

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Interaktivní turistický průvodce pro kraj Vysočina



2017

Vedoucí práce: Mgr. Martin Tr-  
nečka, Ph.D.

Lukáš Krystek

Studijní obor: Informatika, prezenční  
forma

## **Bibliografické údaje**

Autor: Lukáš Krystek  
Název práce: Interaktivní turistický průvodce pro kraj Vysočina  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2017  
Studijní obor: Informatika, prezenční forma  
Vedoucí práce: Mgr. Martin Trnečka, Ph.D.  
Počet stran: 43  
Přílohy: 1 DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Lukáš Krystek  
Title: Interactive tourist guide for the region Vysočina  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2017  
Study field: Computer Science, full-time form  
Supervisor: Mgr. Martin Trnečka, Ph.D.  
Page count: 43  
Supplements: 1 DVD  
Thesis language: Czech

## **Anotace**

*Cílem této závěrečné práce je navržení a realizace turistické, mobilní aplikace s využitím rozšířené reality pro snadnou orientaci v kraji Vysočina.*

## **Synopsis**

*This thesis deals with designing tourist mobile application on the Android platform using the augmented reality for easy orientation tourists in the region Vysočina.*

**Klíčová slova:** Rozšířená realita; Android; závěrečná práce; dokumentace

**Keywords:** Augmented reality; Android; thesis; documentation

Rád bych poděkoval především svému vedoucímu Mgr. Martinu Trnečkovi za profesní i lidskou podporu při realizaci této závěrečné práce a stejně tak bych chtěl poděkovat své přítelkyni za podporu a pochopení při překonávání dílčích problémů.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora

# Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
1.1	Definice pojmů . . . . .	8
1.2	Zadání aplikace . . . . .	9
<b>2</b>	<b>Mobilní platformy</b>	<b>10</b>
2.1	Mobilní aplikace . . . . .	10
2.2	Typy mobilních platforem . . . . .	10
2.2.1	Rozdíly mezi iOS a Android . . . . .	10
2.3	Zastoupení mobilních platforem . . . . .	11
<b>3</b>	<b>Android</b>	<b>12</b>
3.1	Historie . . . . .	12
3.2	Zajímavost verzování platformy Android . . . . .	12
3.3	Klíčové prvky při vývoji . . . . .	12
3.4	Vývojové prostředí . . . . .	13
3.5	Verze vs. vývoj . . . . .	13
<b>4</b>	<b>Postup vývoje</b>	<b>15</b>
4.1	Vodopádový model . . . . .	15
4.2	Verzovací systém . . . . .	17
4.2.1	TFVC vs. Git . . . . .	17
4.2.2	Obecné využití . . . . .	17
<b>5</b>	<b>Analýza dat před tvorbou aplikace</b>	<b>18</b>
5.1	Analýza technologií . . . . .	18
5.2	SQL databáze . . . . .	19
5.2.1	MS SQL . . . . .	20
5.2.2	Normalizace DB . . . . .	20
5.2.3	Systém auditu nad databází . . . . .	20
5.3	Webová administrace . . . . .	20
5.3.1	Ukázka administrace úkolu . . . . .	21
5.4	Webová služba WCF . . . . .	22
5.4.1	Třívrstvá architektura . . . . .	22
5.4.2	Historie . . . . .	22
5.4.3	Co je WCF . . . . .	24
<b>6</b>	<b>Mobilní aplikace GeoHra</b>	<b>25</b>
6.1	Oprávnění aplikace . . . . .	25
6.1.1	Oprávnění pro aplikaci GeoHra . . . . .	26
6.2	Synchronizace dat . . . . .	27
6.2.1	Způsob synchronizace . . . . .	28
6.3	Hlavní obrazovka aplikace . . . . .	28
6.4	Rozhledny . . . . .	29

6.5	Židovské památky . . . . .	29
6.6	GeoHry . . . . .	29
6.7	Úkoly . . . . .	30
6.8	Rozšířená realita . . . . .	32
6.9	AR v aplikaci . . . . .	32
6.10	Popis struktury Android aplikace . . . . .	33
	6.10.1 Adresář app . . . . .	33
	6.10.2 Ikonky . . . . .	34
6.11	Popis tříd . . . . .	34
	6.11.1 Namespace .Activites . . . . .	35
	6.11.2 Namespace .Adapters . . . . .	36
	6.11.3 Namespace .DataObjects . . . . .	36
	6.11.4 Namespace .Fragments . . . . .	37
	6.11.5 Namespace .Services . . . . .	37
	6.11.6 Namespace .Utils . . . . .	37
	<b>Závěr</b>	<b>40</b>
	<b>Conclusions</b>	<b>41</b>
	<b>A Obsah přiloženého DVD</b>	<b>42</b>
	<b>Bibliografie</b>	<b>43</b>

## Seznam obrázků

1	Ukázka verzí platformy Android reprezentované sladkostmi. . . .	13
2	Diagram vodopádového modelu. . . . .	15
3	Serverový databázový model. . . . .	19
4	Databázové normálové formy. . . . .	20
5	Seznam her k rozhlednám. . . . .	21
6	Úprava hry rozhledny. . . . .	21
7	Úprava úkolu rozhledny. . . . .	23
8	Třívrstvá architektura. . . . .	23
9	Synchronizace aplikace. . . . .	28
10	Hlavní obrazovka aplikace. . . . .	29
11	Přehled památek. . . . .	30
12	Přehled her. . . . .	31
13	Herní obrazovka - úkol dojít na místo. . . . .	31
14	Ikonka bodu zájmu v rozšířené realitě. . . . .	32
15	Rozšířená realita aplikace zobrazující nejbližší body zájmu. . . .	33
16	Ikonka aplikace GeoHra. . . . .	34

## Seznam tabulek

1	Přehled prodaných smartphone v tisících. . . . .	11
---	--	----

## Seznam vět

1	Definice (Definice AR) . . . . .	32
2	Definice (Definice AR - překlad) . . . . .	32

## Seznam zdrojových kódů

1	Ukázka definice verze aplikace v <i>build.gradle</i> souboru. . . . .	14
2	Ukázka vyžadování hardware komponenty fotoaparátu. . . . .	26
3	Ukázka oprávnění k přístupu fotoaparátu. . . . .	26
4	Příklad dotazu na server. . . . .	27
5	Příklad odpovědi ze serveru. . . . .	39

# 1 Úvod

Podkladem pro tuto bakalářskou práci bylo vyhlášení tendru na dodavatele dvou mobilních aplikací, které by interaktivní formou zatraktivily některé turistické cíle v kraji Vysočina. Veřejnou soutěž vyhrála společnost ALTAIR SOFTWARE s.r.o. v níž jsem zaměstnaný. Zadavatel soutěže pak definoval základní funkcionality a grafický rámeček, který musel vycházet z grafického manuálu kraje a stanovil základní požadavky na publicitu, vycházející zejména z Evropského fondu pro regionální rozvoj, ze kterého zadavatel částečně financoval celý projekt.

Požadavkem pak bylo vytvoření dvou mobilních aplikací, přičemž jednou aplikací byl mobilní průvodce pro kraj Vysočina a druhým požadavkem pak bylo vytvoření interaktivní aplikace GeoHra, jejíž popis je předmětem této práce.

## 1.1 Definice pojmů

Tato kapitola velmi jednoduše vysvětlí pojmy, které se budou v následujícím textu vyskytovat.

- **Operační systém** je speciální forma software, která je spuštěna se startem zařízení a stará se o běh ostatních programů a řídí periférije.
- **Framework** je ucelená knihovna podpurných a realizačních kódů, zaměřených na jeden problém. Například mějme problém „práce se soubory“ a framework umožňující číst, upravovat, mazat soubor, procházet složkovou strukturu atd.
- **Počítačová/mobilní platforma** je v informatice chápáno, jako pracovní prostředí uživatele. Zahrnuje hardwarovou i softwarovou část. Jde tedy o kompletní zařízení připravené k použití.
- **Augmentovaná/rozšířená realita** je počítačová simulace přidaná do reálného obrazu. Vykreslení se většinou realizuje na obrazovku, kde realita se pořizuje z fotoaparátu zařízení. V textu se může vyskytovat pod zkratkou AR.
- **Server** je platforma, která běží nonstop a pro klienta je dostupná většinou přes internet.
- **Verzovací systém** se rozumí ukládání historie změn s metataggy času, odesílatele změn atp. Následně se lze k verzi vrátit, nebo provést takzvaný diff. (odvozeno od slova difference), neboli provést porovnání a zobrazit rozdíly mezi verzemi souboru.
- **Webová služba** je software, který běží na serveru poskytující data aplikacím.



## 1.2 Zadání aplikace

Na základě obecného popisu uvedého v zadávací dokumentaci veřejné zakázky, kde byl požadavek vytvořit mobilní aplikaci, která turistu fyzicky provede krajem Vysočina po určitých naučných trasách, kdy tyto trasy budou dynamicky měnitelné přes webové administrační rozhraní a zároveň každá stezka může obsahovat libovolný počet dílčích úkolů, jako například zadání správné odpovědi, pořízení multimediálního obsahu, výběr z nabízených možností a další.

Pro tyto požadavky musela být vytvořena databáze obsahující veškeré trasy s úkoly. Pro plnění této databáze byla vytvořena webová aplikace. Synchronizace mezi databází a aplikacemi je dostupná pomocí webové služby. Veškeré části si později popíšeme.

## 2 Mobilní platformy

Mobilní platformou se myslí přenositelná elektronika v kompaktních rozměrech, tedy telefon, tablet, hodinky.

### 2.1 Mobilní aplikace

Mobilní aplikace je software vytvořený pro mobilní platformy využívající intuitivního ovládání. Za mobilní platformy se považují telefony, tablety, hodinky a další zařízení s operačním systémem umožňujícím rozšířit si funkce mobilními aplikacemi přes doplňkové aplikace.

Mobilní aplikace lze stahovat z oficiálních zdrojů, takzvaných obchodů, nebo mít instalační soubor v souborovém systému zařízení. Aplikace běží jen pro tu platformu, pro kterou je naprogramována, a zároveň musí platforma být stejné či vyšší verze, pro kterou je aplikace definována. Pro publikování mobilní aplikace na oficiálním obchodu je potřeba být zaregistrován u příslušného obchodu jako vývojář a mít certifikát, kterým se instalátor aplikace před publikací podepíše.

### 2.2 Typy mobilních platforem

Svět mobilních zařízení, konkrétně chytrých telefonů a tabletů, se pomyslně dělí dle mobilních platforem na „Androidáky“, „iOSáky“ či „Applisty“, jelikož mobilní platforma iOS pochází od firmy Apple, a ostatní.

#### 2.2.1 Rozdíly mezi iOS a Android

Vede se smyšlená válka mezi uživateli dvou nejvíce využívaných mobilních platforem, kde se oba navzájem osočují, že jedna či druhá platforma je lepší. V této době, kdy zařízení dosáhla několikajádrových procesorů a několika gigabajtů paměti ram, je nesmysl se bavit o hardwarové převaze. Sebevýkonnější zařízení, pokud se budeme bavit o telefonech, tak nevyužijeme naplno. Jde tedy o operační systém.

Než začneme shazovat konkurenční operační systém, měli bychom se zamyslet nad filozofií, kterou výrobce razí. U Androidu je to otevřenost a přizpůsobivost uživateli, narozdíl Apple se systémem iOS preferuje soukromí.

Někdo namítne, že iOS je bezpečnější, ale nemá pravdu. Díky nižšímu zastoupení zařízení s iOS je i méně útoků a především Android je v drtivé většině na zařízení neinstaluje surový. Každý výrobce si Android přizpůsobí podle sebe a zde vznikají chyby. Tudiž chybný není operační systém, ale jeho nastavba.

Bezpečností aktualizace Androidu vychází často, chyba však vzniká až u koncového výrobce telefonu, který vydá aktualizaci se svou nastavbou. V tomto případě mají navrch čínští výrobci telefonů, například Xiaomi, Meizu, kteří se o svá zařízení starají i několik let zpětně.

## 2.3 Zastoupení mobilních platforem

Jak již bylo zmíněno, Android je nejrozšířenější mobilní platformou a toto tvrzení dokazuje tabulka 1 prodaných chytrých telefonů zveřejněná na mobilmania.cz [1].

Tabulka 1: Přehled prodaných smartphone v tisících.

Operační systém	Distribuce 1Q16	Tržní podíl 1Q16 [%]	Distribuce 1Q15	Tržní podíl 1Q15 [%]
<b>Android</b>	293.771,2	84,1	264.941,9	78,8
<b>iOS</b>	51.629,5	14,8	60.177,2	17,9
<b>Windows</b>	2.399,7	0,7	8.270,8	2,5
<b>BlackBerry OS</b>	659,9	0,2	1.325,4	0,4
<b>ostatní</b>	791,1	0,2	1.582,5	0,5

Mezi Ostatní platformy se řadí například Tizen od Samsungu, Ubuntu, Sailfish a Firefox OS.

## 3 Android

V předchozích kapitolách bylo velmi ve zkratce vysvětleno, které technologie byly použity pro podpůrnou část mobilní aplikace. Tato část není na první pohled vidět, ale z funkčního hlediska je nepostradatelná.

Tato kapitola vysvětlí problematiku mobilních aplikací a pokusí se vysvětlit, proč jsou předchozí kroky pro vytvoření mobilní aplikace nezbytné. Současně popíše aplikaci GeoHra kraje Vysočina.

### 3.1 Historie

Počátek Android se datuje k roku 2003, kdy malá startupová firma Android inc., která popisovala svůj projekt, jako obrovský potenciál ve vývoji chytřejších mobilních zařízení. Prvotní záměr však mířil na software do digitálních fotoaparátů. Díky poznatku, že zvolené portfólium není dostatečně velké, upřel Android inc. svou vizi v úspěch v software pro sluchátka. V polovině roku 2005 však přišla zlomová situace pro společnost Android inc. a to, nadnárodní korporace Google inc. svou akvizicí pohltila výše zmiňovaný startup a započala tak vývoj stejnojmenné mobilní platformy.

Historie byla čerpána z internetových zdrojů [5], [6]

### 3.2 Zajímavost verzování platformy Android

Kromě standardního očíslování se platformy Android pojmenovávají i jmenným názvem a tyto názvy začínají na následující písmenko abecedy, přičemž slovo reprezentuje nějakou sladkost.

Je pravda, že první dvě verze systému neměli žádná jména, pouze číselné označení, ale verze 1.5 se jménem „Cupcake“ odstartoval tuto máni.

Obrázek 1 není kompletní, v současné době jsou již dostupné verze **Lollipop**, **Marshmallow**, **Nougat** a **Oreo**, přičemž poslední zmiňovaná verze je v současné době zatím v beta verzi.

### 3.3 Klíčové prvky při vývoji

Mobilní zařízení jsou zařízení napájená z baterie, která z pravidla, kvůli kompatibilním rozměrům zařízení, je malá a proto je zde důraz na úsporu elektrické energie.

Tento důraz lze chápat tak, že zdrojový kód nebude plýtvat pamětí a zároveň nebude více, než je nezbytně nutné, vytěžovat procesor mobilního zařízení, stejně tak jeho moduly které jsou GPS, bluetooth a další. Stejně tak záleží na dobré architektuře aplikace, což souvisí s úniky pamětí a rychlostí.



Obrázek 1: Ukázka verzí platformy Android reprezentované sladkostmi.

### 3.4 Vývojové prostředí

V počátcích platformy Android se aplikace psali v Eclipse s příslušným rozšířením ADT (Android Development Toolkit). Tato varianta byla nejjednodušší ze strany Google, ale pro vývojáře hotové peklo. Rychlost Eclipse byla výrazně snížena kvůli spoustě procesů běžících na pozadí. Tohoto faktu si Google byl vědom a intenzivně pracoval na vlastním prostředí. Nayval jej **Android Studio** a jeho první verze vyšla v době Android **Jelly bean** v roce 2013.

V současné době je Android Studio velmi chytré ve smyslu našeptávače ve zdrojovém kódu, různých automatických skriptech a příkladových skriptů. Zde je na místě poděkovat Google, ale na druhou stranu, pokud chtěla mít základnu uživatelů, museli mít k tomu aplikace třetích stran a ty by je nevytvářeli v takovém množství, kdyby vývoj měl zabrat několikanásobně více času než dnes.

### 3.5 Verze vs. vývoj

Při vývoji Android aplikací se vyžadují dvě verze. První z nich se nazývá *versionCode*, která udává pořadí verze. Jde o celočíselnou hodnotu a nová vydávaná verze nesmí být stejná, nebo nižší než aktuální vydaná verze.

Druhou verzí je *versionName* obsahující klasickou verzovací posloupnost, přičemž je zadávaná jako textový řetězec.

Struktura verzování je „<major>.<minor>.<point>-<flavor>“. Pro přehlednost by se měla dodržovat jednoduchá pravidla verzování. Vyžadovaná jsou ale-

spoň dvě hodnoty a každá následující verze ji zvyšuje.

- **<major>** Změna koncepce, nebo výrazná změna UI, nebo spousta nových funkcionalit
- **<minor>** Oprava spousty chyb, nebo změna tlačítek, nebo přidána jedna či dvě nové funkcionality
- **<point>** Různé drobné opravy, např. překlepy atp.
- **<flavor>** Zná pouze hodnotu *debug* nebo *full*, přičemž *full* se nepíše. Používá se k definici, kterou variantu aplikace sestavit.

Příklad zvyšování verze. Uvažujme verzi 1.2 a nová verze obsahuje velký zásah do uživatelského rozhraní, zvedneme verzi *<major>* a zároveň vynulujeme následující verze. Výsledná verze bude 2.0.

```
1 android {
2     ...
3     compileSdkVersion 23
4     buildToolsVersion '25.0.3'
5     ...
6     defaultConfig {
7         applicationId "cz.altairsoftware.vysocina.geohra"
8         minSdkVersion 9
9         targetSdkVersion 23
10        versionCode 16
11        versionName '2.5'
12        signingConfig signingConfigs.AltairSign
13    }
14    ...
15 }
```

Zdrojový kód 1: Ukázka definice verze aplikace v *build.gradle* souboru.

Soubor *build.gradle* obsahuje více než-li jen popis verze aplikace. Obsahuje také informace o sestavení verzí, kterým certifikátem se aplikace podepisuje, minimální verzi Android, verzi Android, případně maximální verzi Android, jmenný prostor aplikace a další důležité informace pro sestavení aplikace.

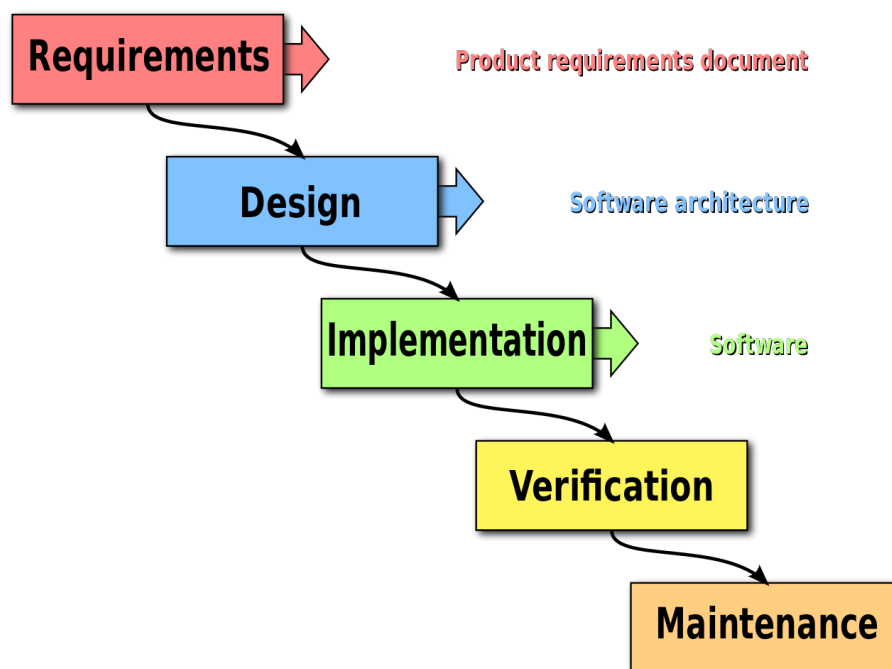
Na příkladu 1 je vidět, že aplikace GeoHra již má **16** verzí. Prozradím, že ne všechny verze byli na Obchodu Google publikované, ale šlo o verze testované Krajem Vysočina.

## 4 Postup vývoje

Z povahy projektu byla volba způsobu vývoje dána předem. Zadavatel si sám definoval funkční a technické požadavky na poptávaný systém, stejně jako grafický manuál, analýzu rizik a časový harmonogram realizace. Využilo se tedy vodopádového modelu se všemi jeho výhodami a slabinami.

### 4.1 Vodopádový model

Jde o vývojový model, kde se postupuje krok za krokem po všech fázích vývoje, tedy od analýzy dat, přes návrh aplikace, realizace aplikace, předání na testování až do finálního nasazení. Diagram vývoje je znázorněn na obrázku 2.



Obrázek 2: Diagram vodopádového modelu.

Zdrojem pro tuto kapitolu byl [4].

- **1. Fáze - requirements**

V rámci první fáze projektu proběhla předimplementační analýza projektu, založená na funkčních a technických požadavcích zadavatele. Součástí této fáze bylo detailní porozumění všem požadavkům zadavatele a jejich rozpracování ve formě FRD – Formal Requirements Document, který byl formálně odsouhlasen před započítím následující fáze.

- **2. Fáze - design**

Na základě odsouhlaseného FRD bylo třeba vytvořit ke každému funkčnímu procesu funkční popis chování všech částí aplikace. Součástí tohoto popisu tak byl funkční popis chování aplikace ze strany jednotlivých subjektů, vstupujících do procesu, tedy uživatele, administrátora, popřípadě aplikace. Nedílnou součástí každého takového FDD – Functional Design Dokumentu byl use case diagram, sekvenční diagram hlavního a alternativních scénářů, slovní popis funkčního požadavku a odkaz na TDD – Technical Design Document. TDD pak obsahoval technický popis realizace funkčního požadavku, zejména pak class model, table model a process diagram.

### • 3. Fáze - implementation

V průběhu fáze implementace se na základě FDD a TDD realizoval proces vývoje mobilní aplikace.

Vzhledem k ne striktně formulované architektuře AR a ke složení vývojářského týmu, v rámci kterého se na projektu podílel projektový manažer a dva vývojáři, kdy každý z vývojářů měl na starost vývoj jedné z platforem, tedy verzi pro Android a verzi pro iOS, bylo žádoucí, aby se některé části vyvíjely agilně. Agilní metodyka se rozumí každodenní scrum, kde se vývojáři a projektový manager seznámí s průběhem předchozího dne a plánem na aktuální den, a týdenních sprintů, které byly založeny na striktně daných funkčních požadavcích.

### • 4. Fáze - verification

Předposlední fází bylo testování aplikace. Tato fáze měla čtyři úrovně testování, které se prolínaly s třetí fází vývoje.

- Unit testování – pro malé funkční části jako metody webových služeb a testy konzistence databáze.
- Testování vývojářem – každá kousek nově naprogramované části aplikace musela být ověřena přímo vývojářem.
- Uživatelské testování – testování testerem na straně dodavatele aplikace, tedy fyzickou osobou, jinak nezainteresovanou do projektu.
- UAT – User Acceptance Testing – závěrečným dvoutýdenním testováním na straně zadavatele.

### • 5. Fáze - maintenance

Poslední fází projektu je maintenance, která je v případě realizace popisovaného projektu uzavřena na 5 let a zahrnuje legislativní upgrade systému, a technickou podporu po dobu životnosti aplikace.

Dokončení prací znamená přesunout strukturu vývojové databáze na produkční server, zpřístupnění Kraji Vysočina administrační prostředí k „Geo-Hrám“, nasazení webové služby a publikování podepsaných mobilní aplikací na příslušné obchody.



## 4.2 Verzovací systém

Při vývoji aplikace bylo nezbytné využít kromě vývojového prostředí také systém pro správu zdrojových kódů. Při vývoji byl využit verzovací systém od společnosti Microsoft s názvem Team Foundation Server Version Control (dále jen TFVC) na Team Foundation Serveru (dále jen TFS), dnes znám a užíván pod zkratkou VSTS Visual Studio Team Services.

Samotná správa zdrojových kódů je pouze omezenou sadou funkcí a v rámci projektu tak bylo využito i dalších funkcionalit TFS, jako automatické spouštění unit testů, nastavení build serveru, release management a využití Work Items pro projektové řízení a QA (Quality Analysis).

### 4.2.1 TFVC vs. Git

Rozdíl mezi TFVC a Git je v tom, že při TFVC má každý developer pouze jednu verzi každého souboru, kdežto u Git má každý developer svou větev vývoje. Historie souboru u Git systému je sjednocení lokálních a serverových změn, kdežto u TFVC je pouze serverová.

### 4.2.2 Obecné využití

S každou novou fungující částí kódu se dělá upload těchto změn na server. Těmto uploadovaným změnám se vytváří takzvaný changeset, který obsahuje všechny změněné soubory proti aktuálnímu stavu na serveru.

Všechny změny lze zpětně dohledat, porovnat mezi sebou, nebo se vrátit k některé z verzí. Vracet se dají jednotlivé soubory nebo kompletně projekt takzvaně rolloutovat na verzi k určitému changesetu.

## 5 Analýza dat před tvorbou aplikace

Na základě funkčních a technických požadavků stanovených zadavatelem bylo třeba zodpovědět několik základních otázek, které zásadním způsobem ovlivňovaly návrh systému a použité technologie.

Kraj Vysočina poskytl webovou službu se všemi daty, ale některá byla expirovaná, a proto se vytvořila druhá služba hostovaná u ALTAIR SOFTWARE s.r.o., která synchronizuje veškerá data z Kraje Vysočina a protřídí jejich platnost. Toto rozhodnutí padlo kvůli úspoře času, baterie, rychlosti a dalších citlivých elementů týkajících se softwarů na mobilním zařízení.

### 5.1 Analýza technologií

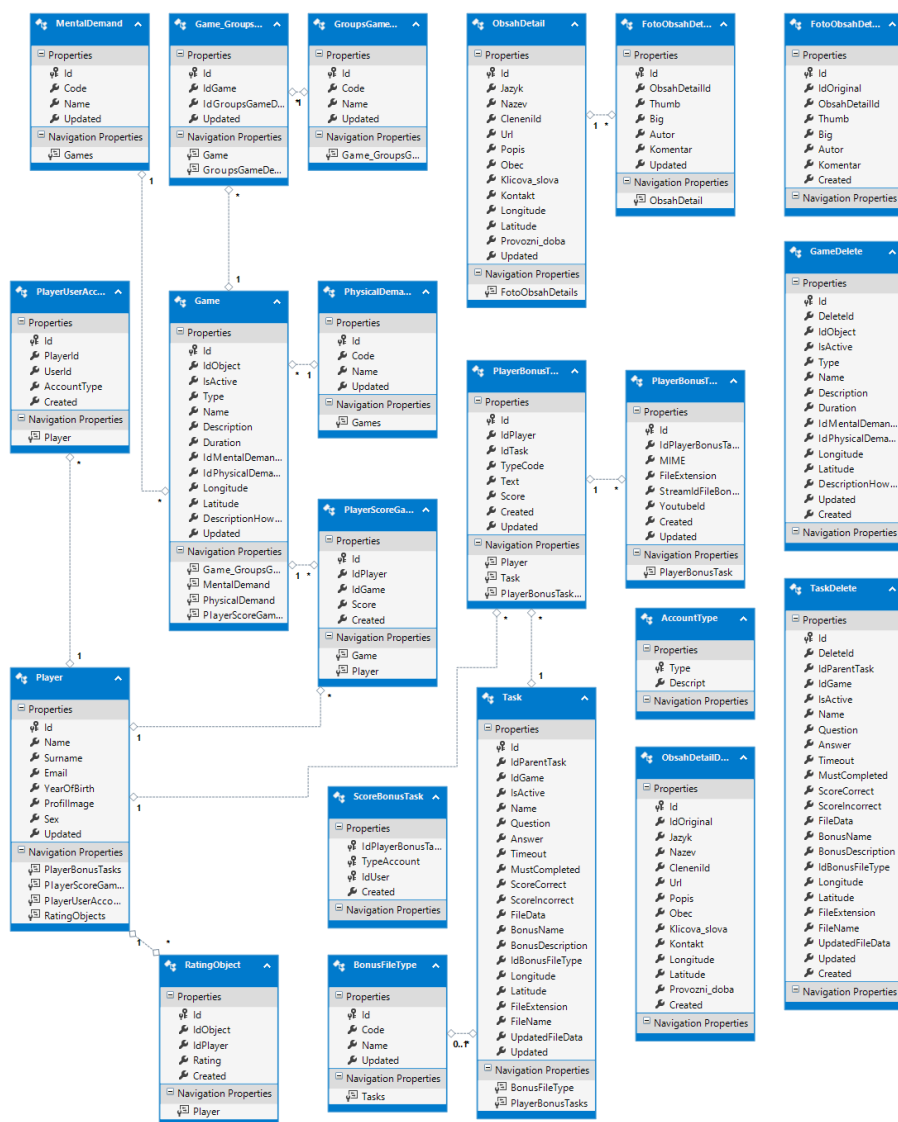
Webová aplikace je velmi jednoduchý redakční systém, na který se použil skriptovací jazyk PHP.

Webová služba stojí na pevném základu platformy .NET s širokou podporou pro webové služby.

Díky naprosto stejnému chování na různých mobilních platformách by šlo využít univerzální platformu Xamarin. Varianta Xamarin nese výhodu úspory času ve smyslu, že aplikace se píše jednou a na konci se vygeneruje do patřičných platforem, konkrétně pro platformu iOS a Android. Ale kvůli špatným zkušenostem při práci s GPS moduly, databázovým přístupem a v neposlední řadě také výsledné velikosti aplikace, se šlo cestou nativních aplikací. Tudíž aplikace pro platformu iOS se psala v jazyce Objective-C a Android v jazyce Java.

## 5.2 SQL databáze

Díky partnerství ALTAIR SOFTWARE s.r.o. a společnosti Microsoft byla rozhodnuto využít jako databázový server MS SQL databázi (dále jen MSSQL) ve verzi 2012 R.2. Databáze přesně kopíruje originální výstup ze služby poskyt-



Obrázek 3: Serverový databázový model.

nuté od Kraje Vysočina, ale navíc obsahuje tabulky s postfixem „Delete“, kde si uchovává expirovaná data. Dále databáze obsahuje tabulky se seznamem registrovaných hráčů, jejich statistiky a hodnocení. Kvůli bezpečnosti je zpřístupnění databázových dat realizováno pomocí webové služby. Datový model je znázorněn na obrázku 3.

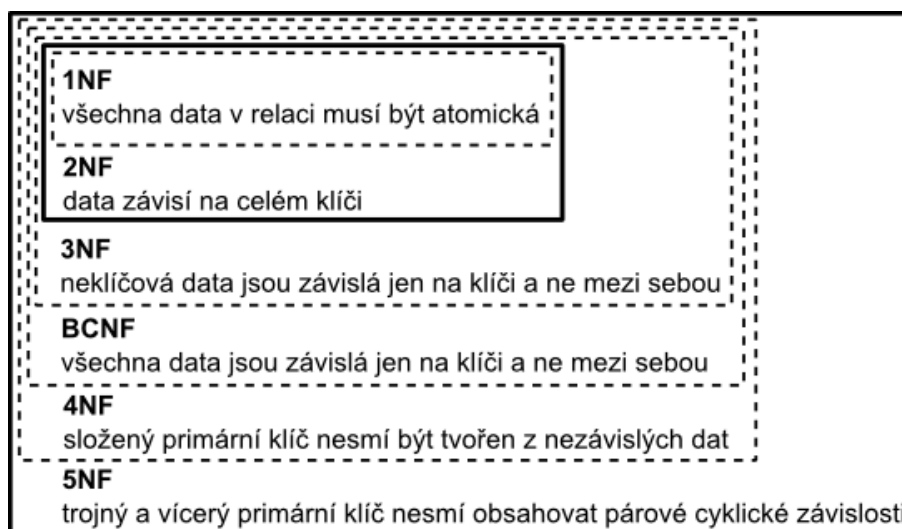
### 5.2.1 MS SQL

Microsoft SQL (Structured Query Language) je relační databáze původně určená obchod, sklady a podobné zaměření. Relační databáze je založená na strukturovaných dat v tabulkách, kde řádky chápeme jako záznamy a sloupce jsou identifikátory. Některé sloupce mohou ukazovat na jinou tabulku, čímž vyniká relace mezi těmito tabulkami. Relaci může tvořit jeden klíč (jeden sloupeček z tabulky), ale klíč k relaci může stanovovat i více sloupečků.

### 5.2.2 Normalizace DB

Normalizací se rozumí proces, ve kterém se snažíme návrh databáze upravovat tak, abychom mohli lépe pracovat a manipulovat s daty, zabránit redundanci dat a zvýšit konzistenci informací napříč databázemi.

Pravidla úpravy databáze se nazývají Normálové formy a obecně platí, že čím vyšší normálová forma, tím lépe navržený databázový model.



Obrázek 4: Databázové normálové formy.

Podkladem pro tuto kapitolu byla [2].

### 5.2.3 Systém auditu nad databází

Tabulky s postfixem „Delete“ reprezentují v databázovém modelu auditní stopu, kdy každý smazaný záznam je zaznamenán v příslušné tabulce s prefixem delete.

## 5.3 Webová administrace

Přes webovou aplikaci 5, která zde reprezentuje specifický redakční systém, lze administrovat seznam jednotlivých „GeoHer“.

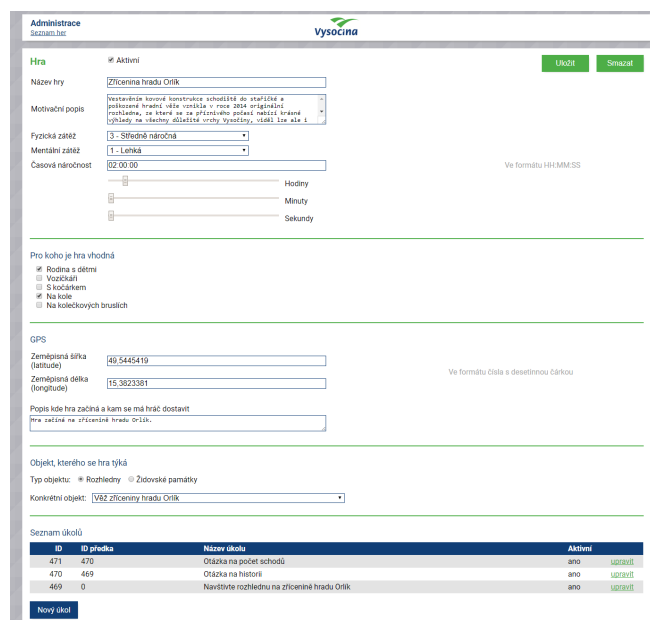
GeoHry mají strukturu hlavního úkolu 6, většinou typu dojít k příslušnému bodu zájmu na dané GPS souřadnice, respektive jeho okruhu, protože z uživatelského hlediska nelze jednoduše přistupovat na přesné souřadnice, a následně několik menších úkolů 7. Například spočítat schody, najít výšku rozhledny a podobně zaměřené úkoly. Každý úkol může obsahovat bonusový úkol, který není povinný a bývá typu zhotovit fotografii, video, nahrát zvuk a jiné média.

### 5.3.1 Ukázka administrace úkolu



ID	Název hry	Popis	Aktivní	
79	Zřícenina hradu Orlik	Vestavěním kovové konstrukce schodiště do stařícké a poškozené hradní věže vznikla v roce 2014 originální rozhledna, ze které se za příznivého počasí nabízí krásné ...	ano	<a href="#">upravit</a>
78	Věž zámku Žirovnice	Raně gotický hrad byl po roce 1550 přestavěn na renesanční zámek a později barokně upraven. V sále jižního křídla a v bývalé kapli je zachován cenný soubor pozdně ...	ano	<a href="#">upravit</a>
77	Věž zámku Police	Zámek je zpřístupněn veřejnosti vždy přes letní měsíce od roku 2005. K vidění jsou zde kvalitní štukové výzdoby stropů, části původního vybavení zámku a mimo ...	ano	<a href="#">upravit</a>
76	Věž kostela sv. Jana Křtitele v Počátkách	Navštivte kostel sv. Jana Křtitele v Počátkách společně s věží, která je od roku 2004 po rozsáhlé rekonstrukci přístupná veřejnosti.	ano	<a href="#">upravit</a>
75	Věž kostela Nanebevzetí Panny Marie v Polné	Navštivte nádherný chrám v Polné, který byl v roce 2008 zapsán do seznamu národních kulturních památek.	ano	<a href="#">upravit</a>
74	Věž hradu Roštejn	Navštivte věž hradu Roštejn - romantický lovecký hrádek v lesích, kde si můžete prohlédnout vnitřní prostory, černou kuchyni, posedět na nádvoří i v hradní hospodě.	ano	<a href="#">upravit</a>
73	Věž hradu Rokštejn	Navštivte zříceninu hradu Rokštejn, která se zvedá na skalnatém vrchu ve výši 677 m v romantických lesích nedaleko Telče.	ano	<a href="#">upravit</a>

Obrázek 5: Seznam her k rozhlednám.



**Hra**  Aktivní [Uložit](#) [Smazat](#)

Název hry:

Motivační popis:

Fyzická zátěž:

Mentální zátěž:

Časová náročnost:  Ve formátu HH:MM:SS

Pro koho je hra vhodná:

- Rodina s dětmi
- Vozíčkáři
- S kočárkem
- Na kole
- Na kolečkových bruslích

GPS:

Zeměpisná šířka (latituda):  Ve formátu čísla s desetinnou částí

Zeměpisná délka (longituda):

Popis kde hra začíná a kam se má hráč dostat:

Objekt, kterého se hra týká:

Typ objektu:  Rozhledny  Židovské památky

Konkrétní objekt:

Seznam úkolů:

ID	ID předka	Název úkolu	Aktivní	
471	470	Otázka na počet schodů	ano	<a href="#">upravit</a>
470	469	Otázka na historii	ano	<a href="#">upravit</a>
469	0	Navštivte rozhlednu na zřícenině hradu Orlik.	ano	<a href="#">upravit</a>

[Nový úkol](#)

Obrázek 6: Úprava hry rozhledny.

Každý z úkolů je ohodnocen body za správnou i špatnou odpověď. V průběhu hry se tyto body sčítají a na konci je skóre uloženo ke konkrétnímu hráči do statistiky.

## 5.4 Webová služba WCF

Abychom si mohli vysvětlit WCF, musíme si vysvětlit pojem SOA (Service Oriented Architecture). Jde o standardizaci metodologií a principů pro výměnu dat mezi aplikacemi nebo službami. Celá architektura mobilních aplikací byla navrhována jako třívrstvá, tedy klient s lokální databází, která pracuje se systémem webových služeb, jež poskytují data z bezpečného databázového úložiště.

Zdrojem byl internet, konkrétně [3].

### 5.4.1 Třívrstvá architektura

Třívrstvá architektura 8 (anglicky Three-layer architecture) označuje jeden specifický typ architektury informačních systémů. Definiuje, jakým způsobem je informační systém rozdělen mezi to, co vidí a používá uživatel (tzv. prezentační vrstva), a to, co se odehrává na pozadí na straně serveru (aplikační a datová vrstva).

- **Prezentační vrstva**

Prezentační vrstva je uživateli viditelná, zajišťuje vstup požadavků a prezentaci výsledků. Je zpravidla na platformě závislá (např. webová aplikace, aplikace pro Windows, Android aplikace atd.).

- **Aplikační vrstva (nebo-li funkční)**

Prostřední vrstva modelu (middleware), zajišťuje výpočty a operace prováděné mezi vstupně-výstupními požadavky a daty. Také nazývána jako aplikační server. Tuto vrstvu v moderních informačních systémech zajišťují právě webové služby.

- **Datová vrstva (také databázová)**

Nejnižší vrstva modelu zajišťuje práci s daty, tedy systém řízení báze dat a základní datově-funkční operace zajišťující ukládání, výběr, agregaci, předzpracování, integritu a audit dat.

### 5.4.2 Historie

Obecně webové služby začali vznikat s příchodem internetu a požadavkem výměny dat mezi aplikacemi. Problém však nastával, když aplikace nebyly postaveny na stejné platformě a například jedna byla postavena na platformě .NET a druhá napsána v Javě na linuxu. Microsoft ve své platformě .NET do verze 2.0 poskytoval několik technologií pro výměnu dat:

Administrace  
Seznam úkolů → 129.22

Vysocma

Úkol

Úkol  Aktivní

Název úkolu

Otázka

Odpověď

Dorazit na místo  Úkol je splněn po dostavení se na místo určené GPS souřadnicemi.

Zeměpisná sířka

Zeměpisná délka

Časový limit

Předcházející úkol (předek)

Úkol musí být splněn aby uživatel mohl dále pokračovat

Skóre

Správná odpověď

Špatná odpověď

Soubor

Soubor nevybrán

Po uložení smazat soubor?

Bonusový úkol

Aktivní  Ano

Název

Popis

Typ dat, které musí hráč dodat

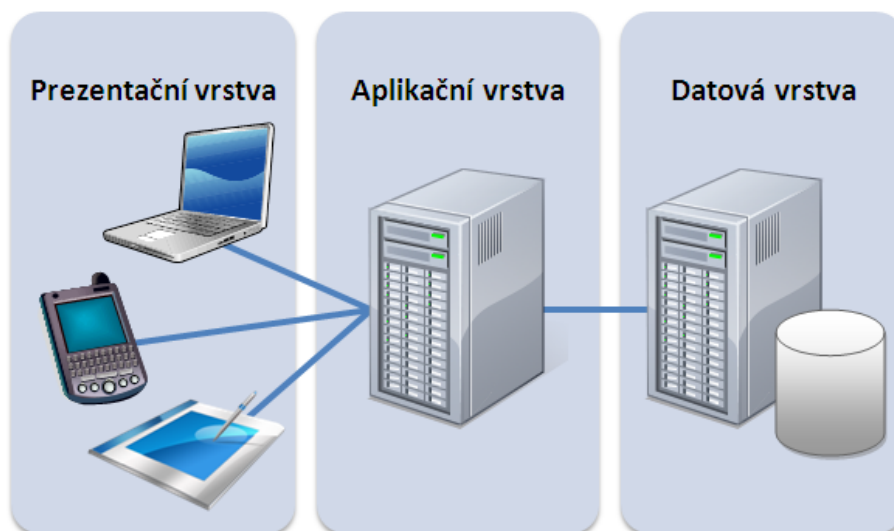
Video

Audio

Obrázek

Text

Obrázek 7: Úprava úkolu rozhledny.



Obrázek 8: Třívrstvá architektura.

- ASP.NET Web Services (ASMX)
- Web Services Enhancements (WSE)
- .NET Remoting
- Microsoft Message Queue (MSMQ)
- Enterprise Services/COM+

Každá ze zmíněných technologií má svá pro a proti. Jedna klade důraz na zabezpečení, kdežto druhá na rychlost.

Zdroje pro webové služby: [3], [7], [8].

### 5.4.3 Co je WCF

WCF (Windows Communication Foundation) je framework sdružující historické SOA modely. Tímto krokem Microsoft odstranil potíže spojené s kompatibilitou a v případě změny technologie, velkým zásahem do již hotové implementace. Jde tedy zjednodušeně říct, že při využití WCF stačí říct, existují data určitého typu a budou posílána konkrétním protokolem.

Každá WCF služba disponuje určitými entitami, které si popíšeme.

- **Endpoint** definuje prostor, kde proběhne výměna dat.
  1. **address** definuje adresu, kde služba běží.
  2. **binding** definuje způsob výměny dat, neboli protokol (JSON nebo SOAP).
  3. **contract** definuje konkrétní data či metody, která jsou službou nabízena.
- **Hostovací prostředí** je prostor, kde je služba umístěna, kde běží. WCF služby mohou běžet na webové službě IIS (Internet Information Service, hostující také webové stránky), jako klasický Windows proces nebo se bude sama hostovat (self-hosting).
- **Kanál** je prostředek pro výměnu dat, který se vytváří v okamžiku navázání spojení mezi klientem a službou, nebo službou a službou.
- **SOAP zprávy** jsou data, která si vyměňuje služba s klientem nebo jinou službou. Jednoduše řečeno jde o zprávy typu otázka, odpověď. Služby lze rozdělit dle posílání a přijímání zpráv na:
  1. **one way** kde klient odešle zprávu a neočekává odpověď.
  2. **two way nebo request and response** jde o klasický model srovnatelný s řečí lidí. Klient pošle otázku (request) a očekává odpověď (response) od serveru.
  3. **duplex** obě strany jsou schopny odeslat request. komunikace probíhá asynchronně.



## 6 Mobilní aplikace GeoHra

Tato kapitola popisuje vlastní mobilní aplikaci pro platformu Android. Verze aplikace pro platformu iOS má totožný datový model, stejný koncept, ale respektovalo se guideline (příručka vývoje) příslušných platform pro ovládací prvky.

### 6.1 Oprávnění aplikace

Každá mobilní aplikace, nezáleží na platformě, vyžaduje k běhu na zařízení určitá oprávnění, se kterými může uživatel souhlasit či nikoliv. Na základě těchto práv se později aplikace chová více přívětivě k uživateli a poskytuje určitý vyšší komfort.

Například při registraci a potřebě ověřit telefonní číslo bývá často vyžadováno oprávnění číst SMS zprávy telefonu. Když se uživatel zaregistruje a přijde mu SMS s ověřovacím kódem a má povolení čtení SMS pro tuto aplikaci, aplikace pozná SMS svým algoritmem a ověří pravost telefonního čísla. Kdežto u zakázaného oprávnění je uživatel vyzván, aby sám přečetl SMS zprávu a kód z ní vepsal do aplikace ručně.

Tohle zní jako fantastická věc pro uživatele, ale je potřeba být ostražitý, protože, jak už to bývá, při velké popularitě čehokoliv, je také velká skupina těch, kteří se snaží škodit. Proto vznikají spousty aplikací tvářících se užitečně, ale ve skutečnosti mohou uživateli uškodit. Pokud uvažujeme výše uvedený příklad aplikace s oprávněním čtení SMS a aplikace bude mít i vedlejší funkci, než kterou popisuje, tak je možné, že při využití internetového bankovníctví bude aplikace rozeznávat i takovéto typy SMS zpráv a může potom jednoduše získat přístup k finančním prostředkům oběti. Je potřeba uvažovat oběma směry a být ostražitý nejen odkud aplikaci stahujeme, ale také jejich smysl. Přestože Google má obrovskou základnu ověřovacích algoritmů aplikací, tak aplikací je stále více a je možné infikovat telefon i z oficiálního zdroje Obchodu Play. Tato možnost je sice mizivá, ale stále zde je.

Android využívá dva typy oprávnění, z nichž jeden *uses-feature* popisuje výstavbu hardware zařízení nebo software součást, která je zapotřebí k instalaci aplikace. Oprávnění *uses-feature* může obsahovat atribut *required*, který toto oprávnění může vypustit a zařízení například bez kompasu, může aplikaci kompas ukazující sever nainstalovat. Následné použití oprávnění se přesto kontroluje ve zdrojovém kódu, kde má dojít k interakci s příslušnou entitou. Ukázka *uses-feature* na části kódu 2.

Druhý typ oprávnění *uses-permission* definuje oprávnění přístupů. Přístup ke kameře, internetu, poloze, atd. Při vývoji se oba typy oprávnění definují do manifestu aplikace. Ukázka kdefinice je na snippetu zdrojového kódu 3.

```
1 <uses-feature
2     android:name="android.hardware.camera"
3     android:required="true"/>
```

Zdrojový kód 2: Ukázka vyžadování hardware komponenty fotoaparátu.

```
1 <uses-permission
2     android:name="android.permission.CAMERA"
3     android:required="true"/>
```

Zdrojový kód 3: Ukázka oprávnění k přístupu fotoaparátu.

### 6.1.1 Oprávnění pro aplikaci GeoHra

Ač se to zdá mnoho, tak aplikace GeoHra vyžaduje přes desítku oprávnění. Veškerá mají své opodstatnění, které budou popsány.

Oprávnění na hardware:

- **android.hardware.camera** podmínka pro rozšířenou realitu
- **android.hardware.Camera.autofocus** možnost zaostření při změně expozice fotoaparátu
- **android.hardware.sensor.accelerometer** potřeba k plnění úkolů s navigací
- **android.hardware.sensor.compass** potřeba k plnění úkolů s navigací

Oprávnění na přístupy:

- **android.permission.CAMERA** možnost zobrazit rozšířenou realitu
- **android.permission.INTERNET** aktualizace databáze s objekty zájmu a her
- **android.permission.ACCESS\_NETWORK\_STATE** umožňuje zjistit, zda-li je zařízení připojeno k internetu
- **android.permission.READ\_EXTERNAL\_STORAGE** čtení offline zálohy databáze a cache obrázků
- **android.permission.WRITE\_EXTERNAL\_STORAGE** zápis cache souborů
- **android.permission.RECORD\_AUDIO** potřeba k plnění úkolů typu nahrát zvukovou stopu

- **android.permission.ACCESS\_COARSE\_LOCATION** urychluje nalezení polohy uživatele na základě triangulace mobilních síťových výsíláčů pro plnění úkolů s navigací
- **android.permission.ACCESS\_FINE\_LOCATION** upřesňuje nalezenou polohu uživatele pro plnění úkolů s navigací
- **android.permission.READ\_PHONE\_STATE** dovoluje číst informace o mobilním zařízení
- **com.google.android.providers.gsf.permission.READ\_GSERVICES** vyžadují Google služby k funkci Google map v aplikaci

## 6.2 Synchronizace dat

Na základě nepsaných zákonů o vývoji mobilních aplikací probíhá synchronizace snapshotů databáze (pouze rozdílová data jsou přenášena, čímž se uspořil čas přenosu a zároveň zpracování).

Aplikace se doptávají již vytvořené webové služby na změny k datu poslední aktualizace dat pomocí protokolu JSON. Dotaz na data probíhá přístupem na adresu služby, zvolením endpointu, zvolením metody služby a data od kdy se data naposledy synchronizovala v zařízení. Na tento dotaz server odpoví seznamem všech změn na dané entitě od zvoleného data.

Pro ukázkou synchronizace dat použijeme XML protokol a dotaz na všechny nové hry od 1. června 2016. Ukázka vstupu 4 a výstupu 5 byla provedena v programu SoapUI 5.3.0 od společnosti SmartBear Software, kam se vloží WSDL, neboli definice služby a na jejím základě, si je schopen program sám poskládat dotazy, kde se doplní pouze hodnoty do parametrů.

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
  envelope/" xmlns:tem="http://tempuri.org/">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <tem:GetGames>
5       <tem:since>2016-06-01</tem:since>
6     </tem:GetGames>
7   </soapenv:Body>
8 </soapenv:Envelope>

```

Zdrojový kód 4: Příklad dotazu na server.

Obdržená data se deserializují do příslušných objektů a provede se buďto vložení, editace, nebo smazání záznamu z lokální databáze na základě volané metody.

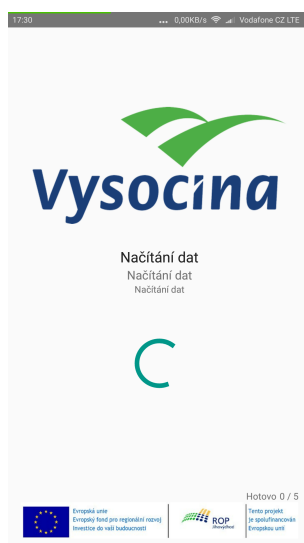
Současně se provádí odesílání dosažených výsledků a pořízených multimediaálních souborů uživatelem aplikace. Veškeré soubory se odesílají na servery

ALTAIR SOFTWARE s.r.o. za kterých se později interní službou odesílají video záznamy na streamovací server. Tato služba se spouští v hodinových intervalech a odesílá veškerá videa na streamovací server a po úspěšném odeslání uloží do databáze odkaz na streamované video. Takto streamovaná videa jsou již přístupná veřejnosti.

Poskytování služeb s daty zajišťují servery ALTAIR SOFTWARE s.r.o. umístěné v datacentru MERIT GROUP a.s., kde jsou připojeny na gigabitové lince přímo do internetu, takže zajišťují dostatek konektivity pro velký nápor dat poskytovaných a zároveň obdržených od uživatelů.

### 6.2.1 Způsob synchronizace

Jak již bylo zmíněno, aplikace po spuštění zobrazí synchronizační obrazovku 9 a vytvoří dotaz na webovou službu na rozdílové změny serverové a lokální databáze. Tyto dotazy jsou realizovány v asynchronním vlákně, kvůli neblokovaní hlavního vlákna a možnosti změny GUI pro informování uživatele o průběhu synchronizace. Po skončení synchronizačního procesu je uživateli zobrazena úvodní obrazovka aplikace 10.



Obrázek 9: Synchronizace aplikace.

## 6.3 Hlavní obrazovka aplikace

Na hlavní obrazovce 10 je uživateli nabídnuta volba „Židovských památek“ či „Rozhleden“, u kterých si uživatel zobrazí detail nebo spustí „geohry“. Dále poskytuje rozcestník do nastavení, kde si uživatel může nastavit přesnost navigačních her. Zde je možnost si vybrat mezi režimy získání souřadnic zařízení dle GPS, které je nejpřesnější, ale spotřebuje nejvíce energie, nebo podle internetové

sítě, či pasivní mód, který závisí na triangulaci polohy dle GSM sítě. Pasivní mód je nejhospodárnější k baterii, ale naopak není tak přesný jako GPS.



Obrázek 10: Hlavní obrazovka aplikace.

## 6.4 Rozhledny

Jedním ze dvou typů objektů, které může uživatel v aplikaci 11 navštívit jsou rozhledny. Každá rozhledna v aplikaci o sobě obsahuje informační text s jasně zmíněnou historií a provozní dobou, kdy se může uživatel přijít podívat na panoramata kraje.

Pro snazší nalezení rozhledny je dostupná funkce mapy poskytované společností Google. Díky přesné poloze uživatele a rozhledny je možné spustit navigaci v externích aplikacích.

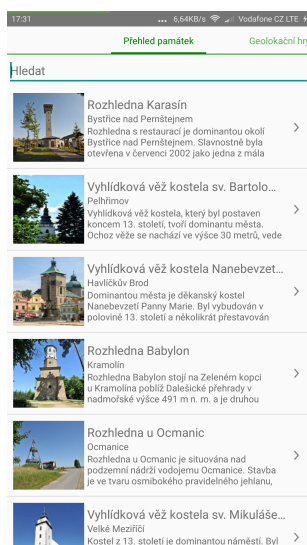
Na každé rozhledně je možné zapnout si rozšířenou realitu a poznat, kterým směrem a jak daleko jsou okolní body zájmu. Rozšířenou realitou se bude zabývat kapitola 6.8.

## 6.5 Židovské památky

Druhým typem uživatelsky přístupných objektů jsou židovské památky. Tento objekt obsahuje totožné informace jako předchozí a proto se v aplikaci jedná o totožný objekt na totožné obrazy 11 pouze s příznakem o který typ objektu se jedná.

## 6.6 GeoHry

Hra jako taková obsahuje souřadnice a metainformace k místu zahájení plnění úkolů. Každý start je podmíněn dostavením se se zařízením na určité místo 13,



Obrázek 11: Přehled památek.

kde se uživatel seznámí s památkami kraje, a následně jsou spuštěny úkoly.

Při výběru hry je uživatel informován o časové, fyzické a locké námaze [12](#). Současně jsou i her piktogramy znázorňující vhodnost dané hry pro handicapované, kočárky, pro brusle a další.

Po výběru hry je uživatel krátce seznámem s danou památkou a zároveň vidí zbývající vzdálenost pro dosažení cíle, ve kterém začne plnit úkoly.

## 6.7 Úkoly

Každý úkol je ohodnocen počtem bodů pro splnění či nesplnění. Při úspěšném zdolání úkolu je uživatel odměněn přičítajícími se body, při nezdárném konci je uživateli odečtena určitá část. Veškeré hodnoty lze v administrativním rozhraní [7](#) dynamicky měnit a do aplikace se dostanou skrze synchronizační službu.

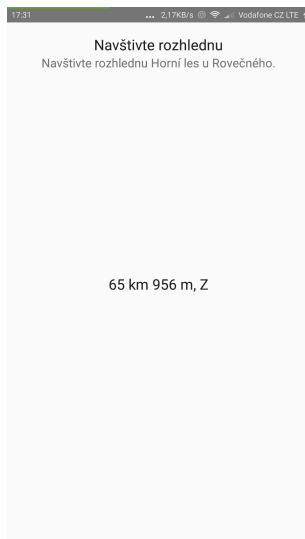
Úkoly jsou rozmanitého typu zadávání výsledků. Co si pod tímto termínem představit? Kraj Vysočina může ve svém administrativním rozhraní vytvořit úkol se zadáváním odpovědi typu *a)*, *b)*, *c)*, nebo celá slova, čísla, ale také pořizovat multimediální záznamy. Uživatel je potom vyzván pro pořízení fotografie, videa či nahrátí zvukové stopy.

Hry se dají hrát offline, neboli v režimu bez internetu a až po připojení k internetu jsou výsledky uživatele odeslány a zařazeny do tabulky hráčů. Stejně tak multimediální záznamy jsou zpracovány po připojení se do internetové sítě.

Po dokončení hry, respektive dohrání všech dílčích úkolů, je uživateli zobrazena tabulka s jeho dosaženými výsledky. Výsledky se uloží do lokální databáze a následně je uživatel přesměrován na detail památky, u níž hrál hru. Z tohoto detailu je možné spustit si rozšířenou realitu.



Obrázek 12: Přehled her.



Obrázek 13: Herní obrazovka - úkol dojít na místo.

## 6.8 Rozšířená realita

Nejpopulárnější, a zároveň dle mého názoru nejdůležitější, popis rozšířené reality definoval pan *Ronald T. Azuma, Ph.D.* v práci *A Survey of Augmented Reality (Azuma, 1997)*.

### Definice 1 (Definice AR)

Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality as it is more commonly called. VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it. Ideally, it would appear to the user that the virtual and real objects coexisted in the same space.

### Definice 2 (Definice AR - překlad)

Augmentovaná Realita (AR) je variace virtuálních prostředí, nebo Virtuálních Realit (VR) jak se běžně nazývá. Technologie virtuálních prostředí kompletně ponoří uživatele do syntetického prostředí. Během ponoření uživatel nevidí skutečný svět kolem sebe. Naproti tomu rozšířená realita dovoluje uživateli vidět skutečný svět s virtuálními objekty, které jsou překládány nebo jsou vloženy do reálného světa. Tedy rozšířená realita spíše realitu doplňuje než ji zcela nahrazuje. V ideálním případě by to bypádalo, že virtuální a skutečné objekty současně existují ve stejném prostoru.

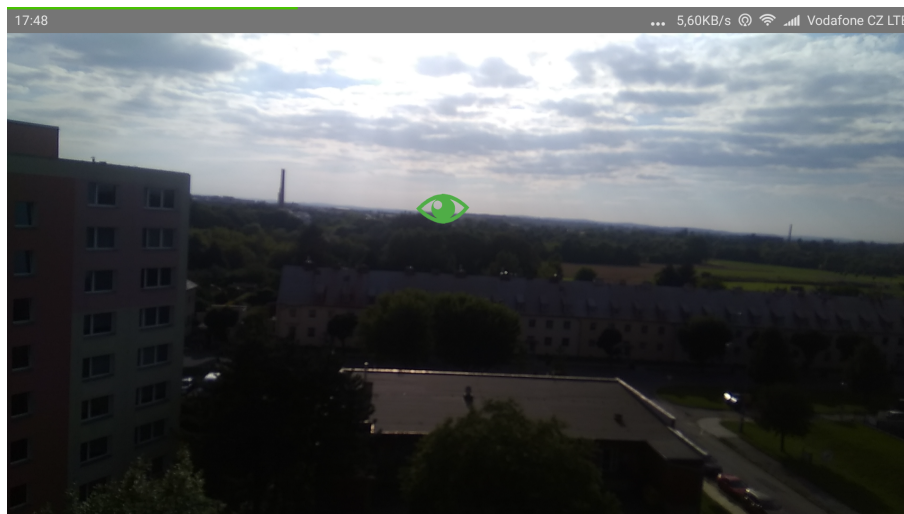
## 6.9 AR v aplikaci

V aplikaci se rozšířená realita realizuje vložení bodu zájmu do obrazu zachycovaného přes zadní fotoaparát mobilního zařízení. Bod zájmu je reprezentován ikonkou 14 a zobrazuje se na základě kompasu, GPS souřadnic a vzdáleností. Body zájmu jsou vkládány do obrazu ve vzdálenosti maximálně 15 kilometrů. Zobrazeny jsou potom ty body zájmu, které jsou v zorném poli zařízení. Zorné pole je určeno GPS pozicí a zároveň kompasem. Výsledná kompozice obrazů vypadá následovně 15.



Obrázek 14: Ikonka bodu zájmu v rozšířené realitě.





Obrázek 15: Rozšířená realita aplikace zobrazující nejbližší body zájmu.

## 6.10 Popis struktury Android aplikace

V kořenovém adresáři projektu nalezneme složky:

- **.gradle** Složka obsahuje gradle build informace, které jsou vytvářeny Android Studiem, respektive během buildu gradle.
- **.idea** Zde ukládá Android Studio metadata k projektu.
- **.navigation** Rozšíření Android Studia Navigation Editor zde ukládá metadata. Navigation Editor se používá k vytvoření flow mezi obrazovkami, nicméně v této aplikaci nebyl Navigation Editor využit, protože v době vývoje neexistoval.
- **app** Jedná se o modul aplikace, v současnosti jediný a výchozí. Obsahuje zdrojové kódy aplikace. Jejím obsahu bude věnována pozornost v jedné z dalších kapitol.
- **build** Uchovává metadata k buildu aplikace.
- **gradle** Zde se nachází sestavený jar balíček pro komunikaci gradle s operačním systémem.

### 6.10.1 Adresář app

Zde se nachází obsah modulu.

- **build** Složka build obsahuje sestavenou verzi zdrojových kódů ve spustitelném balíčku *APK* a další metadata sestavování.



Obrázek 16: Ikonka aplikace GeoHra.

- **libs** Obecně složka obsahuje externí knihovny v balíčcích *jar*. Aplikace GeoHra využívá knihoven, které lze referencovat pomocí maven referencí, ale jedna knihovna, *beyondar-v0.9.3.jar*, podporující rozšířenou realitu, nebyla v maven dostupná, proto byla přidána jako externí zdroj ve výše popsané složce.
- **src** Zde se nachází veškeré kódy modulu aplikace ve starším jazyce pro Android. Starší jazyk je Java, ale Google na své pravidelné konferenci I/O v roce 2017 informoval, že sekundárním jazykem pro psaní aplikací pro Android, se stává Kotlin.

Druhým významným obsahem složky *src* jsou resource v podobě xml souborů definujících vzhled každé aktivity a také ikonky.

### 6.10.2 Ikonky

V aplikaci byly použity ikonky vytvořené výhradně pro účel této aplikace. Ikonky jsou jednoduché a výstižné. Znázorňují obtížnosti hry a doprovází uživatele celou aplikací.

Veškerá grafika byla tvořena s ohledem na grafický manuál kraje Vysočina, kde jsou předepsány barvy, vzdálenosti vztahující se k logu kraje. Manuál dal grafikům ALTAIR SOFTWARE s.r.o. inspiraci pro barevné provedení.

Ikonka pro aplikaci [16](#) respektovala barvy a zaměření aplikace, proto dominantním prvkem jsou špendlíky z mapy znázorňující body zájmu. Body zájmu jsou v kruhu, který sdružuje veškeré elementy ikonky a naznačuje ucelenost.

## 6.11 Popis tříd

Vyvíjená aplikace spadá pod jmenný prostor (namespace) skládajícím se z prefixu společnosti a samotné aplikace: `cz.altairsoftware.vysocina.geohra`. Veškeré zdrojové kódy jsou zanořeny v tomto namespace a následně dle jejich funkce o jednu úroveň níže. Například veškeré aktivity, neboli obrazovky aplikace,

se nachází v `cz.altairsoftware.vysocina.geohra.Activities`. Podobně existují i `Adapters`, `DataObjects`, `Fragments`, `Services` a `Utils`. Následuje popis tříd v jmenných prostorech.

### 6.11.1 Namespace `.Activites`

- **AboutActivity.java** Obrazovka zobrazující informace o aplikaci. Informace jsou pouze statické.
- **AugmentedActivity.java** Implementuje knihovnu Beyonard s objekty staženými z webové služby. Z lokální databáze jsou vylistovány a následně vloženy do rozšířené reality jen ty objekty, které jsou v okruhu patnácti kilometrů od aktuální polohy zařízení.

Zde musím podotknout, že kvůli testování jsou do aplikace na pevně vloženy dva objekty z Olomouce. Konkrétně se jedná o „Sloup nejsvětější trojice“ na souřadnicích *49.593937* zeměpisné šířky a *17.250440* zeměpisné délky. Druhým objektem pro testování se stalo Náklo, pro prověření zmenšování ikonky bodu zájmu na základě vzdálenosti, na souřadnicích *49.651323*, *17.123282*.

- **DetailActivity.java** Aktivita zobrazuje detail objektu, na který se dostal uživatel kliknutím z listu objektů. Aktivita umožňuje zobrazení fotogalerie, spuštění rozšířené reality a zobrazení objektu na mapě s možností spustit navigaci na místo.
- **MainActivity.java** Jde o hlavní obrazovku, na které je rozcestník židovských památek a rozhleden. Je zde možnost se přihlásit na Facebook, který je později využit jako identifikátor ke statistikám her. Hlavní obrazovka nabízí možnost přejít na nastavení či informace o aplikaci.
- **MapsActivity.java** MapsActivity implementuje mapy od Google, které se zobrazují do fragmentu. Fragment je jediný element vzhledu této aktivity. Do map se vloží špendlík s umístěním památky a mapa je automaticky přiblížena k tomuto bodu zájmu.
- **ObjectsGamesActivity.java** Po zvolení typu památek z úvodní obrazovky jsou vylistované památky zobrazeny na této obrazovce spolu s příslušnými hrami. Pro zpřehlednění obrazovky ObjectsGamesActivity bylo použito ViewPager, což jsou dva či více fragmentů vedle sebe, ale v tomto případě dva, a lze se mezi nimi navigovat posunutím, nebo klikem v menu nahoře. Toto chování je pro uživatele Android velmi dobře známo z Obchodu Play, kde jsou k dispozici miliony aplikací pro platformu Android.

Uživatel má možnost památky filtrovat dle názvu v horní části fragmentu. Při výběru památky je uživateli zobrazena aktivita s detailem památky. Pokud zvolil uživatel hru, je zobrazena TaskActivity, kde začíná s hraním hry.

- **SettingsActivity.java** Jak již říká název aktivity, jedná se o obrazovku s nastavením chování aplikace. Uživatel si je schopen nastavit, jaká přesnost polohy má být využita při hraní geohér. Přesnost polohy má přímoúměrný vliv na spotřebu baterie, neboli čím vyšší přesnost, tím vyšší spotřeba.

Dále je zde uživateli nabídnuta možnost stahovat aktualizace při každém spuštění aplikace. Jestli stahování bude probíhat jen na wifi sítích, nebo může aplikace využít i mobilní data. Uživateli je také nabídnuta možnost vynutit stažení aktualizace okamžitě.

- **StartActivity.java** Po spuštění aplikace se jako úplně první aktivita spustí StartActivity, na které aplikace prověří preference uživatele z nastavní, případně stáhne aktualizace z webových služeb.
- **TaskActivity.java** Aktivita obsluhuje celé herní dění. Při spuštění hry je v drtivé většině vyžadováno, aby uživatel stál v okruhu dané památky, nebo je vyzván, aby se dostavil na toto místo. Po splnění prvního úkolu následují další. Může jít o úkol, kde je uživatel vyzván k doplnění čísla, písmenka, odpovědi nebo nahrání multimediálního souboru. Všechny soubory, které jsou v rámci aplikace nahrávány, jsou realizovány službou Android, jež bude v zápětí popsána.

### 6.11.2 Namespace .Adapters

- **GameListAdapter.java** Zajišťuje zobrazení listu her.
- **ImagePagerAdapter.java** Adapter zobrazující obrázky
- **ObjectDetailListAdapter.java** Zobrazuje detail památek. Vzhledem k tomu, že rozdíl mezi židovskou památkou a rozhlednou je pouze v typu, obě památky jsou reprezentovány jednou třídou (ObsahDetail), a proto je může zobrazovat jeden adapter listu.
- **ScreenSlidePagerAdapter.java** Tento adaptér je použit na aktivitě ObjectsGamesActivity, která zobrazuje dva fragmenty. Jeden fragment s židovskými památkami nebo rozhlednami, druhý s geohrami.

### 6.11.3 Namespace .DataObjects

- **FotoObsahDetail.java** Třída držící informace o jedné fotografii k objektu ObsahDetail.
- **Game.java** Třída s popisem hry, její náročností a seznamem dílčích úkolů.
- **ObsahDetail.java** ObsahDetail je hlavní třídou obsahující informace o památce. Jak již bylo zmíněno dříve, tak židovské památky i rozhledny jsou v aplikaci reprezentovány třídou ObsahDetail a rozděleny pouze attribute *Type*. Třída obsahuje také list FotoObsahDetail.

- **ParentObject.java** ParentObject je abstraktní třída, kterou ostatní třídy z DataObjects jmenného prostoru implementují.

ParentObject obsahuje abstraktní metody, které každá implementující třída musí přepsat. Jedná se o metody serializace a deserializace dat. Také obsahuje metodu save, která na základě serializovaných dat uloží objekt do databáze. Pokud objekt již existuje, provede update.

- **Player.java** Identifikuje uživatele při hraní geoher a ve statistikách.
- **PlayerGameTaskInProgress.java** Při splnění dílčího úkolu hry se ukládá postup ve hře touto entitou pro případ nepředvídatelného problému, například vybití baterie mobilního zařízení. Po novém spuštění hry se hráč ocitne zpět v posledním stavu, aby neztratil své výsledky.
- **PlayerScoreGame.java** Po ukončení hry se ukládá skóre do databáze a posílá se výsledek na webové služby do žebříčku všech her.
- **Task.java** Objekt obsahující informace o dílčím úkolu.

#### 6.11.4 Namespace .Fragments

- **GameListViewFragment.java** Obsahuje list geroher.
- **ObjectListViewFragment.java** Zobrazuje všechny památky, vylistované dle typu.
- **PhotoSlidePageFragment.java** Fragment zobrazující obrázky a umožňuje fullscreen náhled obrázku.

#### 6.11.5 Namespace .Services

- **SendFilesIntentService.java** Zajišťuje posílání souborů na webovou službu.

#### 6.11.6 Namespace .Utils

- **AsyncJson.java** Asynchronní třída zajišťující synchronizaci mezi webovou službou a lokální databází aplikace.
- **CompressImage.java** Asynchronní třída upravující velikost pořízené fotografie na menší rozlišení pro úsporu uživatelského FUP (Fair User Policy - síťovým operátorem nastavený limit dat).
- **DatabaseHelper.java** Pomocná třída pro práci s databází.
- **SubmittingType.java** Třída uchovávající konstanty.
- **UILBaseImageDownloader.java** Pomocná třída pro stažení obrázků k památkám.

- **UtilsDatabase.java** Třída s konstantami SQL příkazů tvorby databáze.
- **UtilsFacebook.java** Pomocná třída pro získání uživatelských dat z Facebooku.
- **UtilsFiles.java** Pomocná třída pracující s mediálními soubory.
- **UtilsNetwork.java** Pomocná třída pro práci s internetovou sítí.
- **UtilsObjects.java** Pomocná třída doplňující informace pro objekty `Ob-sahDetail`.
- **UtilsPlayer.java** Pomocná třída sdružující práci s hráčem.
- **UtilsServer.java** Pomocná třída pracující se synchronizovanými daty a lokálním databázovým obrazem.
- **UtilsSettings.java** Pomocná třída k uživatelskému nastavení. Logika uklá-dání a čtení uložených hodnot je v `UtilsSettings`, kdežto uživateli je repre-zentována `SettingsActivity`.
- **UtilsViews.java** Pomocná třída nastavující parametry stahování obrázků, dialogový kontext atd.

```

1 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
2   <s:Body>
3     <GetGamesResponse xmlns="http://tempuri.org/">
4       <GetGamesResult xmlns:a="http://schemas.datacontract.org
5         /2004/07/VysocinaGeoGame.Database" xmlns:i="http://www.w3.
6         org/2001/XMLSchema-instance">
7         <a:Game>
8           <a:CodeGroupsList xmlns:b="http://schemas.microsoft.com
9             /2003/10/Serialization/Arrays">
10            <b:string>family</b:string>
11            <b:string>roller_skates</b:string>
12          </a:CodeGroupsList>
13          <a:Description>...</a:Description>
14          <a:DescriptionHowGetStart>Hra začíná v Boubinském pralese
15            u třetího smrku od cesty....
16          </a:DescriptionHowGetStart>
17          <a:DurationGame>02:00:00</a:DurationGame>
18          <a:Id>54</a:Id>
19          <a:IdObject>872</a:IdObject>
20          <a:IsActive>>false</a:IsActive>
21          <a:Latitude>49.5907189</a:Latitude>
22          <a:Longitude>17.2544133</a:Longitude>
23          <a:MentalDemands>medium</a:MentalDemands>
24          <a:Name>Testovací - Boubínský prales</a:Name>
25          <a:PhysicalDemands>extremely_difficult</a:PhysicalDemands
26          >
27          <a>Type>rozhledny</a>Type>
28        </a:Game>
29        <a:Game>
30          ...
31        </a:Game>
32      </GetGamesResult>
33    </GetGamesResponse>
34  </s:Body>
35 </s:Envelope>

```

Zdrojový kód 5: Příklad odpovědi ze serveru.

## Závěr

Práce naplnila očekávání zadavatele, který dodané dílo akceptoval v předpokládaném termínu a v celém rozsahu bez výhrad. V průběhu realizace se celý realizační tým potýkal s nedostatečnou dokumentací a neochotou ke spolupráci třetích stran. Aplikace také získala čestné ocenění v kategorii turistických aplikací, které každoročně vyhlašuje agentura Czech Tourism.



## **Conclusions**

The bachelor's work fulfilled the expectations of the contracting authority, who accepted the delivered work in the expected date and in the whole extent without reservations. During the implementation, the entire implementation team was struggling with insufficient documentation and unwillingness to cooperate with third parties. The application has also been awarded an honorary award in the category of tourist applications announced annually by Czech Tourism Agency.

## A Obsah přiloženého DVD

Na samotném konci textu práce je uveden stručný popis obsahu přiloženého CD/DVD, tj. jeho závazné adresářové struktury, důležitých souborů apod.

### **doc/**

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

### **src/**

Kompletní zdrojové texty aplikace VYSOČINA GEOHRA / vygenerovaná aplikace ve formátu apk, která je připravena k instalaci v mobilních zařízeních Android.

### **readme.txt**

Instrukce pro instalaci a spuštění aplikace VYSOČINA GOHRA, včetně všech požadavků pro jeho bezproblémový provoz. / Instrukce pro instalaci aplikace a popis testovacích objektů pro testování při tvorbě posudků práce a pro účel obhajoby práce.

U veškerých cizích převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovoluují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na CD/DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.

## Bibliografie

- [1] MOBILMANIA, Jan Láska: Windows a BlackBerry OS skomírají. Nemají ani 1 % trhu. [online] 24. května 2016 dostupné z: <https://www.mobilmania.cz/bleskovky/windows-a-blackberry-os-skomiraji-nemaji-ani-1-trhu/sc-4-a-1334529/default.aspx>
- [2] PROGRAMUJTE, Jakub Kulhan: [online]. 23. července 2008 dostupné z: <http://programujte.com/clanek/2008071900-normalizace-relacnich-databazi/>
- [3] NETSTUDENT, Kryštof Laryš: WCF PRO ZAČÁTEČNÍKY – 1. DÍL: TEORIE, ZÁKLADNÍ POJMY. [online] 3. února 2009 dostupné z: <http://www.netstudent.cz/Articles/wcf-pro-zacatecniky-1-dil-teorie-zakladni-pojmy>
- [4] WIKIPEDIA, Wikipedia: Waterfall model. [online] 21. července 2017 dostupné z: [https://en.wikipedia.org/wiki/Waterfall\\_model](https://en.wikipedia.org/wiki/Waterfall_model)
- [5] WIKIPEDIA, Wikipedia: Android version history [online] 2. srpna 2017 dostupné z [https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history)
- [6] TECHRADAR, Gareth Beavis: A complete history of Android [online] 23. září 2008 dostupné z: <http://www.techradar.com/news/phone-and-communications/mobile-phones/a-complete-history-of-android-470327>
- [7] WIKIPEDIA, Wikipedia: Windows Communication Foundation [online] 1. července 2017 dostupné z: [https://en.wikipedia.org/wiki/Windows\\_Communication\\_Foundation](https://en.wikipedia.org/wiki/Windows_Communication_Foundation)
- [8] WIKIPEDIA, Wikipedia: Service-oriented architecture [online] 31. července 2017 [https://en.wikipedia.org/wiki/Service-oriented\\_architecture](https://en.wikipedia.org/wiki/Service-oriented_architecture)