|  |  |  |
| --- | --- | --- |
| Logo__SSPU_2016_Barva | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Mobilní informační aplikace** | | |
| Michael Hluchník | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2018/2019 | |

#### 

Poděkování

* *poděkování (například vedoucímu práce).*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2016

*podpis autora práce*

**ANOTACE**

Cílem projektu bylo vytvořit mobilní aplikaci vypisující informace o daných předmětech ze hry Warframe. Zajišťuje jednoduché a přehledné vypisování dat, aby uživatel neměl problém najít jakýkoliv předmět ve hře. Aplikace byla vyvinuta pomocí jazyka JAVA ve vývojovém prostředí Android Studio. Data jsou uložena v podobě JSON souborů. Tyto soubory jsou uloženy na internetových stránkách githubu. Aplikaci tvoří čistě informační vzhled. Aplikace obsahuje tří hlavní stránky, mezi kterými se uživatel přemisťuje dle potřeby. Prioritou při tvorbě aplikace byl výsledný vzhled aplikace, aby byla přehledná a jednoduše ovladatelná. Aplikaci lze využívat ve všech mobilních zařízeních, které mají operační systém Android 5 a vyšší.

**Klíčová slova:** mobilní aplikace, Android Studio, JSON soubory, JAVA

OBSAH

[Poděkování 2](#_Toc532985808)

[Úvod 5](#_Toc532985809)

[1 Teoretická a metodická východiska 6](#_Toc532985810)

[1.1 Začátky v Android Studiu 6](#_Toc532985811)

[1.1.1 Emulátor nebo vlastní zařízení? 6](#_Toc532985812)

[1.2 Activity 6](#_Toc532985813)

[1.3 Soubory JSON 6](#_Toc532985814)

[1.4 AsyncTask 7](#_Toc532985815)

[2 Využité technologie 8](#_Toc532985816)

[2.1 Android Studio 3.2.1 8](#_Toc532985817)

[2.2 JAVA 8 8](#_Toc532985818)

[2.3 Android OS 8](#_Toc532985819)

[2.4 SDK 26.1.1 9](#_Toc532985820)

[3 Způsoby řešení a použité postupy 10](#_Toc532985821)

[3.1 Začátky v Android Studiu 10](#_Toc532985822)

[3.1.1 Nastavení pro vývojáře v mobilu 10](#_Toc532985823)

[3.2 Vytvoření projektu 10](#_Toc532985824)

[3.3 Activity main 11](#_Toc532985825)

[3.3.1 List View 11](#_Toc532985826)

[3.3.2 Vytvoření nové activity 11](#_Toc532985827)

[3.3.3 Přepínaní mezi aktivitami 11](#_Toc532985828)

[3.3.4 Posílání proměnné mezi aktivitami 12](#_Toc532985829)

[3.4 JSON parsing 12](#_Toc532985830)

[3.4.1 Přístup k internetu 12](#_Toc532985831)

[3.4.2 Parsování 13](#_Toc532985832)

[3.5 Vypisování 13](#_Toc532985833)

[3.6 Struktura 14](#_Toc532985834)

[3.6.1 Projektu 14](#_Toc532985835)

[3.6.2 JSON souboru 15](#_Toc532985836)

[4 Výsledky řešení, výstupy, uživatelský manuál 16](#_Toc532985837)

[Závěr 17](#_Toc532985838)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 18](#_Toc532985839)

[Seznam příloh 19](#_Toc532985840)

Úvod

Warframe Codex je mobilní aplikace naprogramováno pomocí jazyka JAVA. Tato aplikace vypisuje informace o jednotlivých předmětech ze hry Warframe. Tato aplikace bylo tvořena ve vývojovém prostředí Android Studio.

Cílem tohoto projektu bylo získání dat v podobě JSON souborů z internetu a jejich následné vypisování v přehledné a jednoduché formě. To vše v mobilní aplikaci.

V této dokumentaci se budeme zabývat postupy, které jsem při tvorbě tohoto dokumentu použil. Technologiemi, které jsem využíval. Vytvářením projektu v Android Studiu. Řešením problematiky získávání a parsování JSON souborů z internetu a jejich následné přehledné vypisování. Grafickou úpravou aplikace.

Tuto aplikaci jsem navrhl pro vlastní potřebu, abych měl rychlejší a jednoduší nalezení daných informací, když je budu potřebovat.

# Teoretická a metodická východiska

## Začátky v Android Studiu

Android Studio je vývojové prostředí zaměřené přímo na vývoj mobilních aplikací, z tohoto důvodu jsem si ho vybral.

Je zde také Visual Studio, které má také možnost vytváření mobilních aplikací ale z důvodu náročnějšího ovládání a velkého zabrání paměti jsem od něho upustil.

### Emulátor nebo vlastní zařízení?

Samotný emulátor je jednoduchý na spuštění a ovládání ale bohužel není v hodný pro všechny počítače kvůli jeho náročnosti na hardware.

Vlastní mobilní zařízení si musíte nastavit do režimu pro vývojáře a propojit s počítačem. Poté si ho musíte přepnout na přenos souborů. Nadále by si Android Studio mělo zjistit přístupné mobilní zařízení a s vaším dovolením program na něm spustit.

## Activity

Activity je to co uživatel vidí. Je tvořena dvěma soubory. Třídou, která vykonává úkony a provádí operace. Layoutem, který se zabývá grafickou stránkou. Activita jako celek komunikuje s uživatelem a vytváří okna. Většinou jsou aktivity prezentovány jako celá obrazovka, ale můžou mít i jiné použití, jako třeba plovoucí obrazovka či přímo vložené do jiné aktivity.

## Soubory JSON

JSON (JavaScript Object Nation) je způsob zápisu dat nezávislý na počítačové platformě. Je určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. Používají se pro přenos dat. Je člověkem čitelný a jednoduše editovatelný.

Tento typ souboru jsem zvolil pro jednoduchou manipulaci s daty a jednoduchosti čtení v jazyce JAVA.

## AsyncTask

Generická abstraktní třída AsyncTask umí asynchonně spustit jiný kód. Užitečná je zejména v tom, že nemusíte ručně vytvářet nová vlákna a starat se o ně. Využívá se většinou při náročnějších operacích. Využívám tyto funkce.

Funkce onPreExecute() se volá před započetím práce asynchronního úkolu, zde vytvářím progres dialogu, který informuje uživatele o probíhající činnosti. Ve funkci doInBackground probíhá výpočet. A ve funkci onPostExecute() se zpracovává vrácený výsledek. AsyncTask se spouští zavoláním funkce execute() s parametry.

# Využité technologie

## Android Studio 3.2.1

Android Studio je vývojové prostředí založené na Intellij ADEA které podporuje programovací jazyky jako je JAVA a Groovy. Také podporuje programovací jazyky C++ a Kotlin. Pro tento projekt jsem si vybral jazyk JAVA. Vytvořenou firmou Google pro platformy Windows, Mac OS X a Linux. Android Studio obsahuje mnoho různých funkcí, které usnadňují programování. Mezi užitečné nástroje patří například debugger, kterým si můžeme otestovat vytvořený program před tím, než jej nahrajeme do procesoru.

Toto vývojové prostředí jsem si vybral kvůli jednoduchosti a taky kvůli tomu, že se mi prostě líbilo. Byla to moje první zkušenost s tímto vývojovým prostředím a můžu doporučit.

## JAVA 8

Je objektově orientovaný [programovací jazyk](https://cs.wikipedia.org/wiki/Programovací_jazyk), který vyvinula firma Sun Microsystems. Java nejpopulárnější programovací jazyk. Díky své [přenositelnosti](https://cs.wikipedia.org/wiki/Přenositelnost) je používán pro [programy](https://cs.wikipedia.org/wiki/Počítačový_program), které mají pracovat na různých systémech počínaje čipovými kartami přes [mobilní telefony](https://cs.wikipedia.org/wiki/Mobilní_telefon) a různá zabudovaná zařízení aplikace pro desktopové počítače až po rozsáhlé distribuované systémy pracující na řadě spolupracujících počítačů rozprostřené po celém světě. Je také multiplatformní.

## Android OS

Je [mobilní operační systém](https://cs.wikipedia.org/wiki/Mobilní_operační_systém) založený na [jádře Linuxu](https://cs.wikipedia.org/wiki/Linux_(jádro)), který je dostupný jako otevřený software. Je používán na smartphonech, tabletech, [chytrých televizích](https://cs.wikipedia.org/wiki/Smart_TV) a dalších zařízeních. Jeho vývoj vede firma [Google](https://cs.wikipedia.org/wiki/Google) pod hlavičkou konsorcia firem Open Handset Alliance. Má mnoho verzí, podle kterých se odvádí název. Systém Android má největší zastoupení na světě mezi všemi operačními systémy.

Vytváření aplikace pro tento operační systém jsem si vybral z důvodu vlastnictví mobilu, který užívá tento operační systém. Projekt je kompatibilní s operačním systémem Android 5.0 a vyšší.

## SDK 26.1.1

Software development kit (sdk)je typická sada vývojových nástrojů umožňující vytváření aplikací pro jisté softwarové balíčky, frameworky, platformy, počítačové systémy, herní konzole, operační systémy nebo podobnou platformu. Pro vytvoření vlastní aplikace se bez toho neobejdete. Většinou jsou přiděleny k přímo k vývojovému prostředí.

# Způsoby řešení a použité postupy

## Začátky v Android Studiu

Poněvadž to byla moje první zkušenost s Android Studiem, musel jsem ho naučit používat. Jsou zde tři možnosti zkontrolování kódu.

Takzvané kladívko, které vám jenom zkontroluje kód, jestli je bezchybný.

Debugger – zubaté kolečko s trojúhelníčkem, který vám vypisuje všechny atributy a pomůže vám najít chybu.

Velký zelený trojúhelník vám spustí program, který vám poběží buď na emulátoru nebo vašem mobilním zařízení.

### Nastavení pro vývojáře v mobilu

Abych mohl vyvíjet aplikaci pro můj vlastní telefon, musel jsem si ho takzvaně přepnout do režimu „Pro vývojáře“. Toto mi umožnilo instalovat a zkoušet moji aplikaci v mém mobilu. Docílil jsem k tomu tak že v nastavení v informacích o zařízení jsem 6x rychle za sebou po klikal na číslo sestavení

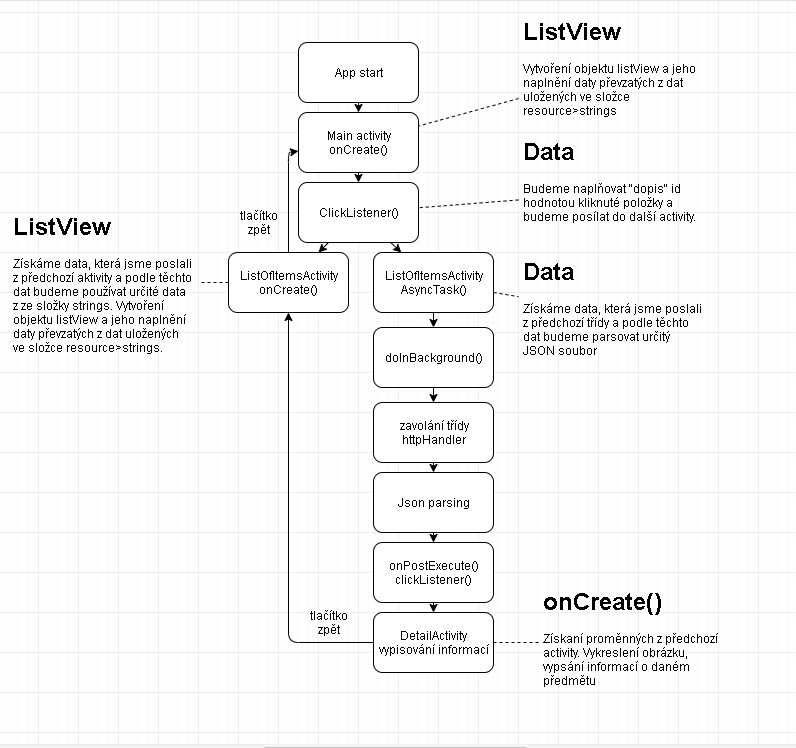
## Vytvoření projektu

Při vytváření nového projektu je důležité si vybrat správně, pro jaký typ operačního systému Android ho vytváříme. Samotné Android Studio nám dá na výběr z mnoha možností a samo nám doporučí ty, které se nejvíce používají. Ale pozor, pokud chcete mít aplikaci pro vlastní použití tak váš mobil nemusí mít stejnou verzi, kterou vám studio doporučí. Z toho důvodu si to raději zkontrolujte v nastavení telefonu.

Poté si budete vybírat, jak by měla vaše úvodní obrazovka vypadat. Budete mít graficky ukázanou nabídku. Já si vybral prázdnou.

Jakmile máte vytvořen projekt, budete mít k dispozici třídu mainActivity a její layout aktivity\_main.

Struktura



## Activity main

### List View

ListView je skupina zobrazení, která zobrazuje seznam rolovacích položek. Položky seznamu se automaticky vkládají do seznamu pomocí adaptéru, který přenáší obsah ze zdroje, jako je pole nebo databázový dotaz, a konvertuje každý výsledek položky do zobrazení, které je umístěno do seznamu.

Vybral jsem si tento objekt, který jsem si graficky vložil do layoutu main\_activity. Pomůže mi jednoduše rolovat mezi položkami v něm vložené. Poté si vytvoříme třídu adaptér, který nám pomůže v zobrazení položek v ListView. V activity main si vytvoříme funkci, která bude reagovat na kliknutí na tyto položky a odkáže nás na další activitu. Tato funkce už je předdefinována v jazyce JAVA tak jsem toho využil.

Musíme si ho ve třídě inicializovat:

ListView mujListView;

mujListView = findViewById(R.id.(ID které jsme mu přiřadili));

### Vytvoření nové activity

Vytvoření nové aktivity je jednoduchá záležitost. Stačí pravým kliknutím myši na danou záložku vybrat „nový“ > „activity“ a potom si vybereme, zda chceme prázdnou či nějak předem upravenou. Já si vybral prázdnou.

V této activitě si taktéž graficky přidáme ListView. Samozřejme mu dáme ID odlišné od minulého ListView.

### Přepínaní mezi aktivitami

Toto je jedna z nejdůležitějších akcí ve všech projektech, které se týkají mobilních aplikací. Její syntaxe je velice jednoduchá a rychlá. Nejprve ji musíme inicializovat.

Intent (název) = new Intent(getApplicationContext(), (název třídy aktivity);

A poté spustit.

startActivity(název);

Toto jsem si vložil do funkce, která reaguje na kliknutí.

### Posílání proměnné mezi aktivitami

I když toto taktéž není náročná činnost, musíme si u ní dávat pozor na správné indexovaní proměnných. Tato činnost má dvě fáze. Nejprve si v activitě ze které chceme posílat, vložíme proměnou do něčeho, co můžeme nazvat dopis. U nás to bude konkrétně ve funkci při kliknutí.

(název).putExtra(„jméno“, proměnná);

Ve druhé activitě tento dopis vybalíme. Samozřejmě můžeme posílat jakýkoliv typ proměnných, avšak pak budeme muset přizpůsobit i náš kód. A to změnou v názvu a atributech funkcí getExtra().

Intent (název) = getIntent();

Proměnná = (název).getExtra(„jméno“);

Já si je získával v rozdělovači switch, protože jsem si posílal hodnoty, které byly různé podle toho na jakou položku z ListView jsem klikl. Toto všechno jsem si zkoušel pomocí obyčejných proměnných, které jsem si vytvořil na zkoušku.

## JSON parsing

Jazyk JAVA má knihovny určené k parsování JSON souborů a proto jsem si myslel, že to bude jednoduchá a rychlá záležitost. Mýlil jsem se. Musel jsem vyřešit problematiku kdy, a kde bych to měl parsovat, aby se mi to už zobrazilo přesně tam, kde jsem potřeboval a kdy jsem potřeboval. Nakonec jsem zvolil asynchronní funkce. Své JSON soubory jsem měl uložené na internetových stránkách, tak jsem ještě potřeboval vyřešit tuto problematiku.

3.4.1 Struktura JSON souboru

Svůj JSON soubor jsem si vytvořil sám. Data jsem si sehnal z internetových stránek nebo jiných JSON souborů, které jsem našel. Struktura JSON souborů je velmi primitivní a skládá se jenom z jednoduchých objektů, které mají atributy a hodnotami.

[{

"name": "Adarza Kavat",

"description": "Piercing eyes reflect deadly instincts.",

"health": 40,

"shield": 70,

"armor": 50,

"stamina": 8,

"power": 100,

"type": "Pets",

"imageName": "adarza-kavat.png"

}, ... ]

### Přístup k internetu

Aby naše aplikace mohla přistupovat k internetu, a využívat informace z něho získané musíme si toto povolit v souboru AndroidManifest.xml. Příkazem:

<uses-permission android:name=“android.permission.INTERNET“/>

Poté jsem si vytvořil třídu, která nám zajistí ověření internetové adresy a zajistí nám přijetí dat z JSON souboru v podobně stringu.

### Parsování

Samotné parsování řeším v mé druhé activitě, kde jsem si pro to vytvořil asynchroní funkce.

V asynchroní funkci „DoInBackground“ si inicializujeme třídu, která nám zajišťuje komunikování s internetovou adresou. Pokud máme více stránek s JSON soubory můžeme si je posílat do této třídy postupně pomocí switche či jiného přepínače. Což já mám více souborů JSON a proto to také využívám.

Poté si pomocí předdefinovaných funkcí v jazyce JAVA na parsovaní JSON souboru jako jsou JSONArray(); JSONObject(); a opakovače for(), rozdělíme tento string podle toho, jak potřebujeme. Jako například:

JSONArray frames = new JSONArray(jsonString);

For(int i = 0;i < frames.lenght();i++){

JSONObject c = frames.getJSONObject(i);

String name = c.getString(„name“);}

Nakonec si tyto data přepošleme kde potřebujeme ve funkci onPostExecute(). Kde mám další funkci při kliknutí na položky v listView a až v ní posílám informace dál a volám svou třetí activitu.

## Vypisování

Ve své třetí aktivitě si přidám obrázek, který mi zobrazí danou položku a obyčejný textView ve kterém vypisuji informace o této položce.

Nakonec si upravím grafický návrh všech aktivit, aby to bylo hezké a přehledné a jsem hotov.

# Výsledky řešení, výstupy, uživatelský manuál

výčet splněných a nesplněných cílů, obrázky (schémata, vzorce apod.) z finálního provedení, prokázání funkčnosti, výsledné parametry výrobku apod.

* podle zaměření a charakteru práce je třeba volit vhodný nadpis pro tuto kapitolu, je samozřejmě možné i rozdělení na více kapitol (např. Uživatelské rozhraní internetové aplikace; Administrace internetové aplikace…)

# Závěr

Cílem tohoto projektu bylo vytvoření funkční mobilní informační aplikaci. Tato aplikace byla vytvořena ve vývojovém prostředí Android Studio pro operační sytém Android 5.0 a vyšší. Určené cíle jako zobrazení dat z JSON souborů, které jsou převzaty z internetu, bylo splněno. Ke splnění, jsem musel vyřešit problematiku zpřístupnění internetu v aplikaci. Následné přehledné vypisování dat bylo splněno s pomocí mých grafických znalostí.

Do budoucna by bylo možné rozšířit aplikaci o přihlašovací systém, který by byl propojen s Warframe API. Toto by umožnilo uživateli zobrazit předměty, které již vlastní. Rovněž zvažuji zprostředkování aplikace pomocí služby Google Play Store.

Seznam použitýCH INFORMAČNÍCH ZDROJů

[0] BOHMAN, Ludvík. Zákon o pojistné smlouvě. Praha: Linde Praha a. s., 2004. 381 s. ISBN80-7201-504-4

[2] DUCHÁČKOVÁ, Eva. Principy pojištění a pojišťovnictví. 3. aktualizované vydání. Praha: Ekopress 2009. 224 s. ISBN 978-80-86929-51-4

[3] KUBALA, Petr. Planetární dvojcata - Věda a technika (Český rozhlas) [online].   
Č. 2000-2008, poslední revize 19. 3. 2008 [cit. 2008-03-20].  
<http://www.rozhlas.cz/veda/vesmir/\_zprava/435849>.

[4] KULDOVÁ, O., FLEISCHMANNOVÁ, E. Metodická příručka k technice administrativy a obchodní korespondence. 1.vyd. Praha: Fortuna 1998. 111 s.   
ISBN 80-7168-574-7. Kapitola 6, Metody nácviku psaní hmatovou metodou,   
s. 28-29.

[5] VLACH, J. JE Temelín a zásobování teplem. Energetika, 2001, roč. 51, č. 3, s. 84 -85. ISSN 0375-8842.

[6] BUTTERFIELD, B.< <https://www.youtube.com/watch?v=dFlPARW5IX8>>,< https://www.youtube.com/watch?v=6ow3L39Wxmg&t>,< https://www.youtube.com/watch?v=rdGpT1pIJlw&t>

[7] PANWAR, A. < <https://www.youtube.com/watch?v=Vcn4OuV4Ixg&t>>

[8] Autor neznámý < <https://www.tutorialspoint.com/android/android_json_parser.htm>>

[9] < [https://cs.wikipedia.org/wiki/Hlavn%C3%AD\_strana](https://cs.wikipedia.org/wiki/Hlavní_strana)>

[10] < <https://stackoverflow.com/>>

[11] <https://github.com>

Seznam příloh

č. 1 Titulní list

č. 2 Čestné prohlášení

č. 3 Poděkování

Nepovinná část – pokud nemáte žádné přílohy ke své práci, tuto část odstraňte!

* Přílohy se zařazují na konec práce.
* Jsou to texty, obrázky, grafy, tabulky, které by přímo v textu byly zbytečně detailní, ale mají být po ruce k dokreslení východisek i výsledku řešení.
* Jsou číslovány a v textu se na ně může odkazovat.
* Před první přílohu se umisťuje seznam příloh.
* Každá příloha je označena číslem - např. Tabulka č.. 1, Schéma č. 2, Obrázek č. 3.
* Každá tabulka by měla mít i vlastní název, který stručně vystihuje její obsah.
* (Tabulka č. 1 Zakázky stavebních prací v roce 2009-2010).
* Pokud je z tabulky vytvořen graf, umístíme jej na stejné stránce jako tabulku.

**Příloha č. 1: Titulní list**