, které jsem řešil a uživatelskou příručku s detailním popisem uživatelského rozhraní.

# Obsah

Anotace .....	2
Abstract .....	2
Zadání práce .....	2
Úvod .....	2
Obsah .....	3
1 Odborné termíny .....	5
2 Použité technologie .....	5
2.1 Programovací, značkovací a objektově notační jazyky .....	5
2.1.1 Java .....	5
2.1.2 XML .....	5
2.1.3 JSON .....	5
2.2 Platforma Android .....	6
2.2.1 Běhová prostředí .....	6
2.3 Vývojová prostředí .....	6
2.3.1 IntelliJ IDEA .....	6
2.3.2 Android Studio .....	6
2.4 Externí knihovny .....	6
2.4.1 JSOUP .....	6
2.4.2 GSON .....	6
3 Architektura projektu .....	7
4 Jak používat Parser .....	7
4.1 Hierarchie tříd .....	7
4.1 Třída Lesson .....	7
4.2 Třída TableData .....	8
4.3 Třída SubsData .....	9
4.4 Třída Tablo .....	11
4.5 Třída TableTablo .....	12
4.6 Třída SubsTablo .....	12
5 Mobilní aplikace .....	13
5.1 Logo aplikace .....	13
5.2 Třídy aplikace .....	13
5.2.1 Hierarchie tříd .....	13
5.2.2 Třída MainActivity .....	14
5.2.3 Třída CustomPreferenceActivity .....	14
5.2.4 Třída GroupPickerActivity .....	14
5.2.5 Třída BootReceiver .....	15
5.2.6 Třída SubsService .....	15
5.2.7 Třída NotificationReceiver .....	15
5.3 Ukládání dat .....	15

5.3.1	Způsob ukládání .....	15
5.3.2	Pojmenovávání souborů .....	15
5.4	XML zdroje.....	16
6	Stěžejní problémy.....	16
6.1	Parsování dělených hodin.....	16
6.1.1	Popis problému .....	16
6.1.2	Nalezení správné buňky .....	16
6.1.3	Parsování informací z buňky .....	16
6.2	Aplikování změn na rozvrh.....	17
6.3	Přepínání mezi týdny .....	17
6.3.1	Výpočet dnů v týdnu .....	17
6.3.2	Přiřazení suplování .....	17
6.4	Porovnávání suplování .....	17
6.1	Laboratorní cvičení .....	18
7	Příručka pro uživatele .....	19
7.1	Instalace a nastavení zařízení .....	19
7.2	Uživatelské rozhraní.....	19
7.2.1	Hlavní obrazovka .....	19
7.2.2	Hlavní nastavení .....	21
7.2.3	Volba skupin .....	21
7.1	Upozornění pro uživatele .....	22
7.1.1	Suplování .....	22
7.1.2	Před začátkem hodiny .....	22
	Závěr.....	22
	Přílohy .....	23
	Seznam použité literatury a zdrojů .....	23
	V tomto dokumentu .....	23
	Seznam diagramů, obrázků, tabulek a ukázek kódu .....	23

## 1 Odborné termíny

ART	Android Runtime – Běhové prostředí Androidu.
AoT compilation	Ahead of Time compilation – Před-kompilace. Program se zkompiluje již před spuštěním. <sup>1</sup>
Parsovat	Rozdělit celek informací na jednotlivé.
Parser	Program nebo část programu, která parsuje.

### Základní části Android aplikace

Aktivita (angl. Activity)	Reprezentuje v Android aplikaci jednu obrazovku, se kterou může uživatel interagovat a přecházet z ní do jiných aktivit nebo aplikací. <sup>2</sup>
Servis (angl. Service)	Servis zajišťuje déle běžící operace na pozadí a nemá uživatelské rozhraní. <sup>3</sup>
Přijímač (angl. BroadcastReceiver)	Přijímač je objekt, který vyčkává v pozadí na určitou událost zařízení a po jejím volání se spustí. <sup>4</sup>

## 2 Použité technologie

### 2.1 Programovací, značkovací a objektově notační jazyky

#### 2.1.1 Java

Java je velice populární a silně objektově orientovaný programovací jazyk. Patří do rodiny jazyků C a v současné době ji vyvíjí společnost Oracle. Java je dostupná pod všeobecnou veřejnou licencí GNU. V tomto projektu je to dominantní jazyk.

#### 2.1.2 XML

XML (Extensible Markup Language – Rozšiřitelný značkovací jazyk) je značkovací jazyk pro uchovávání dat například dokumentů nebo GUI aplikací. <sup>5</sup> Jazyk vytvořila organizace W3C a je volně dostupný. V projektu jej používám pro tvorbu šablon GUI Android aplikace. A to jak celých obrazovek, tak jednotlivých prvků, jako jsou tabulky nebo nápisy, které se rozdělují do samostatných souborů.

#### 2.1.3 JSON

JSON (JavaScript Object Notation – JavaScriptový objektový zápis) je formát pro výměnu dat. Používá konvence rodiny programovacích jazyků C a je taktéž volně dostupný. <sup>6</sup> V projektu ho používám k ukládání objektů do interní paměti telefonu.

## 2.2 Platforma Android

Tento projekt je vyvinut pro platformu Android verze 4.1 Jelly Bean a vyšší.

### 2.2.1 Běhová prostředí

#### 2.2.1.1 Dalvik

Dalvik je běhové prostředí pro platformu Android do verze 4.4 (včetně). <sup>7</sup> V dalších verzích je již plně nahrazen ART.

#### 2.2.1.2 ART

ART je běhové prostředí, které nahrazuje Dalvik a je dostupné pro Android 4.4 a výše. Jeho výhodou je *AoT compilation*, která tak zvyšuje výkon Android zařízení. <sup>8</sup>

## 2.3 Vývojové prostředí

### 2.3.1 IntelliJ IDEA

IntelliJ IDEA je vývojové prostředí společnosti JetBrains, které podporuje programovací jazyky Java, PHP, Python a mnoho dalších. Program podporuje vývojářské nástroje jako je Git a debugování programu za chodu. Jeho Community Edition je volně dostupná pod licencí Apache 2.

### 2.3.2 Android Studio

Android Studio je ve své podstatě IntelliJ IDEA s množstvím pluginů a nástrojů pro vývoj aplikací pro platformu Android. Jedná se o spolupráci mezi JetBrains a Google a je v současné době, jako IntelliJ IDEA Community Edition, bezplatné.

## 2.4 Externí knihovny

### 2.4.1 JSOUP

JSOUP je knihovna pro jazyk Java, která umožňuje získávat obsah webových stránek ze značkovacího jazyka HTML. <sup>9</sup> Autorem knihovny je Jonathan Hedley a je dostupná pod licencí MIT. V projektu slouží jako prostředník mezi webovými stránkami školy a knihovnou rozvrhů a suplování.

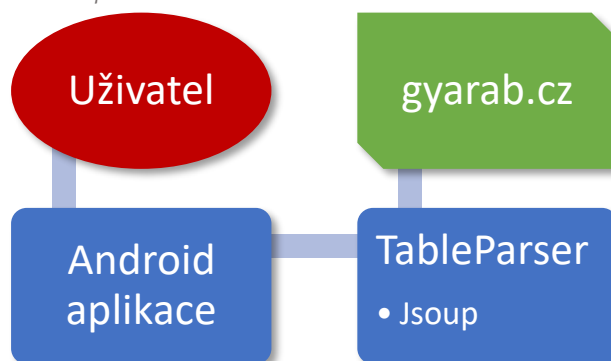
### 2.4.2 GSON

GSON je knihovna pro jazyk Java, která serializuje a deserializuje objekty z a do formátu JSON.<sup>10</sup> Autorem knihovny je společnost Google a je dostupná pod licencí Apache 2. V projektu ji používám na ukládání objektů ve formátu JSON.



### 3 Architektura projektu

Diagram 1: Softwarová architektura Zuplu

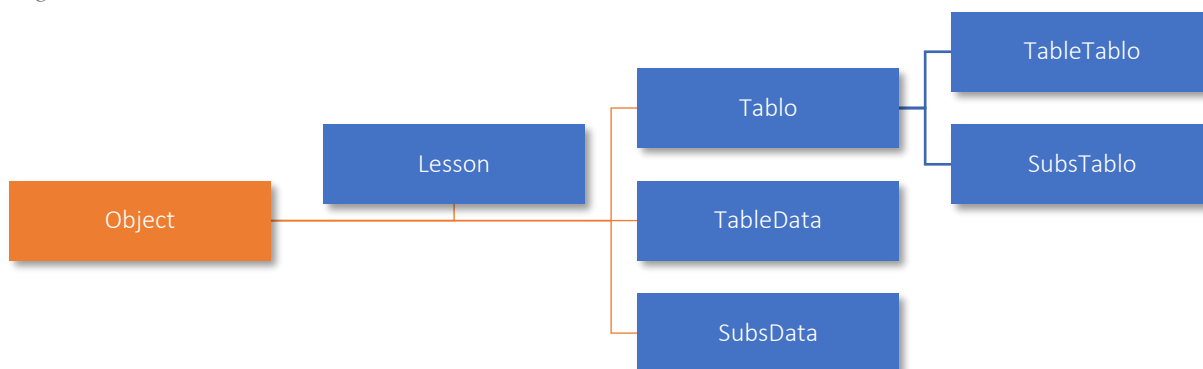


Uživatel pracuje s android aplikací, která vytváří objekty knihovny TableParser, která používá knihovnu JSOUP, a bere z nich informace, které pak prezentuje uživateli. Tyto informace získává TableParser z webových stránek gyarab.cz.

### 4 Jak používat Parser

#### 4.1 Hierarchie tříd

Diagram 2: Hierarchie tříd TableParseru



Pozn.: Oranžově je označena kořenová třída *Object*.

Třídy *TableData* a *SubsData* uchovávají různé informace, a tak není praktické, aby existoval jejich společný předek (kromě třídy *Object* samozřejmě). Všechny třídy implementují rozhraní *Serializable*.

#### 4.1 Třída Lesson

Třída *Lesson* uchovává jednu a více hodin. Pod jedním indexem *ArrayListů* je jedna hodina.

Tabulka 1: Konstruktory, proměnné a metody třídy *Lesson*

Konstruktory
<pre>public Lesson()</pre>

<b>Proměnné</b>
<code>protected ArrayList&lt;String&gt; subjects</code>
Předměty.
<code>protected ArrayList&lt;String&gt; teachers</code>
Učitelé.
<code>protected ArrayList&lt;String&gt; rooms</code>
Místnosti.
<code>protected ArrayList&lt;String&gt; groups</code>
Předmětové skupiny.
<code>protected ArrayList&lt;String&gt; origins</code>
Původy hodin. Původ hodiny může nabývat následujících hodnot: <i>odpadá</i> , <i>změna</i> , <i>spojí</i> , <i>přesun &gt;&gt;</i> , <i>přesun &lt;&lt;</i> , <i>výměna &gt;&gt;</i> , <i>výměna &lt;&lt;</i> , <i>default</i> – beze změny a <i>absent</i> – třída chybí.
<b>Metody</b>
<code>public void addEntry(String subject, String teacher, String room, String group, String origin)</code>
Přidá do každého <i>ArrayListu</i> dané argumenty.
<code>public void setEntry(int index, String subject, String teacher, String room, String group, String origin)</code>
Změní v každém <i>ArrayListu</i> hodnotu podle parametru na pozici <i>index</i> .
<code>public ArrayList&lt;String&gt; getSubjects()</code>
Vrací proměnnou <i>subjects</i> .
<code>public ArrayList&lt;String&gt; getTeachers()</code>
Vrací proměnnou <i>teachers</i> .
<code>public ArrayList&lt;String&gt; getRooms()</code>
Vrací proměnnou <i>rooms</i> .
<code>public ArrayList&lt;String&gt; getGroups()</code>
Vrací proměnnou <i>groups</i> .
<code>public ArrayList&lt;String&gt; getOrigins</code>
Vrací proměnnou <i>origins</i> .

## 4.2 Třída *TableData*

Třída *TableData* reprezentuje jeden rozvrh pro třídu, nebo pro učitele v jednom týdnu.

Tabulka 2: Konstruktory, proměnné a metody třídy *TableData*

<b>Konstruktory</b>
<code>public TableData()</code>
<b>Proměnné</b>
<code>protected String owner</code>
Majitel rozvrhu. Např. „1.E“.
<code>protected String mentor</code>
Třídní učitel. V případě rozvrhů vyučujících je stejný jako majitel rozvrhu.
<code>protected String[] times</code>
Časy začátku a konce hodin. Např. pro nultou hodinu „7:00 -7:45“
<code>protected Lesson[][] lessons</code>
Hodiny v jednom týdnu. První pole jsou jednotlivé dny a druhé jsou hodiny v jednom dni.
<b>Metody</b>
<code>public String getOwner()</code>
Vrací proměnnou owner.
<code>public void setOwner(String owner)</code>
Změní proměnnou owner na daný parameter.
<code>public String getMentor()</code>
Vrací proměnnou mentor.
<code>public void setMentor(String mentor)</code>
Změní proměnnou mentor na daný parameter.
<code>public String[] getTimes()</code>
Vrací proměnnou times.
<code>public void setTimes(String[] times)</code>
Změní proměnnou times na daný parameter.
<code>public Lesson[][] getLesson()</code>
Vrací proměnnou lessons.
<code>public void setLessons(Lesson[][] lessons)</code>
Změní proměnnou lessons na daný parameter.

## 4.3 Třída *SubsData*

Třída *SubsData* reprezentuje suplování v jednom dni žáků i vyučujících.

Tabulka 3: Konstruktory, proměnné a metody třídy SubsData

<b>Konstruktory</b>
<code>public SubsData()</code>
<b>Proměnné</b>
<code>protected Date date</code>
Datum platnosti rozvrhu.
<code>protected Date dateAdded</code>
Datum přidání rozvrhu.
<code>protected HashMap&lt;String, Lesson[]&gt; absentClasses</code>
Tabulka nepřítomných tříd.
<code>protected HashMap&lt;String, Lesson[]&gt; absentTeachers</code>
Tabulka nepřítomných vyučujících.
<code>protected HashMap&lt;String, HashMap&lt;Integer, Lesson&gt;&gt; changesClasses</code>
Tabulka suplování žáků.
<code>protected HashMap&lt;String, HashMap&lt;Integer, Lesson&gt;&gt; changesTeachers</code>
Tabulka suplování vyučujících.
<b>Metody</b>
<code>public Date getDate()</code>
Vrací proměnnou date.
<code>public void setDate(Date date)</code>
Změní proměnnou date na daný parameter.
<code>public Date getDateAdded()</code>
Vrací proměnnou dateAdded.
<code>public void setDateAdded(Date dateAdded)</code>
Změní proměnnou dateAdded na daný parameter.
<code>public HashMap&lt;String, Lesson[]&gt; getAbsentClasses()</code>
Vrací proměnnou absentClasses.
<code>public HashMap&lt;String, Lesson[]&gt; getAbsentTeachers()</code>
Vrací proměnnou absentTeachers.
<code>public HashMap&lt;String, HashMap&lt;Integer, Lesson&gt;&gt; getChangesClasses</code>
Vrací proměnnou changesClasses.

```
public HashMap<String, HashMap<Integer, Lesson>> getChangesTeachers
```

Vrací proměnnou `changesTeachers`.

## 4.4 Třída *Tablo*

Třída *Tablo* stahuje HTML dokument, hledá vnořený HTML dokument a kontroluje, zdali bylo připojení úspěšné.

Tabulka 4: Konstruktory, proměnné a metody třídy *Tablo*

### Konstruktory

```
public Tablo(String address, String name)
```

Parametr `address` je adresa nadřazeného dokumentu a parametr `name` je *name* atribut *iframe* s vnořeným dokumentem.

### Proměnné

```
protected final String address
```

Adresa nadřazeného HTML dokumentu.

```
protected final String fileAddress
```

Adresa vnořeného HTML dokumentu.

```
protected Document doc
```

Vnořený HTML dokument. Objekt knihovny JSOUP (`org.jsoup.nodes.Document`)

```
protected boolean connectionSuccessful
```

Pravda pokud bylo spojení úspěšné.

### Metody

```
public Document getDoc()
```

Vrací proměnnou `doc`.

```
public void emptyDocument()
```

Nastaví hodnotu proměnné `doc` na *null*. Určené k ušetření místa při ukládání objektu.

```
public String getAddress()
```

Vrací proměnnou `address`.

```
public String getFileAddress()
```

Vrací proměnnou `fileAddress`.

```
public boolean isConnectionSuccessful()
```

Vrací proměnnou `connectionSuccessful`.

## 4.5 Třída TableTablo

Třída *TableTablo*, podtřída třídy *Tablo*, parsuje rozvrhy z HTML dokumentu.

Tabulka 5: Konstruktory, proměnné a metody třídy *TableTablo*

<b>Konstruktory</b>
<code>public TableTablo(String address)</code>
Parametr <code>address</code> je adresa nadřazeného dokumentu.
<code>public TableTablo(String address, String cls)</code>
Parametr <code>address</code> je adresa nadřazeného dokumentu a parametr <code>cls</code> je majitel rozvrhu, jehož rozvrh chceme. Není-li rozvrh v dokumentu, bude proměnná <code>tableDataArr</code> <i>null</i> .
<b>Proměnné</b>
<code>protected Elements tablos</code>
HTML kód jednotlivých rozvrhů. Objekt knihovny JSOUP. ( <code>org.jsoup.select.Elements</code> )
<code>protected TableData[] tableDataArr</code>
Rozvrhy.
<b>Metody</b>
<code>public Elements getTablos()</code>
Vrací proměnnou <code>tablos</code> .
<code>public TableData[] getTableDataArr()</code>
Vrací proměnnou <code>tableDataArr</code> .

## 4.6 Třída Substablo

Třída *Substablo*, podtřída třídy *Tablo*, parsuje suplování z HTML dokumentu.

Tabulka 6: Konstruktory, proměnné a metody třídy *Substablo*

<b>Konstruktory</b>
<code>public Substablo(String address, boolean absent)</code>
Parametr <code>address</code> je adresa dokumentu suplováními a parametr <code>absent</code> udává, zdali se mají parsovat i tabulky nepřítomných tříd a učitelů.
<b>Proměnné</b>
<code>protected ArrayList&lt;SubsData&gt; subsDataList</code>
Jednotlivá suplování na jeden den.
<code>public final boolean absent</code>
Předaný parametr <code>absent</code> .

Metody
<code>public ArrayList&lt;SubsData&gt; getSubsData ()</code>
Vrací proměnnou subsDataList.
<code>public static void applyChanges(Lesson[] lessons, HashMap&lt;Integer, Lesson&gt; changes, boolean teacher)</code>
Použije změny rozvrhu na normální rozvrh v jednom dni. Parametr <code>lessons</code> jsou obvyklé hodiny, parameter <code>changes</code> jsou změny a parameter <code>teacher</code> udává, zdali se jedná o učitele.

## 5 Mobilní aplikace

### 5.1 Logo aplikace

Logo aplikace je pravidelný osmiúhelník, jehož vnitřní rovnoramenné trojúhelníky jsou v odstínech šedi. Ve středu se nachází černé Z, jako Zupl. Android 7 v upozorněních zobrazuje pouze obrys bez barev, a proto je nutné vytvořit speciální ikonu.

Obrázek 1: Normální logo aplikace



Obrázek 2: Ikona upozornění pro Android 7



### 5.2 Třídy aplikace

#### 5.2.1 Hierarchie tříd

Diagram 3: Hierarchie aktivit aplikace

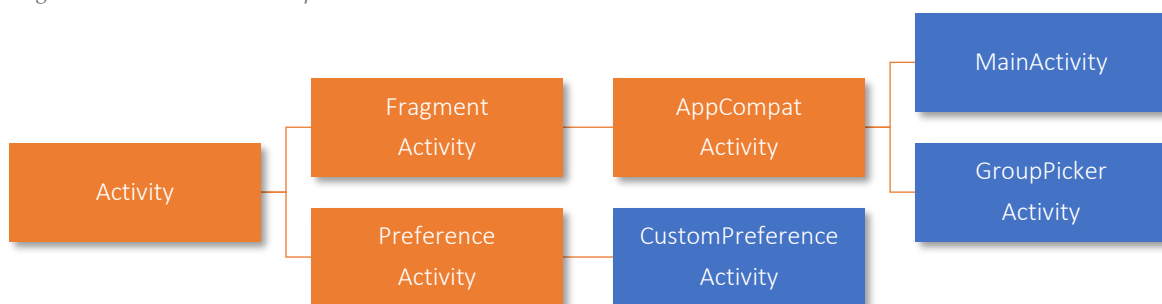


Diagram 4: Hierarchie přijímačů aplikace

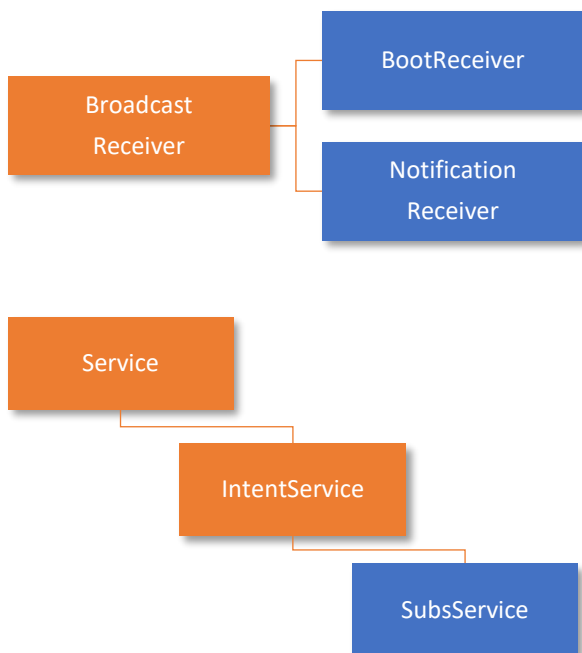


Diagram 5: Hierarchie servisů aplikace

Pozn.: Oranžově jsou označeny třídy Android API.

### 5.2.2 Třída MainActivity

Třída *MainActivity* prezentuje uživateli obrazovku aktuálního rozvrhu se suplováním podle nastavení a startuje pravidelnou aktualizaci suplování v intervalu jedné hodiny a upozornění na budoucí hodiny (metody třídy *SubService*), je-li aplikace řádně nastavena (je vybrán rozvrh a skupiny). Pokud tomu tak není, aplikace přenese uživatele do nastavení (třída *CustomPreferenceActivity*), kam se může přesunout i sám. Dále umožňuje přepínat mezi týdny.

Tato třída rozšiřuje třídu *AppCompatActivity*, která má podpůrnou knihovnu pro panel akcí, a přidává do horní části obrazovky tento panel.

### 5.2.3 Třída CustomPreferenceActivity

Třída *CustomPreferenceActivity* prezentuje uživateli obrazovku nastavení studenta nebo vyučujícího, skupin a zdali a kdy má aplikace informovat o následující hodině. Odtud se uživatel přesouvá do volby skupin (třída *GroupPickerActivity*) a tlačítkem zpět se vrací na hlavní obrazovku (třída *MainActivity*).

Tato třída rozšiřuje třídu *PreferenceActivity*, která umožňuje jednodušší vytváření nastavení a podporuje návrh nastavení v XML.

### 5.2.4 Třída GroupPickerActivity

Třída *GroupPickerActivity* prezentuje uživateli obrazovku nastavení skupin a seminářů podle vybrané třídy v nastavení (třída *CustomPreferenceActivity*), do kterého se může vrátit tlačítkem zpět. Zvolí-li uživatel první skupinu, vybere program i ostatní první skupiny. Tímto se snaží program zrychlit výběr a tato funkce se spustí pouze na první. Vybere-li



uživatel skupiny, povolí třídě *BootReceiver* se spouštět při zapnutí zařízení a spustí pravidelnou aktualizaci suplování a nastaví upozornění na budoucí hodiny (třída *SubsService*).

Tato třída rozšiřuje *AppCompatActivity*, ale na rozdíl od třídy *MainActivity* panel akcí nepoužívá.

### 5.2.5 Třída *BootReceiver*

Třída *BootReceiver* přijímá volání akce, která se volá po zapnutí zařízení, a zapíná v intervalu jedné hodiny aktualizaci suplování (třída *SubsService*). Není-li však aplikace řádně nastavena, *BootReceiver* se po zapnutí zařízení nespustí.

### 5.2.6 Třída *SubsService*

Třída *SubsService* aktualizuje suplování na pozadí a pokud je vydáno nové, nebo je nadcházející změněno, servis upozorní uživatele. Dále registruje alarmy, které zavolají *NotificationReceiver* v nastavený čas před vyučovací hodinou.

Tato třída rozšiřuje třídu *IntentService*, která na rozdíl od své nadřazené třídy *Service* nedělá práci na primárním vláknu, ale na jiném, a po skončení svého úkolu se sám vypne.

### 5.2.7 Třída *NotificationReceiver*

Třída *NotificationReceiver* vytvoří upozornění pro uživatele, když je zavolána.

## 5.3 Ukládání dat

### 5.3.1 Způsob ukládání

Nastavení přímo ovlivnitelné uživatelem (třída nebo vyučující, upozornění, skupiny) je uloženo v datových typech jako *String*, *boolean* a *Set<String>*, které jsou podporované Android API.

Chceme-li ale uložit vlastní objekt, Android API nám v tomto ohledu nepomůže. Proto projekt používá knihovnu GSON, která převede objekt do zápisu JSON a lze ho následně uložit jako *String*. Stejná knihovna je použita i pro opačný proces – získání objektu z JSON zápisu.

*Ukázka kódu 1: Objekt Lesson zapsaný v jazyce JSON*

```
{"subjects": ["PG", "AS"], "teachers": ["Lan", "Šil"], "rooms": ["(P1)",  
"(P2)"], "groups": ["PG2", "AS1"], "origins": ["default", "default"]}
```

### 5.3.2 Pojmenovávání souborů

Všechna nastavení vyjma suplování jsou uložena podle určitého klíče (např.: *notification\_time* pro čas notifikací před hodinou). Názvy souborů se suplováním jsou ale uloženy podle unixového času jejich platnosti.

## 5.4 XML zdroje

Každá obrazovka má svůj vlastní zdrojový soubor, stejně jako mnohokrát se vyskytující části těchto obrazovek, jako buňky rozvrhu, buňky a řádky výběru skupin, barvení okrajů prvků grafického rozhraní<sup>11</sup>, horizontální a vertikální oddělovače<sup>12</sup>. Kromě toho jsou v XML zdrojích texty (téměř všechny), barvy a menu.

## 6 Stěžejní problémy

### 6.1 Parsování dělených hodin

#### 6.1.1 Popis problému

Získávání informací z buněk hodin pro celou třídu je vcelku jednoduché, ale je-li hodina dělená, už to není tak přímočaré. Příčinou je struktura HTML tabulek, které rozvrhy zobrazují. Jednotlivé hodiny jsou umístěny pomocí tagů `<tr>`, kterých je pro jeden den různý počet. Velikost jednotlivých tagů `<td>` je určena pomocí atributu `rowspan`, jehož hodnota se odvíjí podle počtu dělených hodin v jednom časovém intervalu a podle počtu dělení i jiných hodin.

Budeme-li mít hodinu dělenou na dvě skupiny a další na tři, bude `rowspan` nedělených hodin šest, aby ve výsledné tabulce zaplnila celý prostor jednoho dne. Dělená hodina na dvě poloviny bude mít `rowspan` o hodnotě tři a hodina dělená na tři části bude mít `rowspan` dva.

Obrázek 3: Grafická reprezentace tabulky a korespondujících tagů řádků

1	<b>SMBG</b> (422) Hro (B11)	<b>SMBG</b> (422) Hro (B11)	<b>ČJ</b> Von (ZSV)	<b>AJ</b> (A1) Flo (A1)	<b>AJ</b> (A1) Flo (A1)	▶<tr>...</tr>
2						▶<tr>...</tr>
3	<b>SEP</b> (408) Hní (ZSV)	<b>SEP</b> (408) Hní (ZSV)				▶<tr>...</tr>
4						▶<tr>...</tr>
5	<b>SD20</b> (411) Hor (DĚ)	<b>SD20</b> (411) Hor (DĚ)		<b>AJ</b> (A2) Pra (A2)	<b>AJ</b> (A2) Pra (A2)	▶<tr>...</tr>
6						▶<tr>...</tr>

#### 6.1.2 Nalezení správné buňky

Číslo tagu `<tr>` se vypočítá jako součin `rowspanu` jedné dělené hodiny a počítadla, které jde od nuly do celkového počtu dělených hodin v tomto časovém intervalu a mínus jedna. V jednom `<tr>` může být hodin více. To řeší ukládání, jakou hodinu program už parsoval, do celočíselného pole `linePointer` o délce největšího `rowspanu` dne plus jedna.

#### 6.1.3 Parsování informací z buňky

Parsování informací z buňky zajišťuje metoda `parseSplitLesson`, která vezme nalezenou dělenou hodinu a podle její velikosti, od které se odvíjí styly vnitřních polí, se informace uloží do objektu `lesson`.

## 6.2 Aplikování změn na rozvrh

Občas je zápis jednotlivých suplování (předmět, vyučující, místnost, skupina a původ – odpadá atd.) v pořadí, ve kterém by bez určitých podmínek nebyl rozvrh přesný. Například je-li změna předmětu a původní odpadá (v tomto pořadí) nebo se jedna hodina přesune sem a jedna pryč (také v tomto pořadí), tak by se na rozvrhu ukázalo, že máme tuto hodinu volno, což neodpovídá opravdovému rozvrhu. Toto platí pouze pro rozvrhy a suplování tříd.

Nakonec se ukázalo následující řešení jako nejlépe fungující, které prochází každý řádek suplování a postupuje podle následujících podmínek:

1. Není-li skupina v hodině nalezena, přidej ji.
2. Neplatí-li podmínka 1 a je-li hodina nezměněná a
  - a. původ změny je *odpadá, přesun >>*, nebo *výměna >>*, změň původ
  - b. neplatí-li podmínka 2.a, změň vše na změnu
3. Neplatí-li podmínka 2 a 1 a původ změny není *odpadá, přesun >>*, ani *výměna >>*, změň vše na změnu

## 6.3 Přepínání mezi týdny

### 6.3.1 Výpočet dnů v týdnu

Program po stisknutí tlačítek posunutí týdnu vytvoří objekty jednotlivých dnů ve zvoleném týdnu.

*Ukázka kódu 2: Získání objektů dnů v týdnu*

```
Date now = new Date();
Date today = new Date(now.getYear(), now.getMonth(), now.getDate() + 7 * offset);
Date[] thisWeek = new Date[7];
int day = today.getDay();
for (int i = 0; i < thisWeek.length; i++) {
    thisWeek[i] = new Date(today.getYear(), today.getMonth(), today.getDate() + i - day);
}
```

Proměnná *offset* určuje relativně týden. Bude-li *offset* -1, posunou se všechny dny o 7 dní zpět.

### 6.3.2 Přirazení suplování

Program se následně podívá, zdali jsou shody mezi unixovými reprezentacemi dnů a názvy souborů suplování (vysvětleno v [Pojmenovávání souborů](#)). Pokud ano, podívá se, jestli existují změny pro uživatelem zvolenou třídu a následně zobrazí nalezené informace.

## 6.4 Porovnávání suplování

Abychom mohli upozornit uživatele na aktualizaci suplování, musíme mít způsob, jak budeme nové a staré suplování porovnat. V projektu to je zprvu řešeno porovnáním dat vydání, a pokud je nějaká změna, musíme ještě porovnat, jestli nastaly změny v uživatelem zvolené třídě a jeho skupin. Tento případ u vyučujících nenastává, protože nemají skupiny a stačí porovnat JSON těchto objektů (textových řetězců), ale u tříd, kde si student volí skupiny, se musí ještě vyřadit ty, do kterých nepatří.

Ukázka kódu 3: Metoda filtrující hodiny podle skupin

```
private static HashSet<String> getUsersLessons(HashMap<Integer, Lesson> changes,
                                                HashSet<String> groups) {
    HashSet<String> ret = new HashSet<>();
    for (int k : changes.keySet()) {
        for (int i = 0; i < changes.get(k).getGroups().size(); i++) {
            if (groups.contains(changes.get(k).getGroups().get(i))) {
                ret.add(changes.get(k).getSubjects().get(i).toString() +
                        changes.get(k).getTeachers().get(i).toString() +
                        changes.get(k).getRooms().get(i).toString() +
                        changes.get(k).getGroups().get(i).toString() +
                        changes.get(k).getOrigins().get(i).toString());
                ret.size();
            }
        }
    }
    return ret;
}
```

Je-li počet studentových změn (filtrovaných) různý počet, nastala změna. Pokud ne, je potřeba je porovnat. To řeší odebrání stejných prvků množin a, b. Zůstane-li po tomto kroku jakýkoliv prvek, nastala změna.

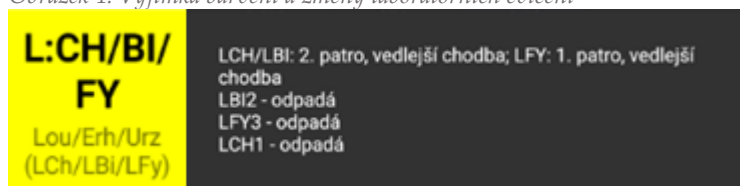
Ukázka kódu 4: Porovnávání změn suplování

```
boolean changed = false;
if (prefs.getString(context.getResources()
    .getString(R.string.key_pref_general_subject), context.getResources()
    .getString(R.string.pupil))
    .equals(context.getResources()
    .getString(R.string.teacher))) {
    // Teacher
    String jsonOld = gson.toJson(old.getChangesTeachers().get(cls));
    String jsonNew = gson.toJson(day.getChangesTeachers().get(cls));
    changed = !jsonOld.equals(jsonNew);
} else {
    // Student
    SharedPreferences tablePrefs = context.getSharedPreferences(context.getResources()
        .getString(R.string.category_tables), Context.MODE_PRIVATE);
    HashSet<String> groups = (HashSet<String>) tablePrefs.getStringSet(context.getResources()
        .getString(R.string.key_table_pref_groups), new HashSet<String>());
    HashSet<String> a = getUsersLessons(old.getChangesClasses().get(cls), groups);
    HashSet<String> b = getUsersLessons(day.getChangesClasses().get(cls), groups);
    if (a.size() != b.size()) {
        changed = true;
    } else {
        b.removeAll(a);
        if (b.size() > 0) {
            changed = true;
        }
    }
}
}
```

## 6.1 Laboratorní cvičení

Hodiny laboratorních cvičení jsou zvláštní případ, protože se každý týden střídají a často i nepravidelně. Bez přihlášení nelze s jistotou určit, jakou hodinu uživatel bude mít. Další problém je, že v rozvrzích občas chybí i celý předmět (tento rok například 1.A chybí laboratorní hodiny chemie). Pokud se nějaká změna ohledně těchto hodin objeví na suplování, bude laboratorní hodina podbarvena ostře žlutou barvou a po kliknutí na ni se zobrazí změny jednotlivých skupin. Toto ale neplatí pro vyučující, kterým se zobrazí hodina se skupinou, která byla napsána v suplování v normálním zobrazení.

Obrázek 4: Výjimka barvení a změny laboratorních cvičení



## 7 Příručka pro uživatele

### 7.1 Instalace a nastavení zařízení

Instalace probíhá spuštěním souboru aplikace APK. Při tomto způsobu instalace je zapotřebí mít povolenou instalaci aplikací třetích stran. Obvykle se tato volba nachází v *Nastavení* > (*Rozšířené nastavení*) > *Zabezpečení* > *Neznámé zdroje* (poloha zapnuto). Instalace z Google Play není v současné době možná.

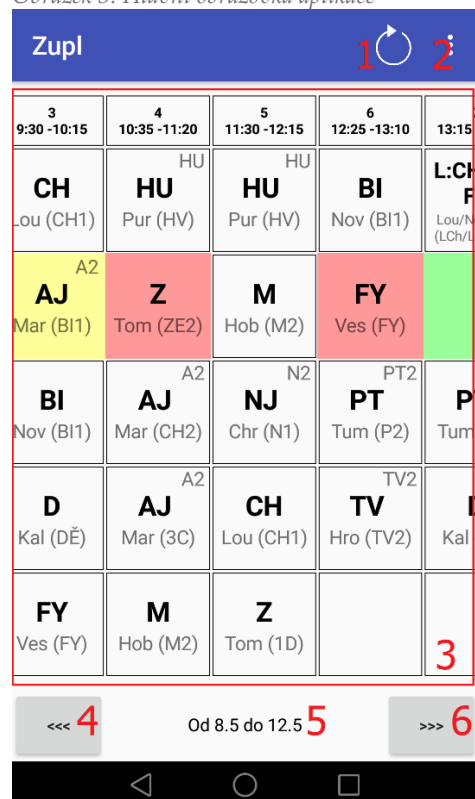
Aby aplikace aktualizovala suplování na pozadí, je potřeba nastavit, aby ji systém při vypnutí obrazovky nevypnul. Typicky se tato volba nachází v *Nastavení* > *Aplikace* > *Zupl* > *Baterie* > *Běžet i po vypnutí obrazovky* (poloha zapnuto).

### 7.2 Uživatelské rozhraní

#### 7.2.1 Hlavní obrazovka

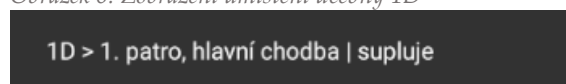
Hlavní obrazovka je místo, kde se budete získávat informace o vašich aktuálních hodinách a přepínat mezi jednotlivými týdny. Také se lze odtud dostat do nastavení. Kliknete-li na libovolnou hodinu v dolní části obrazovky, zobrazí se vám její poloha ve škole.

Obrázek 5: Hlavní obrazovka aplikace



1. Ruční aktualizace suplování
2. Přejít do nastavení
3. Posuvné zobrazení rozvrhů
4. Současný týden (5 dní)
5. Předchozí týden
6. Nadcházející týden

Obrázek 6: Zobrazení umístění učebny 1D



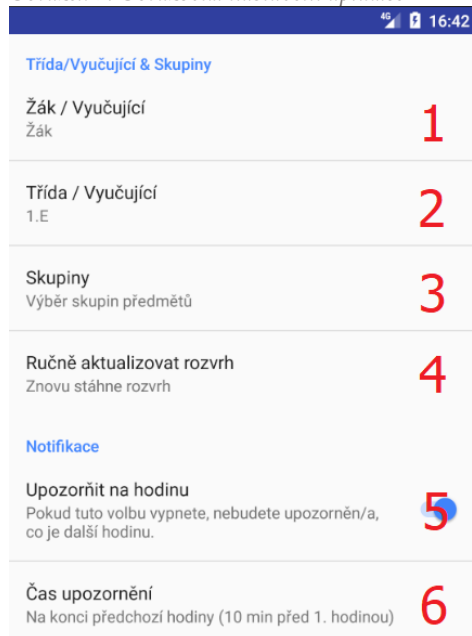
Legenda barev buněk:

- Zelená: hodina odpadá, odchozí přesun a výměna
- Červená: hodina navíc, přichází přesun a výměna
- Žlutá: změna, suplování, ostatní
- Ostrá žluta: jakákoliv změna laboratorních cvičení

## 7.2.2 Hlavní nastavení

V Hlavním nastavení je nastavitelná většina funkcí aplikace až na volbu vašich předmětových skupin, která se nachází pod volbou Skupiny.

Obrázek 7: Obrazovka nastavení aplikace

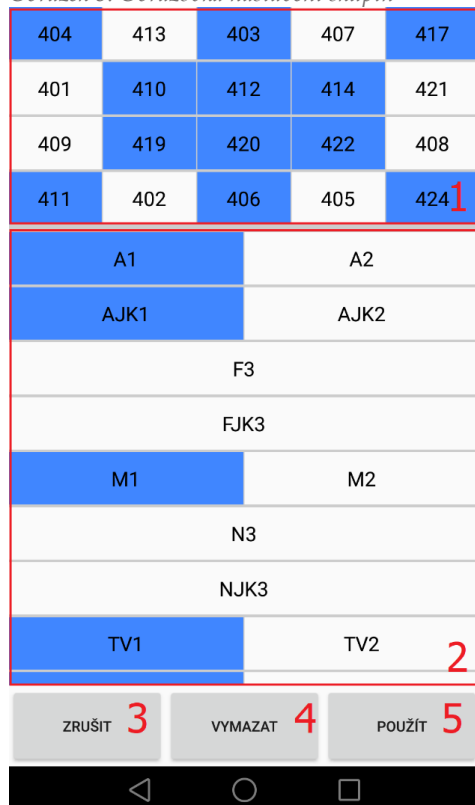


1. Volba žáka, nebo vyučujícího
2. Volba třídy
3. Přístup do volby skupin
4. Ruční aktualizace rozvrhu.
5. Zapnutí/Vypnutí upozornění před
6. Čas upozornění

## 7.2.3 Volba skupin

Volba skupin nabízí volbu předmětových skupin.

Obrázek 8: Obrazovka nastavení skupin

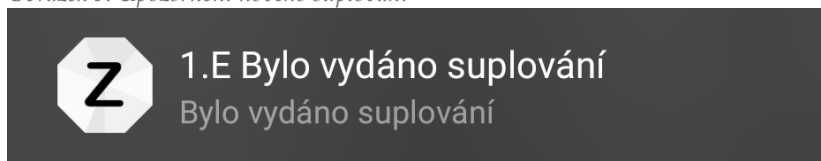


1. Výběr seminářů
2. Výběr ostatních skupin
3. Zrušení výběru a vrácení se do Hlavního nastavení
4. Vymazání výběru
5. Uložení výběru a vrácení se do Hlavního nastavení

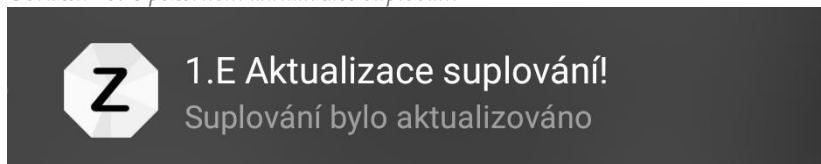
## 7.1 Upozornění pro uživatele

### 7.1.1 Suplování

Obrázek 9: Upozornění nového suplování



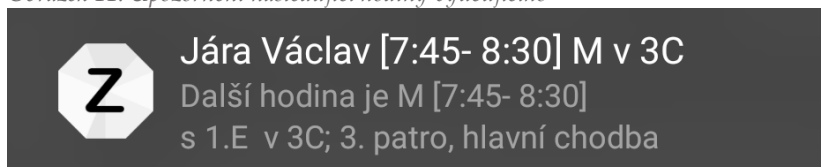
Obrázek 10: Upozornění aktualizace suplování



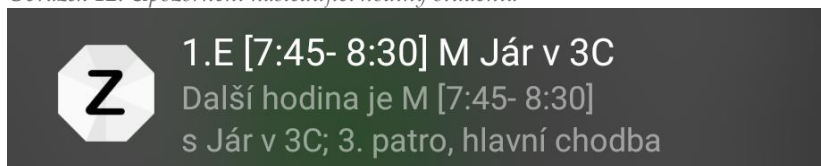
Tato dvě upozornění zobrazují pouze třídu, nebo učitele a stav suplování.

### 7.1.2 Před začátkem hodiny

Obrázek 11: Upozornění následující hodiny vyučujícího



Obrázek 12: Upozornění následující hodiny studenta



Tato upozornění zobrazují následující hodinu s časem, vyučujícím (třídou), učebnou a její pozicí ve škole.

## Závěr

Výsledná knihovna a aplikace splňují všechny body zadání a mají i funkce navíc.

Projekt by bylo možné rozšířit, aby podporoval i jiné školy a případně ho napsat i pro konkurenční operační systém iOS.

Největším problémem a pojídačem času bylo občasné nedoplánování dopředu, které způsobilo, že jsem musel některé části jak knihovny, tak aplikace předělávat a budu na tomto muset zapracovat. Kromě toho jsem se naučil vytvářet aplikace pro Android a mám ze svého projektu radost.



## Přílohy

### Seznam použité literatury a zdrojů

V tomto dokumentu

1. ART Features. ART and Dalvik. 2017-04-27 [cit. 2017-05-01]. Dostupné z [source.android.com/devices/tech/dalvik/](http://source.android.com/devices/tech/dalvik/)
2. Lacko. L. 2015. Vývoj aplikací pro Android. Computer press. Brno. 64. ISBN: 978-80-251-4347-6
3. Tamtéž, s. 64
4. Tamtéž, s. 65
5. Liam. Q. XML [online]. 2016-10-11 [cit. 2017-04-29]. Dostupné z [www.w3.org/XML/](http://www.w3.org/XML/)
6. Úvod do JSON [online]. [cit. 2017-04-29]. Dostupné z [www.json.org/json-cz.html](http://www.json.org/json-cz.html)
7. Marko. V. ART vs Dalvik - introducing the new Android runtime in KitKat [online]. 2013-12-04 [cit. 2017-04-29]. Dostupné z [infinum.co/the-capsized-eight/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat](http://infinum.co/the-capsized-eight/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat)
8. Tamtéž.
9. Hedley. J. 2016 [cit. 2017-05-06]. Dostupné z [jsoup.org](http://jsoup.org)
10. Gson Goals. README.md [online]. 2017-03 [cit. 2017-05-06]. Dostupné z [github.com/google/gson](https://github.com/google/gson)
11. prolink007. How to put a horizontal divisor line between edit text's in an activity. Stackoverflow [online]. 2010-11-10 [převzato 2017-04-09]. Dostupné z <http://stackoverflow.com/questions/4146221/how-to-put-a-horizontal-divisor-line-between-edit-texts-in-a-activity>
12. Vivek Khandelwal. How to draw border on just one side of a linear layout? Stackoverflow [online]. 2012-02-09 [převzato 2017-04-02] Dostupné z <http://stackoverflow.com/questions/9211208/how-to-draw-border-on-just-one-side-of-a-linear-layout>

### Seznam diagramů, obrázků, tabulek a ukázek kódu

Diagram 1: Softwarová architektura Zuplu .....	7
Diagram 2: Hierarchie tříd TableParseru .....	7
Diagram 3: Hierarchie aktivit aplikace .....	13
Diagram 4: Hierarchie přijímačů aplikace .....	14
Diagram 5: Hierarchie servisů aplikace .....	14
Obrázek 1: Normální logo aplikace .....	13
Obrázek 2: Ikona upozornění pro Android 7 .....	13
Obrázek 3: Grafická reprezentace tabulky a korespondujících tagů řádků .....	16
Obrázek 4: Výjimka barvení a změny laboratorních cvičení .....	19
Obrázek 5: Hlavní obrazovka aplikace .....	20
Obrázek 6: Zobrazení umístění učebny 1D .....	20

Obrázek 7: Obrazovka nastavení aplikace.....	21
Obrázek 8: Obrazovka nastavení skupin.....	21
Obrázek 9: Upozornění nového suplování.....	22
Obrázek 10: Upozornění aktualizace suplování .....	22
Obrázek 11: Upozornění následující hodiny vyučujícího .....	22
Obrázek 12: Upozornění následující hodiny studenta.....	22
Tabulka 1: Konstruktory, proměnné a metody třídy Lesson.....	7
Tabulka 2: Konstruktory, proměnné a metody třídy TableData .....	9
Tabulka 3: Konstruktory, proměnné a metody třídy SubsData .....	10
Tabulka 4: Konstruktory, proměnné a metody třídy Tablo .....	11
Tabulka 5: Konstruktory, proměnné a metody třídy TableTablo .....	12
Tabulka 6: Konstruktory, proměnné a metody třídy SubsTablo.....	12
Ukázka kódu 1: Objekt Lesson zapsaný v jazyce JSON.....	15
Ukázka kódu 2: Získání objektů dnů v týdnu.....	17
Ukázka kódu 3: Metoda filtrující hodiny podle skupin.....	18
Ukázka kódu 4: Porovnávání změn suplování .....	18