



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**MOBILNÍ APLIKACE PRO HLEDAČE POKLADŮ**

MOBILE APPLICATION FOR TREASURE HUNTERS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ROBIN SKALIČAN**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JIŘÍ HYNEK, Ph.D.**

**BRNO 2020**

## Zadání bakalářské práce



23127

Student: **Skaličan Robin**  
Program: Informační technologie  
Název: **Mobilní aplikace pro hledače pokladů**  
**Mobile Application for Treasure Hunters**

Kategorie: Informační systémy

Zadání:

1. Seznamte se s problematikou hledání historických předmětů pomocí detektoru kovu.
2. Seznamte se s principem tvorby aplikací pro mobilní zařízení s operačním systémem Android.
3. Analyzujte požadavky hledačů historických předmětů s detektorem kovu. Analyzujte existující aplikace, které usnadňují činnost hledačů (práce s historickými mapami, zaznamenávání nálezů, komunikace s muzei, apod.).
4. V souladu s analýzou požadavků navrhnete mobilní aplikaci, která usnadní činnost hledačů historických předmětů s detektorem kovu.
5. Navrženou aplikaci implementujte.
6. Otestujte použitelnost výsledné aplikace, navrhnete možná rozšíření.

Literatura:

- Android.com: *Android Developers: Documentation for app developers* [online]. 2019 [cit. 2019-10-13]. Dostupné z: <https://developer.android.com/docs>
- Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Burlington: Morgan Kaufmann Publishers/Elsevier, 2010, ISBN: 978-0-12-375030-3.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hynek Jiří, Ing., Ph.D.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 31. července 2020

Datum schválení: 16. října 2019

## Abstrakt

Cílem této bakalářské práce je návrh a implementace mobilní aplikace pro zařízení s operačním systémem Android. Aplikace má ulehčit práci se záznamem nálezů hledačům s detektorem kovu při svém hledání. Cílem je také usnadnit další komunikaci s archeology ohledně nálezů. Při zjišťování požadavků uživatelů a kontaktování archeologů vznikl požadavek na rozšíření zadání o webovou aplikaci. Její implementace umožňuje zaznamenaná data z mobilní aplikace zobrazit a zjednodušuje případnou další komunikaci s nálezcem. Tato data jsou uložena v databázi Firebase Realtime Database. V mobilní aplikaci je implementované mapové zobrazení pomocí knihovny Leaflet a funkce vycházející z požadavků uživatelů. Pro většinu implementace byly použity programovací jazyky Java a JavaScript. Výsledkem mé práce jsou dvě aplikace pracující se společnou databází.

## Abstract

The goal of this bachelor's thesis was the design and implementation of a mobile application for devices with the Android operating system. The application intends to simplify the work of findings documentation for searchers with a metal detector. The goal is also to provide an easier way for further communication with archaeologists about the findings. A request was created to extend the thesis assignment by a web application on meetings with archaeologists. The web application displays the recorded data from the mobile application and simplifies any further communication with the data author. This data is stored in the Firebase Realtime Database. The mobile application implements a map view using the Leaflet library and functions based on user requirements. Java and JavaScript programming languages were used for most of the implementation. The result of my work is composed of two applications working with the shared database.

## Klíčová slova

mobilní aplikace, databáze, webová aplikace, archeologie, mapy, Leaflet, Android, Java, JavaScript, Firebase, detektor kovu

## Keywords

mobile application, database, web application, archeology, maps, Leaflet, Android, Java, JavaScript, Firebase, metal detector

## Citace

SKALIČAN, Robin. *Mobilní aplikace pro hledače pokladů*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

# Mobilní aplikace pro hledače pokladů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka, Ph.D. Další informace mi poskytl pan Mgr. Balázs Komoróczy, Ph.D., vedoucí střediska pro výzkum doby římské a doby stěhování národů archeologického ústavu AV ČR, Brno. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
Robin Skaličan  
30. července 2020

## Poděkování

Rád bych poděkoval svému vedoucímu práce panu Ing. Jiřímu Hynkovi, Ph.D. za odborné vedení, čas, ochotu a cenné rady při tvorbě této práce. Dále bych zde chtěl poděkovat panu Mgr. Balázsi Komoróczymu, Ph.D. za jeho čas pro nastínění celé problematiky z pohledu archeologů a za návrh rozšíření zadání v podobě webové aplikace. V neposlední řadě bych zde rád poděkoval své rodině a přítelkyni za jejich velkou a nenahraditelnou podporu.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Archeologie</b>	<b>4</b>
2.1	Věda . . . . .	4
2.1.1	Nález . . . . .	4
2.2	Detektor kovu . . . . .	5
2.2.1	Historie . . . . .	5
2.2.2	Princip . . . . .	5
2.3	Problematika . . . . .	5
2.3.1	Běžný postup . . . . .	5
2.3.2	Pohled archeologů . . . . .	6
2.3.3	Ústavní pohled . . . . .	7
2.3.4	Shrnutí . . . . .	7
<b>3</b>	<b>Analýza problému</b>	<b>8</b>
3.1	Požadavky . . . . .	8
3.1.1	Detektoráři . . . . .	8
3.1.2	Archeologové . . . . .	8
3.2	Existující řešení pro detektoráře . . . . .	9
3.2.1	Mapové aplikace . . . . .	10
3.2.2	Pomocné aplikace . . . . .	12
3.2.3	Shrnutí . . . . .	13
<b>4</b>	<b>Vývoj mobilní aplikace pro Android</b>	<b>15</b>
4.1	Operační systém Android . . . . .	15
4.2	Verze Androidu . . . . .	15
4.3	Jazyk pro vývoj . . . . .	16
4.3.1	Java . . . . .	16
4.3.2	Kotlin . . . . .	16
4.3.3	C++ . . . . .	16
4.4	Aktivita . . . . .	16
4.5	Skládování dat . . . . .	17
4.6	Nástroje . . . . .	18
4.6.1	Google Play . . . . .	19
4.6.2	Firebase . . . . .	19
4.6.3	Android Studio . . . . .	19
4.7	Mapové API . . . . .	19
4.7.1	Google Maps API . . . . .	20

4.7.2	Leaflet . . . . .	20
4.7.3	OpenLayers . . . . .	20
<b>5</b>	<b>Návrh</b>	<b>21</b>
5.1	Architektura . . . . .	21
5.1.1	Model . . . . .	21
5.2	Mobilní aplikace . . . . .	22
5.2.1	Požadavky . . . . .	22
5.2.2	Uživatelské rozhraní . . . . .	22
5.3	Databáze . . . . .	25
5.3.1	Vlastní server . . . . .	25
5.3.2	Existující služby . . . . .	25
5.3.3	Uložená data . . . . .	26
5.3.4	Autorizace . . . . .	27
5.3.5	Úložiště . . . . .	27
5.4	Webová aplikace . . . . .	27
5.4.1	Uživatelské rozhraní . . . . .	27
<b>6</b>	<b>Implementace</b>	<b>29</b>
6.1	Mobilní aplikace . . . . .	29
6.1.1	Programovací jazyk . . . . .	29
6.1.2	Mapy . . . . .	29
6.1.3	Přehled aktivit . . . . .	33
6.1.4	Lokální úložiště . . . . .	36
6.1.5	Uživatelské rozhraní . . . . .	37
6.2	Databáze . . . . .	37
6.2.1	Realtime Database . . . . .	37
6.2.2	Úložiště . . . . .	38
6.2.3	Oprávnění . . . . .	38
6.3	Webová aplikace . . . . .	39
6.3.1	Použité technologie . . . . .	39
6.3.2	Propojení s databází . . . . .	39
6.3.3	Funkce . . . . .	39
6.3.4	Uživatelské rozhraní . . . . .	40
<b>7</b>	<b>Testování</b>	<b>41</b>
7.1	Mobilní aplikace . . . . .	41
7.1.1	Vlastní testování . . . . .	41
7.1.2	Uživatelské testování . . . . .	42
7.1.3	Shrnutí . . . . .	42
7.2	Webová aplikace . . . . .	43
<b>8</b>	<b>Závěr</b>	<b>44</b>
	<b>Literatura</b>	<b>45</b>
<b>A</b>	<b>Uživatelské rozhraní</b>	<b>47</b>
<b>B</b>	<b>Obsah přiloženého DVD</b>	<b>50</b>

# Kapitola 1

## Úvod

Amatérským hledačem pokladů se dnes dá stát celkem snadno. Běžnou vstupenkou do této oblasti je investice do detektoru kovu. Málokterý čerstvý hledač avšak ví, jak vlastně hledat, na co si dát pozor a jak vůbec při i po výkopu správně postupovat. Samotná činnost hledání za pomoci detektoru není bez souhlasu archeologa ústavně dovolena. Přestože dnes existují aplikace na nejrůznější účely, neexistuje taková, která by se na tuto pracovní komunikaci s archeology zaměřila. Cílem této bakalářské práce je vytvořit mobilní aplikaci, která bude tyto aspekty „detektorářství“ zahrnovat a usnadní uživatelům komunikaci s archeology, místo pracovního hledání pověřených osob a pokusů o jejich zastížení.

Několik aplikací zaměřujících se hledače existuje a cílí především na zaznamenávání údajů o nálezech na mapě. Jedná se však o jednoduché aplikace s pouze omezenou funkcí. Proto, jakožto člen zájmové skupiny, jsem si navrhl toto téma, abych vytvořil užitečnou a ucelenou aplikaci, kombinující zajímavé prvky ostatních implementací s vlastními nápady.

Důvod pro výběr implementace právě pro mobilní zařízení spočívá v usnadnění a zpřesnění zaznamenaných údajů v polních podmínkách, protože samotné mobilní zařízení má tyto údaje většinou k dispozici. Vzhledem k dnešnímu širokému rozšíření telefonů se jedná o ideální řešení.

Práce je rozdělena do několika samostatných kapitol a podkapitol. Prvním důležitým krokem (popsaným v kapitole 2) bylo zjistit od hledačů jejich požadavky, zejména na funkčnost. Dále diskutovat s archeology připomínky k návrhu aplikace, metody a postupy pro šetrné zacházení s nálezy i při samotném výkopu. Kapitola 3 je zaměřena na zkoumání existujících řešení pro záznam nálezů, jejich funkcí a použitých technologií. Pojednává také o požadavcích na řešení, které vyplývají z komunikace s cílovými uživateli. Kapitola 4 se věnuje nastudování procesu vývoje mobilních aplikací pro operační systém Android. Zde je kladen důraz především na metody a sadu technik pro tvorbu této implementace. Cílem kapitoly je také studie mapových knihoven vhodných pro platformu Android. Kapitola 5 pojednává o vytvořeném návrhu aplikace a grafickém rozhraní na základě znalostí z předchozích kapitol. Kapitola 6 hovoří o samotné implementaci. Kapitulu 7 tvoří testování a verifikace implementace s pomocí testování na uživateli. Kapitola 8 obsahuje finální zhodnocení dosažených cílů.

## Kapitola 2

# Archeologie

Kapitola se zaměřuje na definici pojmů archeologie a nález, popisuje princip detektoru kovu a problematiku při hledání s jeho využitím z pohledu archeologů, ústavy i běžného amatérského. Také analyzuje vzniklé problémy při kolizi názorů nebo postupu nakládání s nálezy.

### 2.1 Věda

Zdroj [3] definuje archeologii jako vědní obor, který se věnuje zkoumání nejstarších dějin lidstva. Nejedná se o obor s cílem co nejvíce naplnit muzejní vitríny různými předměty. Objevuje, zachraňuje, odborně zpracovává a zveřejňuje veškeré hmotné pozůstatky, které nesou nějakou nepsanou informaci o minulém lidském světě. Na jejich základě se pokouší rekonstruovat politické, ekonomické, společenské a přírodní poměry v jednotlivých úsecích dějinného vývoje.

#### 2.1.1 Nález

Archeologické nálezy jsou nejen ozdobou muzejních vitrín, ale především nositeli celého komplexu důležitých informací. Bez těchto dat není schopna archeologická a historická věda naplňovat své společenské poslání, není schopna plnohodnotně rekonstruovat život dřívějších obyvatel naší země. Jednotlivé archeologické nálezy, bez ohledu na jejich kvalitu, rozměry, krásu či materiál, jsou základními střípky do mozaiky o minulých kulturách a civilizacích na našem území, kterou odborníci postupně skládají. Informační hodnota předmětů v zemi však závisí na mnoha okolnostech:

- jak s nimi nakládali jejich tvůrci,
- co dovolily přežít klimatické a půdní podmínky,
- jak jsou ze země vyzvednuty.

Tento třetí aspekt je mimořádně důležitý a lze jej znázornit na jednom příkladu: Najde-li někdo na svém poli vyoranou bronzovou sponu např. z doby římské a odnese ji domů, stane se z ní zajímavá ozdoba jeho domácnosti. Informační hodnota takového předmětu pro poznání minulosti je však prakticky nulová. Jestliže ji později odevzdá do muzea, stane se součástí sbírky bez možnosti hlubší výpovědi. Pokud je však takový nález přesně zaměřen



a zdokumentován, na místo je přivolán archeolog nebo alespoň neprodleně obdrží nalezený předmět i s dokumentací, lze pak prozkoumat i jeho okolí (např. pro laika zdánlivě nezajímavé či nepostřehnutelné předměty, zbarvení půdy atd.). Mohou tak být objeveny dosud neznámé pozůstatky sídliště či pohřebiště. Zjednodušeně lze říci, že pro smysluplné zkoumání minulosti mají nálezové okolnosti stejný význam jako nález samotný.

## 2.2 Detektor kovu

Drtivá většina všech hledačů historických předmětů v přírodě používá jako pomůcku detektor kovu. Jeho použití je jednoduché stejně jako jeho princip. Některé nálezy jsou i na povrchu a není potřeba jakéhokoli specializovaného nástroje, ale četnost takových nálezů v poměru s úspěšností při využití detektoru je mizivá.

### 2.2.1 Historie

Nejedná se o vynález moderní doby, protože první pokusy o sestrojení sahají až do 19. století [5]. Prvotní účel byl najít přítomnost kovu v rudě, který by zjednodušil jeho pracné manuální hledání nebo v medicíně pro nalezení kovových předmětů v lidském těle.

K výraznému zdokonalení a používání detektoru došlo až během druhé světové války s úmyslem přesně a bezpečně lokalizovat miny [18]. Stále se jednalo o těžké nástroje s vysokou spotřebou elektrické energie, která se musela nosit v podobě velkých baterií. S vynálezem tranzistorů se velikost a váha detektoru zmenšila.

Moderní mobilní detektory, určené pro hledání kovu v půdě, používají integrované obvody a některé modely umožňují uživateli nastavit citlivost cívky, filtrování určitých kovů a zobrazit lokalizaci pomocí zabudované GPS.

### 2.2.2 Princip

Nejpoužívanější technologií detekce kovu v detektorech je VLF (Very Low Frequency), která používá dvě cívky [7, 23]. První cívka vysílá magnetické pole na určenou oblast, které při kontaktu s kovem přenesou elektrickou energii do kovu. Druhá cívka se chová jako přijímač a detekuje změny ve vysílání magnetického pole pomocí sledování jeho napětí.

Po detekování vyšle ovládací jednotce přijatý oslabený signál a ta ho interpretuje uživateli pomocí zvuku nebo informace zobrazí na displeji. Moderní detektory (obrázek 2.1) většinou disponují technologiemi, které signály dokáží rozlišit a uživateli tak zobrazit informace o odhadované hloubce nálezů nebo druhu kovu.

## 2.3 Problematika

Vstupenkou na pole amatérské archeologie je nejčastěji vlastnictví detektoru kovu. Účel samotného hledání se ale liší, stejně jako názor archeologů na tuto problematiku. Existují tři důležité pohledy na věc, kterými se zabývám.

### 2.3.1 Běžný postup

Spousta informací včetně cíle a motivace při hledání je zachycena v podobě odpovědí na otázku z anonymního dotazníku zveřejněného v článku [11]. Z odpovědí lze vidět, že většina dotazujících má detektoristiku především jako doprovodnou zálibu při venkovním relaxování



Obrázek 2.1: Kvalitní moderní detektor s citlivou cívkou a s vysokou přesností vyhodnocování signálu ovládací jednotkou.

s možností nalézt něco zajímavého. Pouze malé procento dotazovaných považuje nálezy za možnost vlastního obohacení (sbírka/prodej). Většina dotazovaných má s nálezy dobrý úmysl a o nález a postupu při jeho zacházení chce komunikovat s archeology. Běžně jde o zaznačení a pozdější předání nálezů nebo přivolání na místo nálezů. Z dotazníku vyplývá, že se jedná spíše o přátelskou spolupráci nežli nepřijemnou povinnost.

Nový hledač ale často neví, co by měl správně s nálezem dělat a nemá kontakt s archeologem. Z osobní zkušenosti vím, že myšlenka nález nahlásit nebo nějak dále řešit tímto směrem je přítomna až při hodnotnějším nález (zhodnoceným samotným hledačem). Pokud se amatér rozhodne nález správně zdokumentovat nebo spojit se s archeologem na místě vznikne problém. Zatímco stojí u místa nálezů (nejčastěji na poli, louce nebo v lese [11]) musí pomocí svého telefonu najít, do které oblasti spadá a s kým má komunikovat. Jedná se o zdoluhavé a pracné hledání, které nemusí skončit úspěšně.

### 2.3.2 Pohled archeologů

Dosavadní převažující formy přístupu k detektorovému hledání nepřinesly adekvátní řešení [4]. Absence vzájemné komunikace a nedostatek příležitostí k zapojení veřejnosti do odbornými pravidly regulované záchrany a objevování movitých archeologických památek dlouhodobě vedlo a vede jen ke ztrátě předmětů, které mají být součástí společně sdíleného kulturního bohatství.

Vlastní výzkumy a dlouhodobé zkušenosti partnerů jednoznačně ukazují, že podstatná část tzv. hledačské obce je tvořena zapálenými zájemci o historii a archeologii a představuje vysoce aktivní složku společenského zázemí oboru. Lze jim nabídnout dostatečný prostor pro smysluplné zapojení při objevování stop naší minulosti i v rámci platných zákonných norem ČR. Hledání není vždy jen individuální akcí, někdy je možné zavolat detektoristy i na detektorovou prospekci dané oblasti pod přímým dozorem.

Spolupráce s detektoráři je proto široce vítaná z řad aktivních archeologů [11]. Jejich nahlášené a předané nálezy jsou velkým přínosem, jelikož archeologická obec nemá kapacity na plnohodnotnou ochranu a záchranu všech archeologických památek a nálezů. Zajímavý nález může nastartovat průzkum okolí a tímto vést k získání důležitých informací [12]. A přitom jsou archeologické památky dennodenně ohrožovány, ať už intenzivní zemědělskou činností, různými stavebními aktivitami, nebo i jedinci, kteří je považují za volnou kořist pro obohacení soukromých sbírek či sebe samotných.

### 2.3.3 Ústavní pohled

Pro úvod do ústavního pohledu přejímám stanovisko ministerstva kultury České republiky [17].

„Ochranu archeologického kulturního dědictví na území České republiky upravlí Úmluva o ochraně archeologického dědictví Evropy (revidovaná; Sbírka mezinárodních smluv č. 99/2000) a zákon č. 20/1987 Sb., o státní památkové péči, ve znění pozdějších předpisů (dále jen zákon) v § 21 – 24. Úmluva o ochraně archeologického dědictví Evropy chápe archeologické dědictví jako zdroj evropské kolektivní paměti a jako nástroj historického a vědeckého studia. Za tímto účelem za archeologické dědictví považuje veškeré pozůstatky a objekty a jakékoli jiné stopy po lidstvu z minulých období:

- jejichž uchování a studium umožňuje vysledovat vývoj historie lidstva a jeho vztah k přirozenému prostředí
- o nichž jsou hlavními zdroji informací vykopávky nebo objevy a další metody výzkumu lidstva
- které jsou situovány na jakémkoli místě, které spadá pod jurisdikci smluvních stran

Do archeologického dědictví se zahrnují stavby, konstrukce, skupiny budov, zastavěná území, movité objekty, památky dalšího druhu a také jejich související prostředí nacházející se jak na souši, tak pod vodou.“

### 2.3.4 Shrnutí

Z těchto pohledů vyplývá, že je pro detektoráře nejtěžší kontaktovat archeology a zároveň navést amatérské hledače ke správnému postupu už na začátku. Nejen čerstvému hledači tedy chybí moderní nástroj pro usnadnění procesu získávání těchto informací a především komunikaci. Moje řešení je umožnit jednoduchou možnost přenosu informací archeologům a nabídnout hledačům funkce, které jim pomohou hlavně se záznamem nálezu.

## Kapitola 3

# Analýza problému

Kapitola se zabývá analýzou požadavků na řešení problematiky z minulé kapitoly a existujícími řešeními, které alespoň částečně splňují požadavky nebo jsou pomůckou při detektorové prospekci.

### 3.1 Požadavky

Znalost požadavků je velice důležitá pro návrh a tedy samotný vývoj řešení. Poskytují jistou představu o funkčnosti, vzhledu a použitelnosti. V mém případě bylo nutné získat požadavky na řešení od obou cílových skupin.

#### 3.1.1 Detektoráři

Moje cílová skupina pro řešení jsou především detektoráři. Pro získání požadavků jsem se vydal cestou rozhovorů a anonymního dotazníku sdíleném především na webových stránkách<sup>1</sup> věnovaných detektoristice. Všichni dotazovaní (60) uvedli, že jsou vlastníci detektoru kovu.

Získal jsem tak přehled o tom, co budoucí uživatelé chtějí a očekávají. Můj návrh na implementaci řešení v podobě mobilní aplikace byl valně kladně přijat. Nejvíce vítanější platformou (91%) byl zvolen mobilní operační systém Android. Implementace pro systém iOS i webová aplikace měla méně než 20% popularity, a proto jsem toto řešení zavrhnul. Rozdělení dotazovaných přišlo u otázky, jestli někdy spolupracují s archeology (obrázek 3.1a). U výsledné aplikace, kterou by 50% určitě uvítalo, je možnost mít nález a informace o něm sdílet s archeologem kladně přijata (90%).

Další část se dotazuje na aktuální způsob zaznamenávání dat (obrázek 3.1b). Někteří dotazovaní (23%) si ukládají pozici nálezu do mapové aplikace a 20% stále používá pro záznam fyzickou mapu. Největší část dotazovaných (38%) si vůbec souřadnice nálezu nezaznamenává. Jaké funkce by tato aplikace měla mít jsem vypsál i s možností přidání vlastního návrhu. Nejvíce vítanými funkcemi (nad 60% hlasů) byly zvoleny možnost záznamů na mapě, sledování trasy pohybu, vyznačení prohledávané mapy a jiné mapové podklady.

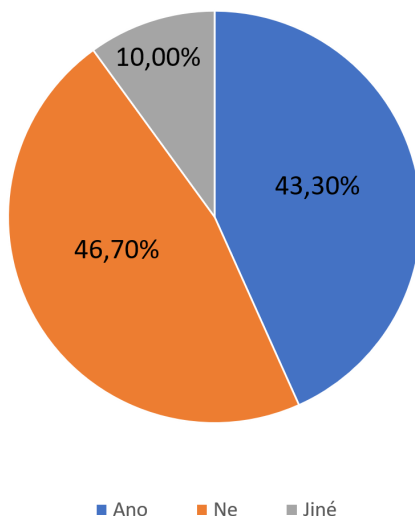
#### 3.1.2 Archeologové

S požadavky na řešení z pohledu archeologů mi pomohl pan Mgr. Balázs Komoróczy, Ph.D. pomocí e-mailové korespondence a osobního setkání. Jeho představa spočívala v tom, že

---

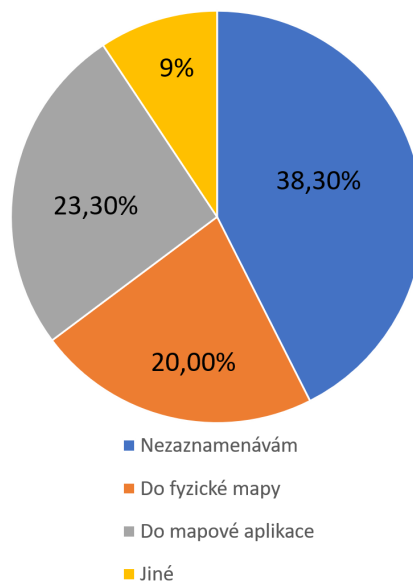
<sup>1</sup><https://www.lovecpokladu.cz/>

Spolupracujete někdy s archeology při nálezů?



(a)

Jakým způsobem si zaznamenáváte místo nálezů?



(b)

Obrázek 3.1: Obrázek (a) ukazuje rozdělení dotazovaných ohledně jejich spolupráce s archeology. Zbylých 10% převážně uvedlo, že by chtěli spolupracovat. Na obrázku (b) lze vidět, že velká část dotazovaných (40%) si nevede záznam o místě nálezů. Chybějící záznamy ústí ve ztrátu možných cenných dat. Mezi jinými odpověďmi se nachází hlavně způsob zápisu GPS souřadnic přímo u fotografie nálezů. Ať už se jedná o funkci telefonu nebo ruční zápis.

řešení pro detektoráře bude spojeno s webovou stránkou. Tato webová aplikace by obsahovala funkce pro práci s nálezů. Dovolovala by také zjistit, který detektorář dnes provádí detektorovou prospekci a zhruba kde. Tímto by se pokryla legálnost činnosti ze strany ústavy.

Důvodem k webové aplikaci, a tedy samostatného prvku od mobilní aplikace, je nepodstatnost mobilnosti řešení. Toto řešení by značně zjednodušilo náročnost komunikace s archeology při běžné činnosti detektorářů.

## 3.2 Existující řešení pro detektoráře

Tato část se věnuje řešení problematiky ukládání dat o nálezů, jejich sdílení a jiným nástrojům užitečných pro hledače. Samotné zaznamenávání přímo nevyžaduje specializovanou aplikaci, ale vyžaduje použitelnost daného řešení při hledání v terénu. Sekce se nevěnuje fyzickému tvoření záznamů jako je například záznam do tištěné mapy, který může být použit.

### 3.2.1 Mapové aplikace

K nahrazení fyzické mapy stačí mapa na mobilním zařízení schopná záznamu na dané pozici. Výhodou může být vestavěná lokalizace pomocí GPS pro přesnější zaznamenání pozice nálezu. Některá řešení nabízí i různé užitečné podklady jako satelitní snímky nebo historické mapy.

Právě na historických mapách (většinou lokálních) bývají zaznačené cesty, které se později nepoužívaly a zanikly. Tyto především polní a lesní cesty mohou skrývat nálezy mnohem starší, než lze nalézt na stále využívaných cestách. Zaměřil jsem se především na známé aplikace a na ty, které byly uvedeny dotazovanými (v části 3.1.1).

#### Google Maps

Google Maps je řešení mapového zobrazení od firmy Google, které je dnes využíváno nejen pro navigaci nebo komerci. Širokému rozšíření mobilní verze velkým dílem přispívá fakt, že se jedná o jednu z předinstalovaných aplikací v zařízeních se systémem Android. Mapy nabízejí nejrozličnější nástroje a jednoduché rozhraní.

Pro účely záznamu nálezů je nejvhodnějším řešením možnost vytvořit si vlastní mapu (Google My Maps<sup>2</sup>), kde na svoje podkladové vrstvy můžeme přidávat značky. Těmito značkám lze přiřadit pozici, název, popis a dokonce i obrázek. Výhodou je možnost mapu sdílet s ostatními nebo si ji zálohovat na úložiště. Mapy mají problém bez internetového připojení. Relativně rychlé propadávání licence a měnící se informace o místech (včetně podniků) na uživatelem určené části mapy nutí často aktualizovat nemalý počet dat, pokud chtějí mapu i nadále používat v offline režimu. Správa záznamů na vlastní mapě bez připojení k internetu také není praktická. Mapa se totiž zaměřuje i na GPS navigaci, zobrazení podniků a jejich hodnocení nebo vyhledávání spojů mezi danými zastávkami.

Toto řešení (Google My Maps) jsem krátce používal při záznamu nálezů, kvůli možnosti mapu sdílet s dalšími uživateli. Použití aplikace nebylo jednoduché a svádělo k dodatečnému záznamu nálezů na nemobilní webové verzi. Mobilní aplikace je zřídka kdy aktualizovaná, díky čemuž obsahuje dost chyb a není pohodlně použitelná, jak dokazuje spousta nových záporných recenzí na Google Play [9].

#### Mapy.cz

Mapové rozhraní<sup>3</sup> od české firmy Seznam nabízí často detailnější a přesnější zobrazení České republiky nežli Google Maps. Řešení rovněž nabízí lokalizaci pomocí GPS, vlastní mapu a záznam bodů. Možnosti u těchto bodů jsou značně omezené oproti řešení od Google. Nedá se připojit obrázek nebo přidat vlastní informace.

Výhodou je sbírka leteckých map a také užitečný historický podklad oblasti České republiky z 19. století. Ten je poskládaný ze spousty lokálních map, což velice usnadňuje práci, protože nenutí uživatele na každou malou oblast stahovat další podklady. Kvůli malé podpoře pro vlastní úpravu záznamů a absenci sdílení nejsou Mapy.cz dostatečným řešením pro častý záznam nálezů.

---

<sup>2</sup><https://www.google.com/maps/d/u/0/>

<sup>3</sup><https://mapy.cz/>

## Openstreet Maps

Openstreet Maps<sup>4</sup> je známé díky své webové aplikaci pro zobrazování a poskytování volně licencovaných map, vhodných pro další použití. Hlavním cílem projektu je poskytovat geografická data. Místní data (o podnicích, zastávkách apod.) zobrazená na mapách jsou tvořena uživateli. Běžně se aplikace využívá k zobrazení lokace s bodem ukazujícím na sídlo firmy na firemních webových stránkách. Ve webové aplikaci není možnost vytvoření vlastní mapy pro účely záznamu nálezů, bez využití programování.

## Old Maps

Aplikace<sup>5</sup> vznikla pro záchranu starých map s využitím OpenStreetMap API. Nabízí pro danou oblast možnost zobrazit veškeré historické podklady, které má k dispozici (uživatel je nemusí ručně hledat v seznamu). Těchto podkladů obsahuje přes 400 000, které jsou správně promítnuté na osy. Uživatel tedy může s pomocí GPS zobrazit svoji pozici na historické mapě.

Oproti předcházejícím řešením je práce s mapou pracnější a neumožňuje záznam vlastních nálezů. Jedná se tedy pouze o pomocnou aplikaci s velkým potenciálem ohledně historických podkladů. Má webovou, androidovou i verzi pro IOS. Androidová aplikace je déle než jeden rok neaktualizována a podle nedávných recenzí na Google Play ztrácí na funkčnosti i stabilitě. Aplikace je zdarma s občasným zobrazováním reklam.

## Vetus Maps

Aplikace<sup>6</sup> pro práci s mapou, která obsahuje historické podklady. Používá API Google Maps (popsané v části 3.2.1) a používá některé její vestavěné funkce. Uživatel má k dispozici nástroje pro lokalizaci, značení cesty a záznam bodů.

Navigace mezi jednotlivými mapami není jednoduchá z důvodu názvů v ruském jazyce. Samotné podklady je možné pouze vyzkoušet a později je nutné za ně zaplatit. Celkově aplikace působí nedodělaným dojem a některé funkce pro malování na mapu jsou nepraktické. Podle nedávných recenzí má aplikace chyby, které znemožňují její použití.

## Avenza Maps

Avenza Maps<sup>7</sup> si zakládá si na svém obchodě, kde si uživatel vybere mapu a má možnost stáhnout ji do svého zařízení pro zobrazení i bez připojení k internetu. Většina map je placená. Řešení nabízí široké možnosti vytváření a úprav vrstev. Do těchto vrstev lze přidávat vlastní body i s fotografiemi. Všechna uživatelská nastavení, zakoupené mapy i vytvořené vrstvy jsou synchronizovány a zálohovány s účtem Avenza. Aplikace je dostupná pro Android i pro iOS.

Uživatelé jako největší nedostatek uvádějí malý výběr map pro danou oblast, jejich kvalitu a vysokou cenu. Proto je pro účely záznamu nálezů a jejich následný export aplikace nevyhovujícím řešením.

---

<sup>4</sup><https://www.openstreetmap.org/>

<sup>5</sup><https://www.oldmapsonline.org/>

<sup>6</sup><https://www.vetusmaps.com/>

<sup>7</sup><https://play.google.com/store/apps/details?id=com.Avenza>



## Tect O Trak

Aplikace<sup>8</sup> se přímo specializuje na hledače s detektorem kovu. Uživatel si označí oblast, kde chce provádět hledání, a aplikace ho v této zóně následně sleduje a ukazuje, kterou část už prošel (3.2a). Nabízí možnosti záznamu nálezů a všechna data si ukládá do interní SQLite databáze. Nemá možnost zobrazení jiného podkladu, pouze satelitního od firmy Google. Aplikace je zdarma jen ve zkušební verzi (Tect O Trak Lite) s omezením na dvě zóny. Používá Google Maps API se systémem ukládání již dříve načtených částí mapy.

Aplikace je přes svoji omezenou funkčnost užitečná pro detektoristy díky možnosti záznamu a sledování pohybu. Nesplňuje požadavek řešení na možnost jednoduchého předávání informací o nálezech.



Obrázek 3.2: Obrázek (a) zobrazující zaznamenané trasy v aplikaci s přidáním záznamem nálezů. Každá část trasy se může zobrazovat pomocí jiné barvy. Výsledné zobrazení zaznamenaných nálezů po dokončení hledání (obrázek b). Aplikace umožňuje pouze tento pohled z úhlu na 2D mapu, vytvářející dojem 3D mapy.

### 3.2.2 Pomocné aplikace

Existují užitečné aplikace na Android, které hledačům neslouží pro záznam nálezů, ale jako pomůcky při hledání. Níže popisují dvě nejzajímavější.

#### Coinoscope

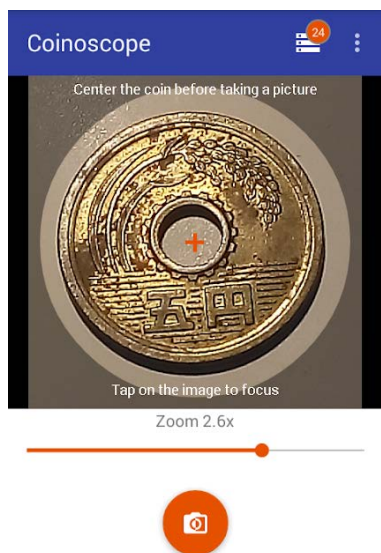
Účelem aplikace je identifikovat mince za pomoci fotografie, kterou porovná se svojí databází a uživateli nabídne svůj výsledek v podobě několika nejpodobnějších nalezených mincí. Úspěšnost určení záleží na raritě a kvalitě vyfotografované mince.

<sup>8</sup><https://play.google.com/store/apps/details?id=com.trib.app.tectotrak&hl=cs>



Identifikace mince umožní zjistit složení materiálu, které vede ke správnému zacházení v případě pokusů o očištění. Aplikace při úspěšné identifikaci nabízí údaje o minci, včetně původu nebo nákladu ražby.

Zajímavostí je, že tato funkce (galerie a rozpoznávání mincí), byla uvedena jedním dotazovaným jako možná funkce obsažená v mém řešení. Vlastní implementaci jsem zavrhl, jelikož Coinoscope funguje velice dobře a tak nová realizace není potřeba. Fotografie přiložené k nálezů v mém řešení bude možno využít i v této aplikaci, jelikož budou obsaženy ve standardní galerii.



(a)



(b)

Obrázek 3.3: Minci stačí vyfotit prostřednictvím aplikace (vlevo). A aplikace ji porovná se svojí databází a uživateli nabídne svoje nejlepší odhady na identifikaci mince (vpravo).

## Metal Detector

Aplikace měří sílu magnetického pole za pomoci magnetického senzoru, který není základní výbavou všech mobilních zařízení. Běžné použití zahrnuje hledání kabelů ve zdi nebo lokalizaci zapadlých věcí na nedostupných místech.

Nejedná se o úplnou náhradu detektoru kovu. Aplikace může být použita jako pomůcka pro dohledání nálezů pokud má detektor příliš velkou oblast měření a objekt obsahuje kov, který ovlivní přírodní úroveň magnetického pole.

### 3.2.3 Shrnutí

Výše zmíněné nástroje poskytují hledačům především jednotlivé funkce. Některé mapové aplikace jsou dobře zpracované a umožňují správu vlastních záznamů. Požadavek na uchování dalších informací o nálezů (název, fotografie, GPS souřadnice, ...) některé splňují, avšak možnosti tyto údaje jednoduše sdílet nejsou přítomny nebo jsou nedostatečné. Z analýzy existujících řešení se hodlám inspirovat převážně intuitivními uživatelskými rozhraními a některými unikátními funkcemi.

Pro splnění analyzovaných detektorářských požadavků (popsané v podsekci 3.1.1) je zapotřebí vytvořit vlastní mobilní aplikaci zahrnující záznam nálezů s možností sdílení archeologům. Sdílená data je pro splnění archeologických požadavků (podsekce 3.1.2) vhodné zpracovat a následně zobrazovat ve vlastní oddělené části řešení, ideálně webové aplikaci. Architektura propojení těchto částí je tedy klíčovým prvkem funkčnosti řešení a hlavním bodem návrhu.

## Kapitola 4

# Vývoj mobilní aplikace pro Android

Tato kapitola pojednává o operačním systému Android, jeho verzích a možnostech, prostudovaných vývojových prostředích, programovacích jazycích a nástrojích od společnosti Google pro tento operační systém. Zaměřuje se také na průzkum možností mapových API, pro využití ve vlastní implementaci.

### 4.1 Operační systém Android

Operační systém Android je dnes nejrozšířenějším operačním systémem především pro mobilní platformu [21]. Jedná se o open-source systém běžící na linuxovém jádře a jeho vývoj spadá pod giganta mezi společnostmi, firmu Google a organizaci Open Handset Alliance<sup>1</sup>. Tato organizace, která je konsorcium firem výrobců telefonů a společností, zaměřující se také na mobilní technologie (např. Nvidia, Intel, Sony, ...), se snaží o urychlený vývoj a propagaci systému Android, jakožto sjednoceného systému pro všechna mobilní zařízení.

Systém je silně multiplatformní a používá se nejen na mobilních telefonech různých výrobců, ale v upravených verzích i na jiných zařízeních jako je např. TV, tablet nebo chytré hodinky. Široká škála podporovaných typů telefonů si avšak vybírá svou daň při specializaci verzí samotného systému. Ten se totiž nemůže optimalizovat na nespočetné jednotlivé hardwarové konfigurace. Toto je částečně řešeno často využívanou možností výrobců telefonů si danou verzi nebo aktualizaci upravit a svým uživatelům ji modifikovanou poskytnout. Vzniká tím problém s dobou poskytnutí vylepšení koncovému uživateli, které se může projevit osudným například při kritických bezpečnostních aktualizacích systému.

### 4.2 Verze Androidu

Jednotlivé verze systému Android jsou označovány pomocí tzv. API a pro uživatele známějšími názvy (většinou) zákusků a číslem verze. Pro vývojáře je důležité, si před vývojem aplikace zvolit verzi API, které mu určí kompatibilitu pouze pro vyšší verze API a poskytne vývojářské nástroje pro tuto verzi. Aktuálně (k datu 10. 4. 2020) je nejnovější verzí Android Q (10.0) s API 29 [2]. Většina nyní používaných zařízení novou verzí nedisponuje, proto je pro výběr verze nutný průzkum právě používaných verzí.

<sup>1</sup><https://www.openhandsetalliance.com/index.html>

Nejpoužívanější verzí je podle zdroje [20] verze 9.0 Pie. Zdroj přejímá data z oficiálních zdrojů, které jsou dostupné pouze v aplikaci Android Studio (pro vývojáře). Při výběru minimální verze 6.0 (API 23) bude výsledná aplikace podle statistik aktuálně kompatibilní nejméně s 84% zařízeními.

## 4.3 Jazyk pro vývoj

Vývoj není omezen pouze na jeden jazyk. Nejčastěji použíty jazyky jsou Java, Kotlin a C++, které vývojové prostředí Android Studio přímo podporuje. Do aplikace lze avšak propojit i další jazyky např. JavaScript.

### 4.3.1 Java

Jazyk Java je velmi rozšířený a multiplatformní jazyk navržený pro WORA (*write once, run anywhere*), což znamená, že se aplikace nemusí znovu kompilovat při nasazení na jiné zařízení nebo platformu podporující jazyk Java. Jednalo se o doporučený jazyk firmou Google k vývoji aplikace pro systém Android nahrazený jazykem Kotlin [13]. Jazyk si užívá vysoké popularity i podpory a řešení na různé implementační problémy.

### 4.3.2 Kotlin

Jazyk Kotlin je staticky typovaný programovací jazyk určený pro zařízení podporující Javu. Kotlin přináší vylepšení oproti Javě, je zaměřen na lepší programovací zážitek a efektivitu. Jazyk cílí na sníženou potřebu programátora stále psát jednotlivé, standardizované struktury, které dělají kód více formálně srozumitelný, zato příliš dlouhý a nepřehledný.

Jde o podstatně novější jazyk, nežli jazyk Java a nabízí daleko modernější přístup k programování. Od května roku 2019 je Kotlin firmou Google preferovaným jazykem pro vývoj pro OS Android, čímž nahradil jazyk Java.

### 4.3.3 C++

Pro vývoj je možnost použít jazyk C++, přestože je označován jako nedoporučovaný, pokud se nejedná o hru využívající např. OpenGL či Vulkan nebo samotný přenos hry z jiné platformy. Jazyk nemůže využívat všechny funkce, které mu Android nabízí a je z části odkázan na rozhraní jazyků Java/Kotlin.

## 4.4 Aktivita

Aktivita (*Activity*) je jedním ze základních komponent aplikace. V Androidovém aplikačním modelu není funkce `main()`, která by zajišťovala start aplikace jako je tomu u programovacích paradigmat, nýbrž se s aplikací spustí předem definovaná aktivita.

Slouží k vykreslení okna s příslušnými prvky UI a následně zpracovává jejich interakci s uživatelem. Slouží rovněž k volání jiných aktivit buďto přímo pomocí specifikace ID dané aktivity nebo s voláním účelu (*Intent*). Například při dotazu „Zobraz jpg“ systém najde aktivity, které deklarují tento účel a při nálezů více aktivit dá uživateli možnost výběru aktivity (popřípadě aplikace), kterou chce pro vykonání činnosti použít.

Všechny aktivity mají svůj životní cyklus, který se mění na základě uživatelských interakcí s aplikací a je reprezentován pomocí metod. Ty jsou navrženy tak, aby poskytl

vývojáři možnosti pro ošetřování stavů, ve kterých se aktivita nachází. Jedná se například o ukládání údajů, interakci prvků nebo načítání rozhraní. Jednotlivé stavy na sebe navazují, jak ukazuje obrázek 4.1.

- **onCreate()** – jedná se o funkci, která je zavolána při vytvoření aktivity a je jediná povinná pro implementaci. Obsahuje základní startovní logiku, která by měla nastat pouze jednou v celém životním cyklu aktivity. Příkladem je načítání uživatelského rozhraní včetně přidávání funkcí a posluchačů k jeho interaktivním prvkům.
- **onStart()** – funkce je zavolána, když je aktivita zobrazena uživateli a byla vytvořena.
- **onResume()** – zavolána, jakmile uživatel začne interagovat s aktivitou.
- **onPause()** – jedná se o stav, kdy je aktivita stále alespoň částečně zobrazena.
- **onStop()** – aktivita už není zobrazena uživateli.
- **onRestart()** – zavolána v případě, že nezobrazená existující aktivita se má znovu zobrazit.
- **onDestroy()** – nezobrazená aktivita už není potřeba a bude systémem zničena.

## 4.5 Skladování dat

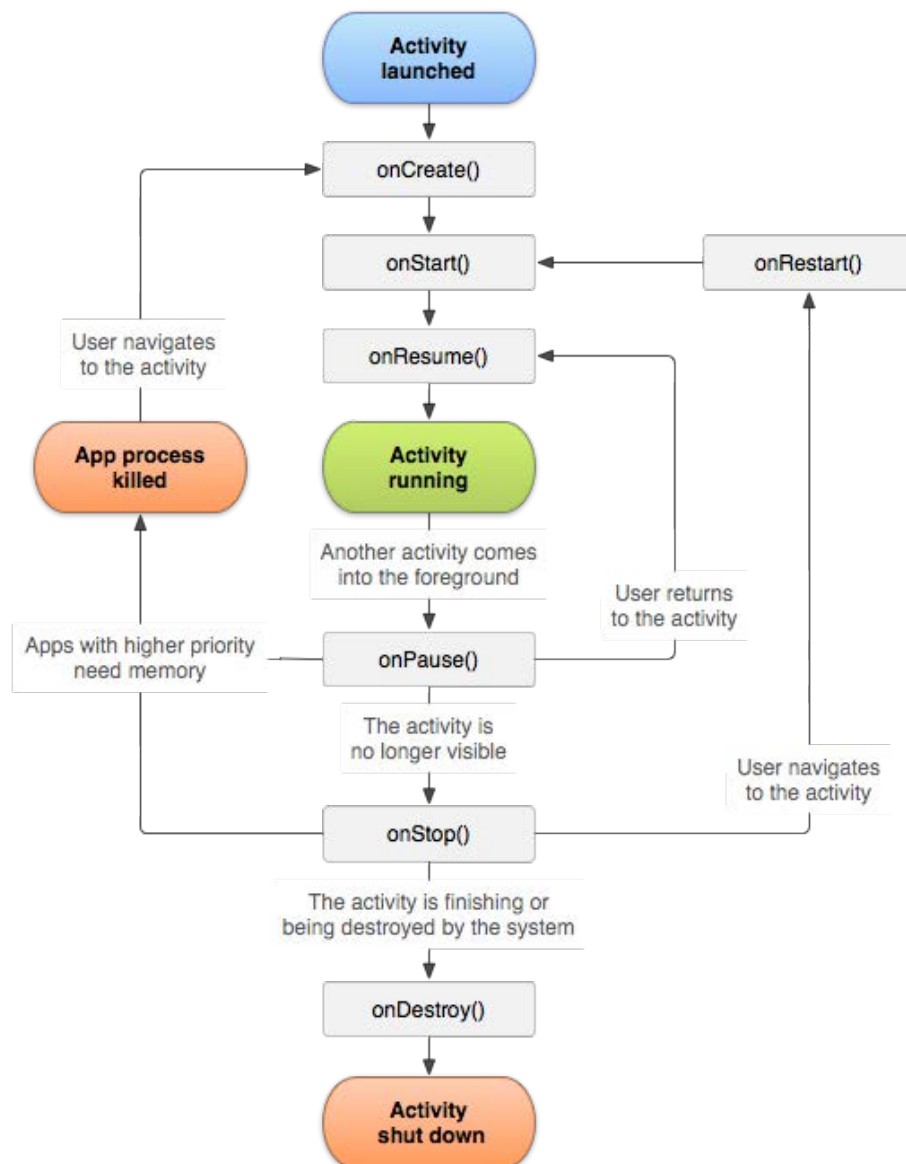
Android se neliší ve způsobu organizace dat od ostatních platforem. Vývojáři poskytuje možnosti<sup>2</sup> pro ukládání dat z aplikace, které mají různé možnosti sdílení, zálohování a souborová omezení.

- **Shared Preferences** – základní primitivní uložení hodnot, ve kterém má unikátní klíč právě jednu hodnotu. Data jsou uložena lokálně a lze k nim umožnit přístup i ostatním aplikacím. Určeno jen pro malý objem dat.
- **Databáze** – pro více dat, než je doporučeno pro Shared Preferences, slouží lokální databáze Room. Room poskytuje abstraktní vrstvu nad běžnou databází SQLite beze ztráty funkčnosti. Použití čisté databáze SQLite je rovněž umožněno.
- **App-specific files** – úložiště je privátní a nabízí možnost pro uchování dočasných i perzistentních souborů. Od verze Android 7 je umožněno data zašifrovat<sup>3</sup> (od verze Android 10 je zašifrování povinné). Z tohoto důvodu je toto úložiště doporučováno i pro skladování citlivých dat.
- **Shared storage** – určená pro uložení dat za účelem použití i pro další aplikace. Tato data se nemažou s odinstalováním aplikace a jsou běžně využívána pro dokumenty nebo mediální obsah.
- **Vlastní** – data je rovněž umožněno spravovat vlastním způsobem, například externě nástrojem Firebase.

---

<sup>2</sup><https://developer.android.com/training/data-storage>

<sup>3</sup><https://source.android.com/security/encryption/file-based>



Obrázek 4.1: Na schéma (ze zdroje<sup>4</sup>) lze vidět návaznost všech metod implementujících stavy života aktivity a důležité životní stavy aplikace (barevně). Díky znalosti schématu lze bezpečně spravovat aktivitu na základě uživatelských interakcí.

## 4.6 Nástroje

Pro androidové vývojáře byly vytvořeny nástroje pro vývoj, testování i distribuci aplikace. Následující zmíněné nástroje jsou velice podstatné a razantně zjednodušily fáze vývoje softwaru.

<sup>4</sup><https://developer.android.com/guide/components/activities/activity-lifecycle>

### 4.6.1 Google Play

Jde o službu od firmy Google pro distribuci aplikací, knížek, filmů a dalších produktů včetně jejich aktualizací pro operační systém Android. Už několik verzí je nedílnou součástí Androidu a nabízí pro vývojáře spoustu nástrojů ke správě aplikací, včetně statistik např. o počtu stažení nebo uživatelská hodnocení programu.

Ze strany uživatelů se jedná o jednoduchý distribuční nástroj, který nabízí kompatibilní a zabezpečené aplikace pro použité zařízení. Také si ukládá některá data spojená s jejich Google účtem, a tak uživatel snadno najde dříve odinstalovanou aplikaci nebo svůj uložený postup ve hře na jiném zařízení, kde se přihlásí.

### 4.6.2 Firebase

Jedná se o platformu pro mobilní a webové aplikace obsahující produkty, které pomáhají s vývojem, testováním a monitorováním těchto aplikací. Umožňuje vývojářům soustředit se na svoji aplikaci, zatímco jim nabízí převážně serverové funkce, které stačí jen použít a nemusí být vývojářem napsány a testovány.

Všechny tyto produkty jsou přehledně zdokumentované<sup>5</sup> a jednoduše použitelné. Firebase nabízí i chytré funkce pro správu publikovaných aplikací.

### 4.6.3 Android Studio

Google ve spolupráci se společností JetBrains vytvořil oficiální IDE pro vývoj Androidových aplikací s názvem Android Studio. IDE obsahuje množství prvků šetřící čas při vývoji. Základní typy aktivit jsou dostupné jako šablony. Umožňuje vyvíjenou aplikaci testovat na virtuálních mobilních zařízeních s vývojářem určenou verzí API, velikostí obrazovky nebo modelu. Uktovování a určování pozice jednotlivých prvků GUI lze jednoduše provádět v grafickém rozhraní, které generuje patřičné XML.

#### Gradle

Jedná se o nástroj pro automatizaci sestavování programu zaměřený na výkon a flexibilitu. Při překladu aplikace chytře používá své předchozí výstupy a znovu generuje jen ty, kterým se změnil vstup. Je nedílnou součástí Android Studia a oficiálním sestavovacím nástrojem pro Android s podporou všech jazyků, určených pro vývoj na tuto platformu.

## 4.7 Mapové API

Téměř všechny uživatelské funkce aplikace slouží pro práci s mapou. Implementace mapového zobrazení je možná i pomocí použití existujících API mapových knihoven, která vývoj značně zjednoduší. Výběr API ale zároveň ovlivní programovací jazyk pro vývoj i používání vestavěných funkcí a může mít omezení ohledně jeho použití. Je proto důležité věnovat čas výběru poskytovatele a určit si nejvýhodnější řešení pro vlastní účely implementace.

---

<sup>5</sup><https://firebase.google.com/docs>

### 4.7.1 Google Maps API

Google Maps má přímo v programu Android Studio šablonu, která umožňuje jejich jednoduché a rychlé použití. Obsahuje sadu funkcí, které jsou dobře popsány v dokumentaci<sup>6</sup>, a proto jejich implementace není náročná. S knihovnou se pracuje pomocí jazyka Java. Využívání této knihovny je pro aplikace vytvořené na systém Android zdarma.

Vytvoření mapy s možností tvoření a správy záznamů, včetně informací a fotografií, je celkem snadné. Uživatelské rozhraní je intuitivní a pro většinu mobilních uživatelů známé. Nejedná se ovšem o volný software a tak tvoření vlastních funkcí není jednoduché. Předávání informací o vybraných záznamech by muselo fungovat na principu sdílení celé mapy s dalším uživatelem nebo ručního exportování. Obě varianty jsou velice nepraktické a při správě více detektorů jedním archeologem neefektivní.

### 4.7.2 Leaflet

Leaflet je moderní knihovna napsaná v jazyce JavaScript pro tvorbu interaktivních map. Jedná se o open-source projekt a obsahuje spoustu funkcí a pluginů pro vývojáře. Knihovna je zaměřena na jednoduchost, výkon a použitelnost. Je vhodná pro webové i mobilní nasazení. Má velice dobře zpracovanou dokumentaci<sup>7</sup>, díky které lze vytvořit funkční mapové zobrazení se základními funkcemi v krátkém čase. Nesnaží se přizpůsobit požadovaným datům a ta zobrazit, ale vyžaduje data zpracovaná v podporovaných formátech a ta je pak schopná velice dobře znázornit. Knihovna nabízí i zapojení se do vývoje pomocí přímého zásahu do kódu nebo pomocí nahlašování chyb či vylepšování dokumentace.

Využití knihovny přináší nevýhodu v podobě nutnosti implementace funkcí spojené přímo s mapou pomocí jazyka JavaScript. Tyto funkce budou muset úzce spolupracovat s rozhraním pro Android napsaném v jazyce Java nebo Kotlin, což nemusí být lehké vyřešit.

### 4.7.3 OpenLayers

OpenLayers je open-source knihovna pro zobrazování mapových dat ve webových prohlížečích. Je napsaná v jazyce JavaScript. Oproti knihovně Leaflet je komplexní, schopná zobrazovat velké množství formátů dat a poskytuje spousty vestavěných funkcí bez nutnosti importování dalších knihoven. Zprovoznění základních funkcí zde trvá déle a část dokumentace<sup>8</sup> je zastaralá. Nevýhodou je také nižší podpora ze strany vývojářů a celkově komunity. Knihovna je zdarma a sdílená pod FreeBSD licencí.

---

<sup>6</sup><https://developers.google.com/maps/documentation/android-sdk/intro>

<sup>7</sup><https://leafletjs.com/reference-1.6.0.html>

<sup>8</sup><https://openlayers.org/en/latest/doc/>



## Kapitola 5

# Návrh

Návrh je pro vývoj mého řešení velmi důležitý a těžší z hotového výzkumu a analýz z předchozích kapitol. Zabývá se výběrem a použitím vhodných technologií, zpracovává uživatelské rozhraní a řeší celkové fungování jednotlivých komponent.

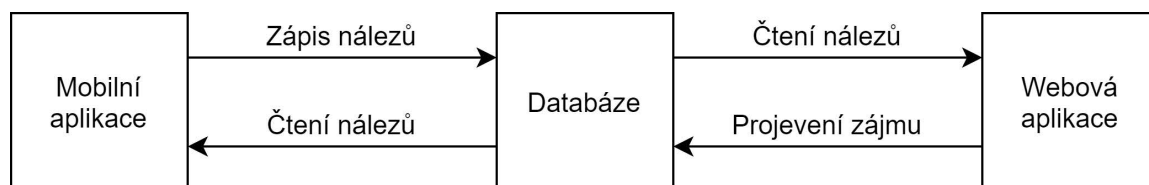
Mé řešení je podle požadavků rozděleno na dvě části a to mobilní a webovou aplikaci. Je ale nutností umožnit těmto aplikacím mezi sebou komunikovat. Kapitola se proto zabývá průzkumem možností pro implementaci obou aplikací a komunikací mezi nimi.

### 5.1 Architektura

Architektura, která mimo jiné specifikuje komunikaci mezi prvky architektury, spočívá v implementaci řešení klient-server nebo vytvoření prvku zprostředkovávajícího přenos dat. Jeden z požadavků na mobilní aplikaci je možnost nálezy upravovat, sdílet a mít je zálohované. Řešení, které se vhodně nabízí, je databáze. Ta pokryje požadavky na uložení dat a pokud bude externí i internetově dostupná, může být čtena i webovou aplikací. Toto řešení umožňuje vynechat implementaci přímé komunikace mezi aplikacemi.

#### 5.1.1 Model

Pro vizualizaci návrhu architektury jsem vytvořil jednoduchý model zobrazený na obrázku 5.1. Model ukazuje vzniklá nebezpečí této architektury spočívající v databázi. Výpadek tohoto prvku by znemožnil čtení a tedy sdílení dat. Proto je volba databáze a samotného hostingu velice důležitým předmětem návrhu.



Obrázek 5.1: Obrázek návrhu zjednodušeného modelu architektury. Webová aplikace nebude moci provádět správu celého záznamu o nálezu, ale jen jeden z atributů (projevení zájmu). Databázový systém řeší také problém souběžného čtení a zápisu obou aplikací.

## 5.2 Mobilní aplikace

Nejdůležitějším prvkem celého řešení je mobilní aplikace na Android určená pro detektoráře. Důvod výběru mobilní platformy a právě systému Android jsem již popsal v podsekcí 3.1.1.

Hlavní myšlenkou je implementace funkcí pro práci s mapovým zobrazením. Aplikace musí být schopná přidávat a spravovat záznamy uložené na daném bodě na mapě. Je potřeba zajistit připnutí dalších informací a dat (název, popis, fotografie) k těmto záznamům. Tato data (kromě fotografií) je vhodné ukládat do databáze (jak ukazuje model v podsekcí 5.1.1). Samotná aplikace ale nebude implementovat vlastní mapové zobrazení, ale použije existující mapové API (prozkoumané v sekci 4.7). Aplikace se tak stane prostředníkem mezi databází a vybraným API.

### 5.2.1 Požadavky

Požadavky na aplikaci vycházejí na základě osobních rozhovorů, vlastních zkušeností a převážně anonymního dotazníku (zpracovaného v sekci 3.1.1).

- *Možnost záznamu na mapě* – jedná se o stěžejní funkční prvek, umožňující uložení dat a případné jejich sdílení. Zaznamenaná pozice musí být co nejpřesnější, a proto je vhodné použít GPS, která je součástí téměř všech cílových zařízení s umožněním případné uživatelské ruční korektury.

K uloženým informacím patří i název, popis a případná fotografická dokumentace. Všechna data je potřeba ukládat v rozumném formátu pro jednoduší možnosti exportu. Proto je důležité prozkoumat možnosti využití databáze (sekce 5.3).

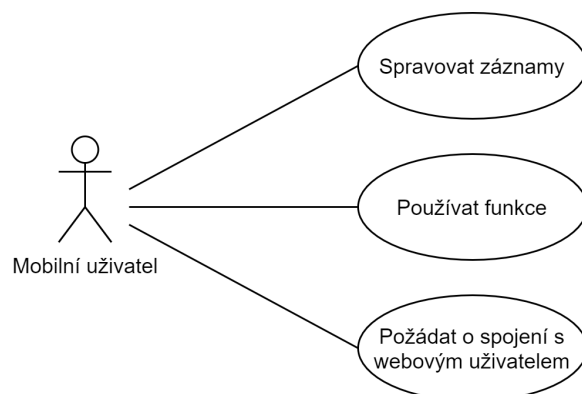
- *Sledování pohybu uživatele* – s pomocí GPS lokalizace je implementace sledování pohybu otázkou sběru a následného zobrazování dat.
- *Vyznačení prohledávané oblasti* – bude nutné implementovat kreslicí funkce nad mapovým zobrazením. Náročnost se odvíjí od možností použitého mapového API.
- *Možnost sdílet data s archeologem* – řešením je přímá komunikace s externí databází s uživatelským parametrem určujícím, jestli je nález sdílený.

### 5.2.2 Uživatelské rozhraní

Všechny funkce jsou schované za uživatelským rozhraním. Je to prvek, který uživatel vidí jako první a také po celou dobu používání aplikace. Sofistikovanost aplikace není ve výsledku důležitá, pokud nemá uživatel jednoduché nástroje pro její použití. Zaměřím se proto na vytvoření jednoduchého a intuitivního rozhraní podle pravidel z knihy [10].

#### Případy užití

Pro vytvoření návrhů uživatelského rozhraní je třeba si stanovit alespoň základní případy užití. Ty popisují chování systému za účelem popsat jeho funkcionalitu. Pro demonstraci této činnosti jsem vytvořil diagram případů užití (obrázek 5.2).

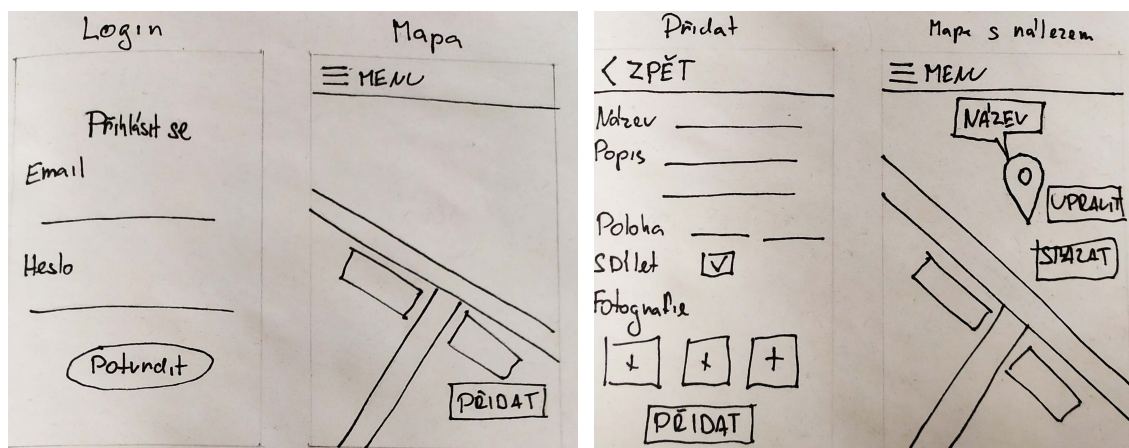


Obrázek 5.2: Do správy záznamu patří možnost záznam přidat, sdílet, upravit, smazat a zobrazit. Možnost sdílet se na záznamech projeví až je mobilní uživatel úspěšně spojen s webovým uživatelem (uživatel požádal o spojení a byl přijat).

## Skica

Skica (sketch) je prvním vizuálním krokem procesu vývoje uživatelského rozhraní. Je nejméně graficky zpracovaný, protože se nejčastěji jedná o ruční náčrty. Cílem je vytvořit vizuální reprezentaci své představy o grafickém rozvržení. Na základě tohoto náčrtu, který položí především základy distribuce prvků, se vytvoří drátěný model.

Svůj náčrt (obrázek 5.3) jsem vytvořil na základě své představy o rozvržení a dříve určených případů užití (podsekce 5.2.2). Ukazuje jednotlivé obrazovky, které považuji za nejdůležitější. Tento náčrt mi poskytl předlohu pro vytvoření další fáze vývoje grafického rozhraní – drátěný model.

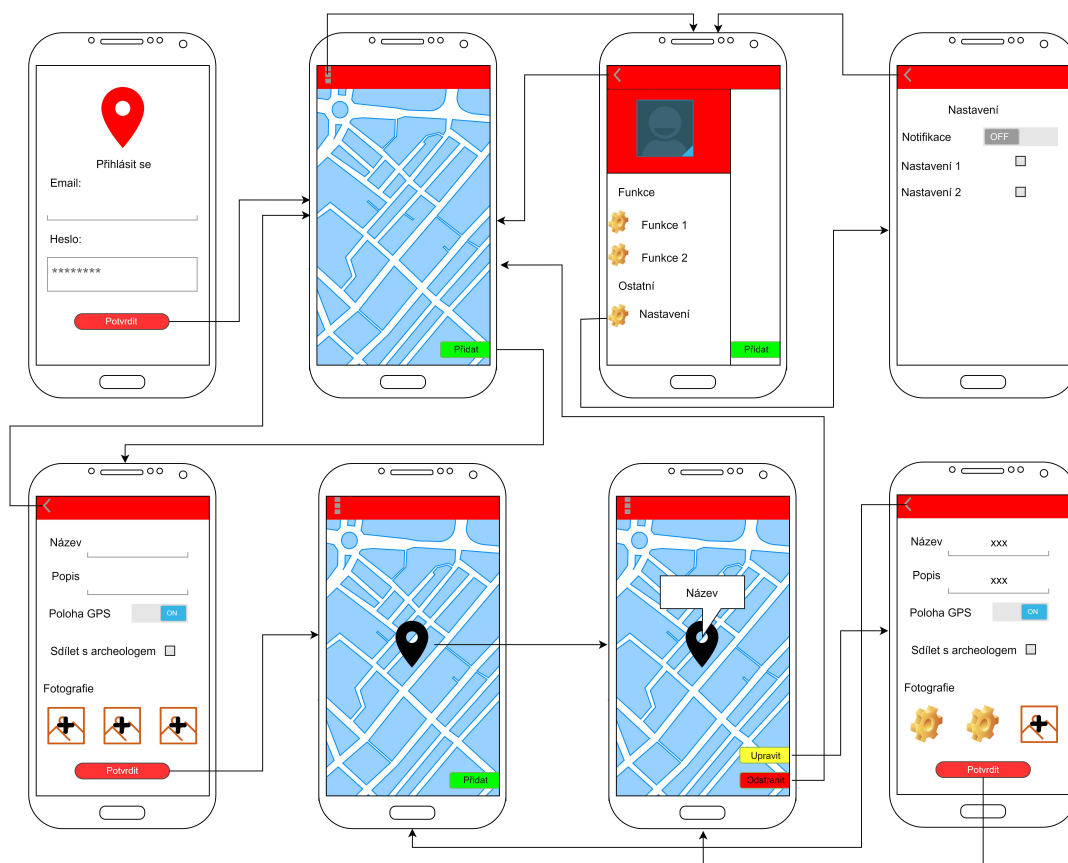


Obrázek 5.3: Náčrt ukazuje i proces základního použití aplikace. Po přihlášení (náčrt Login) se zobrazí mapové zobrazení aplikace s jednoduchým a intuitivním menu (náčrt Mapa). Pro přidání záznamu se zobrazí obrazovka pro vyplnění údajů (náčrt Přidat). Po úspěšném uložení nálezu se záznam zobrazí na mapě a uživatel bude mít možnost tento nález upravit (změna údajů) nebo smazat (náčrt Mapa s nálezem).

## Drátěný model

Drátěný model (*wireframe*) je obrys kostry aplikace. Poskytuje čistý přehled o struktuře jednotlivých stránek, rozhraní, funkčnosti, uživatelských akcí a zamýšleného chování. Není vyladěný pomocí všech konceptů nebo grafiky a míří na nízkou náročnost na výrobu, přesto je kritickým krokem procesu návrhu interakce. Drátěný model může být vytvořen rukou nebo digitálně, dle požadovaného důrazu na detaily.

Vytvořená kostra pro moji aplikaci je prezentována na obrázku 5.4. Kostra ukazuje, jak se funkčnost aplikace ubíhá hlavně okolo mapového zobrazení. K tomuto návrhu jsem přistoupil s větším důrazem na detaily. Z tohoto důvodu jsem nepokračoval v dalších běžných fázích vývoje grafického rozhraní (tzv. mockup a prototyp).



Obrázek 5.4: Kostra (wireframe) ukazuje základní rozvržení a interakci jednotlivých obrazovek aplikace. Většina funkcí je přímo svázána s mapovým zobrazením, které spolu s obrazovkou pro správu záznamů tvoří nejdůležitější prvky celé aplikace. Kostru jsem vytvořil za pomoci nástroje draw.io<sup>1</sup>.

## Shrnutí finálního návrhu UI

Popis konečného návrhu vychází především z vytvořené kostry (obrázek 5.4). První co uživatel uvidí bude přihlašovací obrazovka, která slouží pouze k přihlášení. Graficky je navržena velmi jednoduše, přestože vytváří část prvního dojmu. Běžně se k ní uživatel

<sup>1</sup><https://draw.io/>

znovu nedostane, pokud se neodhlásí. Aplikace si totiž bude pamatovat jeho přihlášení. Po přihlášení se zobrazí hlavní okno aplikace s dominantním mapovým zobrazením. Mapa bude interaktivní a bude mít běžné dotykové interakce pro posunutí, přiblížení a podobně. Na mapě budou zobrazeny všechny dříve uložené záznamy. Kostra dále ukazuje rozbalení postranního menu, obsahující možnosti odhlášení, použití mapových funkcí nebo nastavení. Tento typ menu je dnes běžný v mobilních aplikacích díky jeho intuitivnímu použití. Také uživatele nepřesměrovává do nové obrazovky, ale jen částečně překryje hlavní obrazovku, což přidává na jednoduchosti orientace v aplikaci.

Umístění tlačítka přidat (nebo později upravit a odstranit) je ideální právě vpravo dole, protože nebude překážet menu ani mapovým funkcím (zobrazeným vpravo nahoře). Rozvržení obrazovky pro přidání nálezu vyplývá z požadavků na uložená data. Uživatel nejdříve zadá vlastní text do kolonek název a popis a už mu jen stačí vybrat si způsob určení GPS souřadnic, sdílení a vybrat fotografii. Úspěšné uložení nálezu uživatele přesměruje opět na obrazovku s mapovým zobrazením s již promítnutým novým záznamem. Po označení nálezu (kliknutí na jeho ikonku) se uživateli místo tlačítka „Přidat“ zobrazí „Upravit“ a „Odstranit“. Volba upravit zobrazí stejnou obrazovku jako při přidávání nového nálezu, avšak s vyplněnými údaji. Tímto způsobem se uživatel také může podívat na všechna data, která si k nálezu uložil včetně fotografie.

## 5.3 Databáze

Komunikačním uzlem mezi aplikacemi je v mém případě databáze. Tento systém především poskytuje možnost uložení dat v elektronické podobě. Takto uložená data bude možné využít pro obě aplikace (jak ukazuje model v sekci 5.1). Druhů databází je více a jejich použitelnost se převážně odvíjí od požadavků. V mém případě se jedná o důraz na rychlost zobrazování vybraných dat, možnost zabezpečení přístupu a dostupnost z internetu. Dvě řešení pro danou specifikaci jsou popsány níže.

### 5.3.1 Vlastní server

Jednou z možností je provozovat relační databázi na vlastním serveru. Toto řešení umožňuje plnou kontrolu nad databází včetně tvorby zabezpečení, správy paměti nebo správy procesu ověření identity uživatele. Jelikož bude výsledná aplikace poskytována zdarma, je nutné najít řešení bez nákladů. Pro práci, správu a komunikaci s databází je ale potřeba použít dotazovací programovací jazyk například SQL. Dnešní nejpoužívanější databázové systémy si zakládají na jeho podpoře [6].

### 5.3.2 Existující služby

Správa vlastní databáze nemusí být ve všech případech vhodná. Pokud chce vývojář pouze jednoduchou databázi a neztrácet čas s nastavováním celého serveru, může využít už hotová řešení běžící na cizích serverech. Většina poskytovatelů těchto služeb má možnost používání zdarma do určitého limitu využití toku dat.

#### Firestore Realtime Database

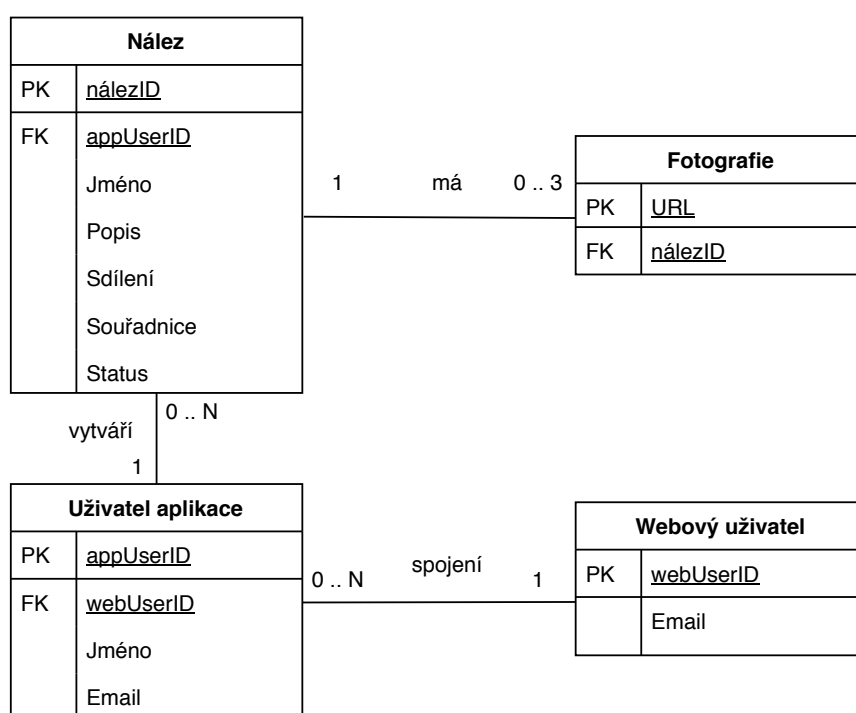
Pro Android aplikace jsou připraveny funkce z balíčku služeb Firestore, představeném dříve v podsekcí 4.6.2. Jedním z nabízených nástrojů je Firestore Realtime Database. Jedná se

o databázi běžící na serverech firmy Google. Veškeré funkce pro správu dat nebo přístupů jsou připraveny k použití a dobře zdokumentované<sup>2</sup>.

Data si databáze ukládá ve formátu JSON a jsou synchronizovaná v reálném čase s každým připojeným klientem. Tato funkce umožňuje sdílení informací mezi webovou a mobilní aplikací bez implementace dalších komunikačních prostředků. Každý připojený klient dostane upozornění na změnu v určené části databáze a už je na vývojáři, jak informaci zpracuje.

### 5.3.3 Uložená data

Výběr databáze se také odvíjí od uložených dat. V mém případě je nutné si uchovávat údaje o nálezech i mobilních a webových uživateli. Navržený způsob architektury uložených dat je na obrázku 5.5. Způsob ukládání dat pod složkou s unikátním ID pomůže při nastavování oprávnění přístupu.



Obrázek 5.5: Obrázek ukazuje tři hlavní tabulky, uchovávající navržená potřebná data. Atribut sdílení určuje, jestli je nález sdílený se spárovaným archeologem. Samotné nálezy se nacházejí pod tabulkou s unikátním ID uživatele, který je zaznamenal. Nález také může obsahovat odkazy až na tři fotografie. Informace o uživateli jsou v samostatné tabulce. Ta obsahuje základní údaje a ID webového uživatele (pokud je s nějakým spárovaný). Poslední tabulka uchovává všechny spojení daného webového uživatele spolu s jeho e-mailovou adresou.

<sup>2</sup><https://firebase.google.com/docs/database>

### 5.3.4 Autorizace

Zabezpečení databáze probíhá nejčastěji pomocí nastavení oprávnění. Přístup k databázi bude muset být povolen pouze pro přihlášené uživatele mobilní aplikace a autorizované webové uživatele. Přihlašovací údaje k mobilní aplikaci musí být spojené s unikátním ID, které umožní uživateli s tímto ID přístup pouze k vlastním nálezům. V případě nastavení sdílení na tomto nález bude nutné umožnit přístup i spárovanému webovému uživateli. Tato oprávnění se budou muset řešit pomocí databázových pravidel přístupů.

### 5.3.5 Úložiště

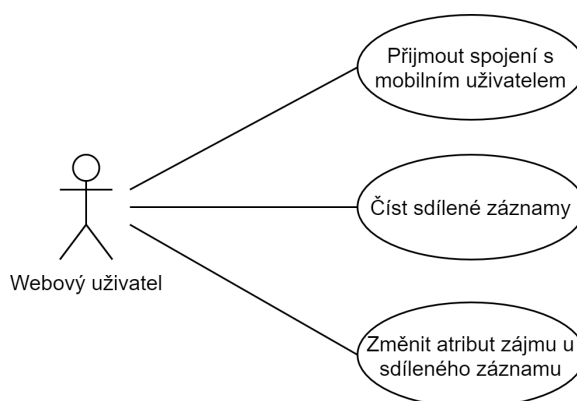
Fotografie se nehodí pro přímé vložení do databáze vzhledem k požadavku na vyšší kvalitu. K databázi tedy bude muset být připojeno i úložiště. Každá fotografie bude vázána na konkrétní nález v databázi. Mobilní uživatel bude moci tyto fotografie zobrazovat a mazat. Webový uživatel si bude moci fotografii zobrazit, pokud má oprávnění číst nález, ke kterému je daná fotografie vázaná.

## 5.4 Webová aplikace

Na základě požadavků od archeologů (popsané v podsekcí 3.1.2) jsem se rozhodl vytvořit i webovou aplikaci. Jedná se o rozšíření zadání mé bakalářské práce. Aplikace bude schopná pracovat s databází, protože archeolog bude moci číst nálezy s ním sdílené. Zároveň bude mít možnost upravit jednu hodnotu (status), která se projeví i v mobilní aplikaci. Tato změna upozorní detektoráře, že archeolog projevil nebo neprojevil zájem o daný nález. Když mobilní uživatel zažádá o spárování, archeolog bude mít možnost žádost přijmout a mít tak přístup k uživatelským sdíleným nálezům.

### 5.4.1 Uživatelské rozhraní

Webová aplikace musí zajistit funkčnost požadovaných funkcí, zobrazených na diagramu případů užití 5.6. Tyto funkce musí být uživateli dostupné skrze uživatelské rozhraní. Zvolil jsem stejný postup pro vytvoření návrhu jako u mobilní aplikace (podsekcí 5.2.2).

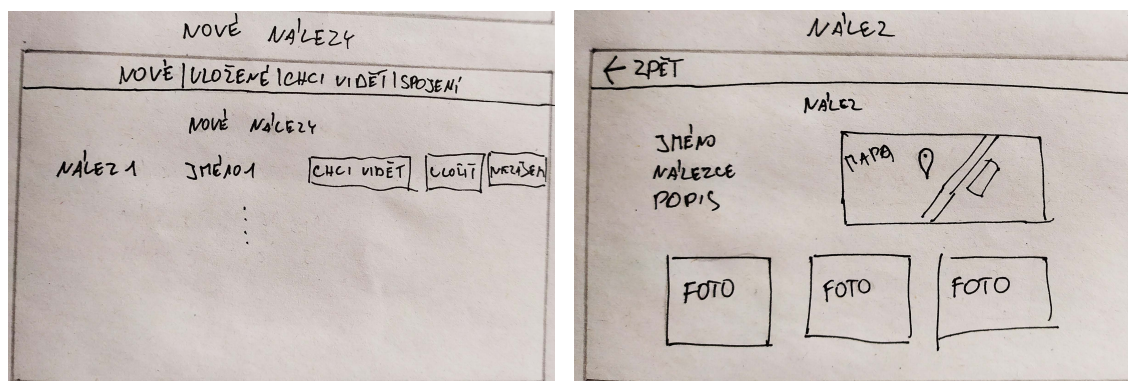


Obrázek 5.6: Sdílený záznam je ten, který je označen mobilním uživatelem jako sdílený a zároveň je webový uživatel s tímto uživatelem spárovaný.



## Skica

Vytvořil jsem náčrt (obrázek 5.7), který zobrazuje základní rozložení stránky. Jedná se o jednoduchý formát s horní lištou jako menu pro navigaci a zbytkem stránky pro obsah. Po vytvoření náčrtu jsem už měl jasnou představu, jak bude výsledek implementace vypadat. Z tohoto důvodu jsem nepokračoval v dalších krocích vývoje uživatelského rozhraní.



Obrázek 5.7: Náčrt ukazuje návrh dvou nejdůležitějších stránek. Stránka pro nové nálezy (vlevo) má jako obsah tabulku s nálezy. U každého nálezu je jméno autora, tlačítka pro reagování a název. Právě nález je interaktivní a po kliknutí je uživatel přesměrován na stránku obsahující další údaje o náleze (vpravo). Stránka obsahuje polohu místa nálezu v podobě bodu na mapě, zadané údaje a případně přiložené fotografie.



## Kapitola 6

# Implementace

V kapitole je rozebrána realizace navrženého systému dle vytvořeného návrhu. Popisuje jednotlivé zajímavější prvky implementace a nalezené problémy. Soustředí se na vysvětlení důležitých částí celého řešení. První část se zabývá hlavním prvkem řešení a to mobilní aplikací. Dále popisuje použitou databázi. Následuje část věnující se webové aplikaci. Na závěr popisuje použité nástroje a technologie.

### 6.1 Mobilní aplikace

Cílem implementace mobilní aplikace je vytvořit řešení na operační systém Android splňující uživatelské požadavky (popsané v sekci 3.1.1). Využívá prozkoumaných možností z kapitoly 4 a navrženého řešení (kapitola 5). Implementačně se aplikace dělí na dvě hlavní části. První část je aplikace pro Android a druhá zajišťuje práci s mapou v této aplikaci.

#### 6.1.1 Programovací jazyk

Pro implementaci mobilní aplikace jsem zvolil jazyk Java popsáný v části 4.3.1. Hlavním důvodem je fakt, že už mám zkušenosti s programováním v tomto jazyce přímo pro platformu Android. Přestože je jazyk Kotlin (4.3.2) doporučovaný firmou Google [13] a mnoha dalšími zdroji [14, 16, 19], jeho výběr jsem zavrhl z důvodu mé nedostatečné zkušenosti s ním.

#### 6.1.2 Mapy

Hlavní prvky aplikace spočívají v práci s mapou. Většina funkcí s ní přímo pracuje, a tak prvním krokem v realizaci aplikace byla implementace samotného mapového zobrazení. Realizovat čistě vlastní řešení nemá význam, protože už existují funkční a snadno použitelné mapové API. Ty jsem prozkoumal v rámci sekce 4.7.

Výběr knihovny spočíval v jednoduchosti použití, užitečnosti vestavěných funkcí (pro moje účely) a složitosti implementace vlastních funkcí. Z tohoto důvodu jsem se rozhodl pro použití knihovny Leaflet.

#### Leaflet

Knihovna (prozkoumána v podsekci 4.7.2) je určená i pro webové aplikace a je napsána v jazyce JavaScript. Implementace veškerých mapových funkcí proto proběhla mimo rozhraní

jazyka Java. Právě JavaScript není přímo podporován programem Android Studio (nástrojem pro tvorbu aplikací na Android). Z tohoto důvodu jsem použil i nástroj WebStorm. Nejdříve jsem implementoval základní mapové zobrazení odděleně od aplikace. Celý proces testování tohoto zobrazení probíhal v prohlížeči, jelikož je pro tuto platformu knihovna Leaflet určená. Server byl vytvořen za pomoci nástroje WebStorm, který službu lokálního serveru poskytuje.

Promítnutí tohoto zobrazení do aplikace jsem provedl použitím třídy `WebView` (zobrazeno v úryvku kódu 6.1). Tato třída umožňuje zobrazení webového obsahu a částečně dokáže nahradit funkci webového prohlížeče (avšak postrádá některé funkce). Použitím metody `loadUrl` se mi podařilo nahrát základní mapu (z tutoriálů). Toto načtení umožňuje volat funkce z mapy aplikací, ale mapa nemá možnost jakékoliv zpětné komunikace. Uskutečňuje pouze promítnutí webové realizace mapy do aplikace a uživatel má tedy možnost pracovat s mapou, aniž by od ní aplikace dostávala jakékoliv údaje. Bylo tedy nutné vytvořit rozhraní, ke kterému mají oba prvky přístup.

```
// Přiřazení grafického prvku mapovému zobrazení
WebView webView = findViewById(R.id.webView);
// Načtení zdrojových souborů
webView.loadUrl("file:///android_asset/www/index.html");
// Přiřazení rozhraní
webView.addJavascriptInterface(new WebAppInterface(this), "Android");
// Ukázka volání funkce mapového zobrazení pro odstranění záznamu
webView.loadUrl("javascript:GPSPinDelete()");

/**
 * Třída rozhraní
 */
public class WebAppInterface {

    // Funkce rozhraní pro nastavení klíče
    @JavascriptInterface
    public void setKey(String key) {
        uniqueKey = key;
    }
}
```

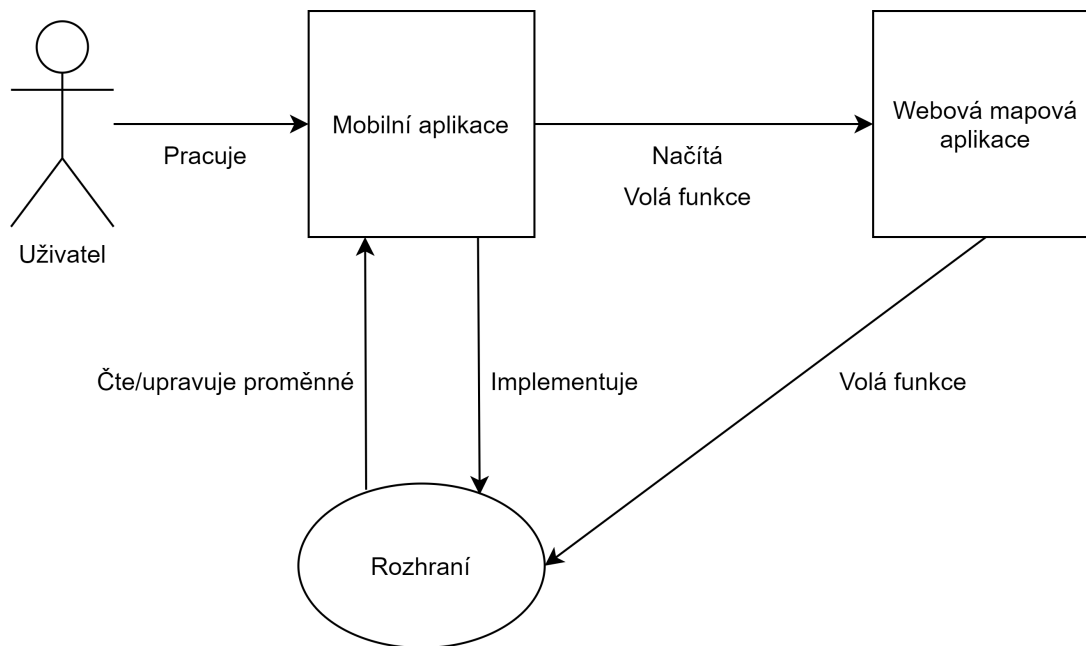
Výpis 6.1: Ukázka pasáže načtení mapového zobrazení s voláním funkce pro odstranění záznamu. Dále je ukázána třída rozhraní `WebAppInterface`, kterou ve fragmentu kódu 6.2 používáme pomocí volání jejích funkcí z načteného webového zobrazení.

```
// Funkce pro nastavení klíče zobrazeného nálezu do mobilní aplikace
function onMarkerClick(e) {
    Android.setKey(e.target.options.keyId);
}
```

Výpis 6.2: Fragment kódu v jazyce JavaScript ukazuje volání funkce z rozhraní pro nastavení ID v části psané v jazyce Java. Funkce `onMarkerClick(e)` je přiřazena jako funkčnost záznamu (markeru) po kliknutí. Tento záznam má v sobě uložené vygenerované ID, které funkce získá (pomocí atributu `e`, které představuje označený objekt) a posílá do Java části.

## Rozhraní

Pro tuto formu komunikace slouží moje implementovaná třída `WebAppInterface` na straně Android aplikace. Aby byl obsah této třídy dostupný mé instanci třídy `WebView`, bylo jí nutné toto rozhraní přidělit pomocí příkazu `addJavascriptInterface`. Ten umožňuje webovému prvku volat všechny funkce obsažené v tomto rozhraní, které mají anotaci *JavaScriptInterface*. Implementoval jsem proto funkce, které jsou běžně používané pro čtení a změny atributů objektů zvenčí (tzv. *getter* a *setter*) pro důležité proměnné. Toto řešení umožnilo volat funkce přímo pomocí jazyka JavaScript z mé mapové implementace a reagovat tak aplikaci na uživatelskou interakci s mapou (obrázek 6.1). Takto pracující rozhraní mi umožnilo plnou realizaci mapových funkcí.



Obrázek 6.1: Obrázek ukazuje důležitou funkci rozhraní. Například když uživatel na mapě klikne na nález, tak přes mobilní aplikaci načtená mapa pošle rozhraní ID nálezu a zavolá funkci pro změnu UI (změna tlačítek). Pokud uživatel zažádá o úpravu nálezu, aplikace uživateli zobrazí data z databáze o označeném nálezu díky zaslanému ID.

## Mapový základ

Vytvoření základního zobrazení je díky přehledným tutoriálům<sup>1</sup> knihovny Leaflet snadné. Stačí si vytvořit vlastní instanci třídy **Map**. Této instanci je poté možno přiřadit počáteční souřadnice a úroveň přiblížení. Vytvořená mapa obsahuje základní uživatelské rozhraní, které umožňuje mapu přibližovat pomocí tlačítek. Zároveň podporuje i ovládání pomocí prstových gest pro případné mobilní nasazení. Knihovna ale neposkytuje žádný mapový podklad. Třída očekává import podkladu za pomoci mapových dlaždic.

## Mapové dlaždice

Dříve se mapa běžně skládala z několika velkých obrázků, které měly dostatečně velkou kvalitu i pro účel přiblížení [22]. Tento způsob byl velice neefektivní, protože takovéto obrázky mají vysoké nároky na načítání. I z tohoto důvodu se používají mapové dlaždice (*tiles*). Jedná se o proces, kde se mapa rozdělí na spoustu malých částí s fixní velikostí. Při zobrazení mapy se načítají a ukazují pouze obrázky pokrývající zadanou geografickou část spojené v jeden celek. Běžně je načteno 20-30 obrázků (dlaždic) pro zobrazení jakékoliv lokace nezávisle na úrovni přiblížení. Tato technika umožňuje zobrazit mapy i na slabém hardwaru. Tento proces může být použit i pro jiné typy geografických dat například vektorové dlaždice [15]. Implementoval jsem celkem čtyři druhy mapových podkladů.

- **GrayScale** – jedná se o podklad od poskytovatele MapBox<sup>2</sup>. Podklad se dá silně upravit podle potřeby pomocí nástrojů nabízených na webových stránkách poskytovatele. Moje verze se snaží být co nejméně rušivá a je proto černobílá. Poskytuje uživatelům silný kontrast mezi záznamy a samotnou mapou. Aplikace si o jednotlivé dlaždice žádá pomocí přístupového tokenu od API MapBox. Při velkém množství požadavků si MapBox za načítání účtuje poplatky.
- **WorldImagery** – jediný implementovaný podklad používající satelitní fotografie. Poskytovatelem je Esri, který nabízí tyto dlaždice zdarma. Leaflet neumožňuje import podobných poskytovatelů. Řešením je modul *leaflet-providers*<sup>3</sup>, který obsahuje konfigurace pro načtení více poskytovatelů než nabízí samotná knihovna.
- **Voyager** – mapa s malým počtem popisů míst a oblastí. Oproti GrayScale se jedná o barevný podklad, který navíc zobrazuje například lesy. Poskytovatel je CARTO basemap<sup>4</sup> a pro implementaci bylo opět nutné použití modulu *leaflet-providers*.
- **Offline** – jedná se o jediný podklad, který je přímo součástí instalačního balíčku aplikace. Jednotlivé dlaždice jsou staženy pouze pro oblast České republiky a jsou načítány stejným způsobem jako mapy z externího zdroje. Touto implementací je zajištěna mapová dostupnost i při nízkém nebo žádném internetovém připojení.

## Mapové funkce

Pro splnění požadavků jsem implementoval níže popsané mapové funkce. Používají data z mobilní aplikace a přímou práci s instancí třídy **Map** z knihovny Leaflet. Z tohoto důvodu bylo praktické funkce implementovat právě v mapovém zobrazení pomocí jazyka JavaScript.

<sup>1</sup><https://leafletjs.com/examples.html>

<sup>2</sup><https://www.mapbox.com/>

<sup>3</sup><https://github.com/leaflet-extras/leaflet-providers>

<sup>4</sup><https://carto.com/location-data-services/basemaps/>

- **Správa záznamu** – aplikace je schopná vytvořit nález, umístit ho na mapu a následně jej upravit nebo odstranit. Pro tento účel slouží trojice funkcí `GPSPinAdd`, `GPSPinEdit` a `GPSPinDelete`. K bodu se připevní jeho klíč, název, souřadnice a status. Název a popis tvoří obsah vyskakovacího okna, které se uživateli zobrazí při označení nálezu. Klíč slouží k identifikaci bodu a při označení je poslán aplikaci prostřednictvím rozhraní, aby mohla reagovat na uživatelské požadavky pro zadaný bod. Atribut status není spravován prostřednictvím Android aplikace, ale webové aplikace. Umožňuje projevený zájem archeologa o nález zobrazit uživateli prostřednictvím změny barvy ikony nálezu. Základní typ záznamu v Leaflet knihovně nemá možnost si ukládat jakákoliv jiná data. Z tohoto důvodu jsem vytvořil vlastní třídu záznamu `CustomMarker`, který je schopný uchovat atribut klíče.
- **Sledování pohybu** – jedná se o funkci umožňující uživateli sledovat svůj pohyb na mapě. V kontextu činnosti detektorové prospekce má funkce význam zobrazení přesných úseků, které uživatel už prozkoumal. Implementace spočívá ve sledování polohy zařízení pomocí GPS v časových intervalech. Takto získané body jsou vždy spojeny s bodem předchozím (vyjma prvního záznamu). Vzniká tak trasa pohybu pomocí úseků mezi jednotlivými navazujícími body. Uživatel má možnost funkci pozastavit a ukončit. Všechny takto zaznamenané cesty jsou zobrazovány ve vrstvě „Cesty“.
- **Značení oblasti** – uživatel má možnost si vyznačit oblast. Význam funkce spočívá ve vyznačení si již prohledaných oblastí nebo oblastí, které chce uživatel prohledat. Při implementaci byl importován modul *Leaflet.draw*<sup>5</sup> do knihovny Leaflet, který umožňuje použití kreslicích funkcí. Podporuje tvary mnohoúhelníků, trojúhelníků, kruhů a dalších. Pro účely označení oblasti jsem použil právě mnohoúhelníky, které dovolují kreslit různé tvarované oblasti s vysokou přesností pomocí zadaných okrajových bodů. Funkce ovládá i možnost dokončení označení (spojí poslední a první bod). Zaznamenané oblasti jsou zobrazovány ve vrstvě „Oblasti“ a načítány a ukládány v interním úložišti (popsané v části 6.1.4).
- **Vrstvy** – uživatel má možnost vybrat si z různých mapových podkladů (představené v podsekcí 6.1.2) prostřednictvím mapového menu. Menu také umožňuje zobrazovat a skrývat další vrstvy, které obsahují všechny záznamy, uložené cesty, označené oblasti a historický podklad.

### 6.1.3 Přehled aktivit

Obrazovky jsou implementovány pomocí aktivit (popsány v sekci 4.4). Tyto prvky jsou základním stavebním prvkem aplikací. Níže popisují pouze zajímavější implementované aktivity a k čemu slouží.

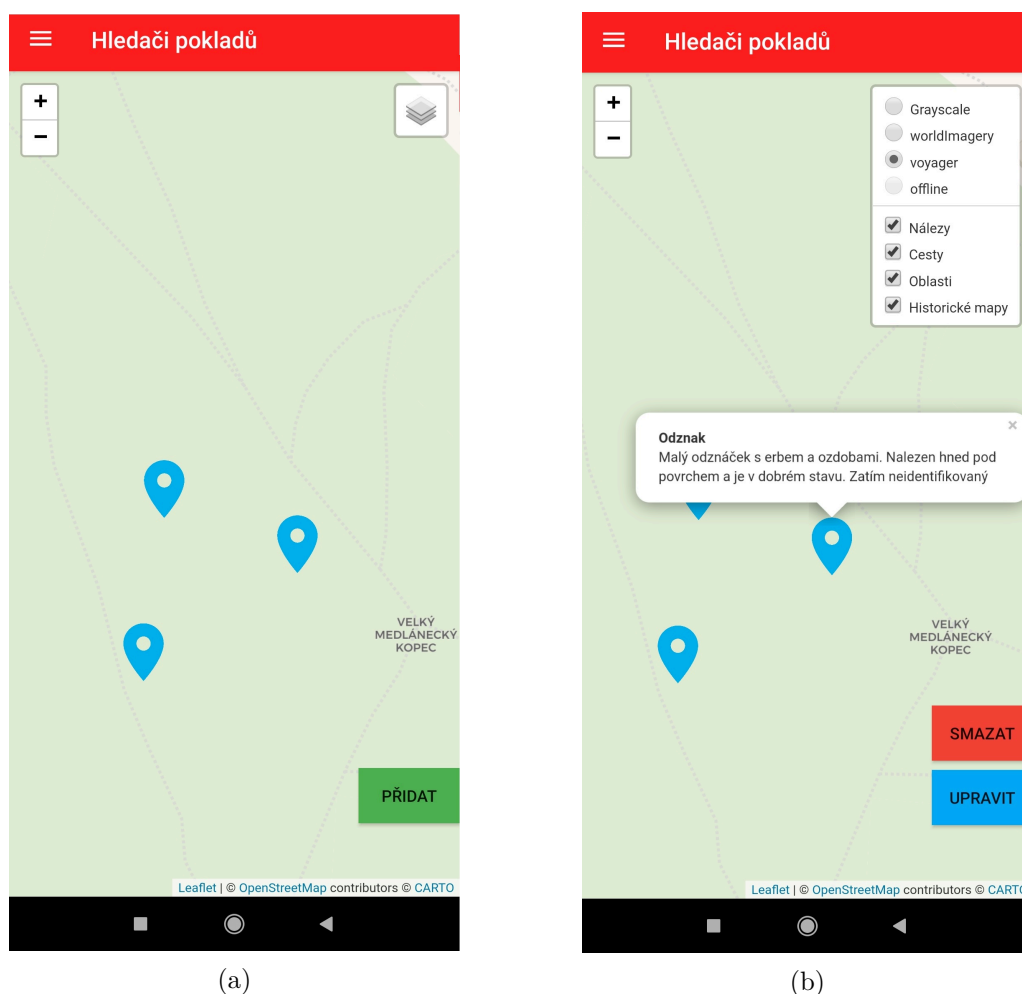
#### Hlavní aktivita

Hlavní aktivita, která je reprezentovaná třídou `MainActivity`, jako jediná propojuje aplikaci s mapovým zobrazením (implementuje i rozhraní popsané v části 6.1.2). Funguje také zároveň jako rozcestí do ostatních aktivit pomocí tlačítek a vysouvacího menu. Zobrazí se uživateli po úspěšném přihlášení a všechny ostatní aktivity se k ní můžou dostat pomocí tlačítka zpět. Důvodem pro použití modelu jedné hlavní aktivity je lepší přehlednost celé

<sup>5</sup><https://github.com/Leaflet/Leaflet.draw>

aplikace. Uživatel se nemůže tak lehce ztratit, pokud se bude chtít vrátit na dříve zobrazenou obrazovku.

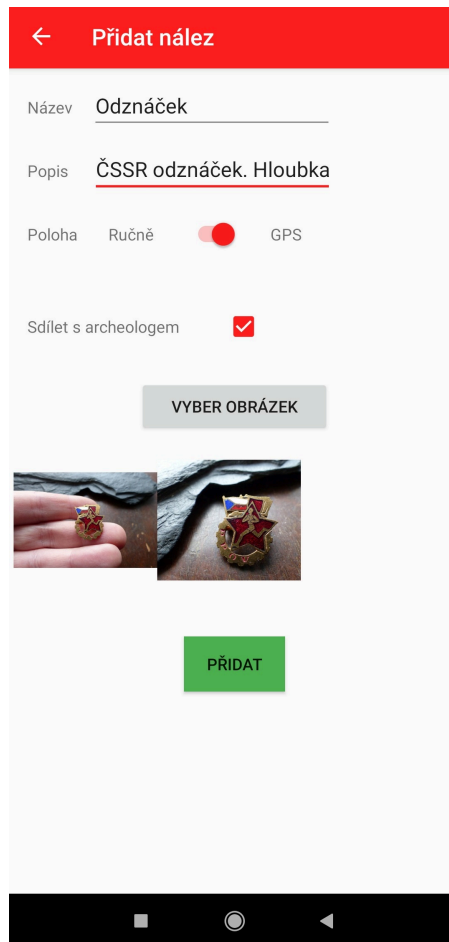
Aktivita také umožňuje uživateli vidět mapové zobrazení se všemi zaznamenanými nálezy, oblastmi a trasami (obrázek 6.2 a 6.3a). Mapa je interaktivní a reaguje na základní uživatelské pokyny jako je úprava přiblížení nebo posunování zobrazené oblasti. Obsahuje tlačítko „Přidat“, které uživatele přesune do aktivity pro správu nálezu (6.1.3). Po označení dříve zaznamenaného nálezu se přímo na mapě zobrazí základní uložené údaje (název a popis) a objeví se tlačítka pro úpravu nebo odstranění tohoto nálezu. Tato tlačítka nahradí „Přidat“ a uživateli se zobrazují pouze s funkcemi pro správu tohoto záznamu. Aktivita také implementuje vysouvací menu obsahující možnosti odhlášení, spuštění dalších mapových funkcí, spárování a zobrazení nastavení.



Obrázek 6.2: Obrázek (a) ukazuje hlavní obrazovku obsahující mapové zobrazení se třemi záznamy. Jednoduché uživatelské rozhraní umožňuje práci s mapou, zobrazení menu a přidávání nebo správu záznamu. Obrázek (b) ukazuje nastavení u mapy a označený nález. Základní data (název a popis) je zobrazen nad nálezem. Provedla se také změna tlačítek pro úpravu.



(a)



(b)

Obrázek 6.3: Obrázek (a) demonstruje funkci záznamu pohybu uživatele pomocí GPS. Funkce ukazuje i startovní a aktuální bod. Obrázek (b) ukazuje aktivitu pro správu záznamu. Uživatel může přidat až tři fotografie, název a popis. Určení polohy je provedeno ručně nebo pomocí GPS. Dále je uživateli umožněno nález sdílet.

### Aktivita pro správu záznamu

Aktivita (ukázaná na obrázku 6.3b) řeší možnost připnutí údajů k vytvořenému záznamu o nález. Uživatel je do ní přesměrován při žádosti o přidání nového nebo úpravu existujícího nález z hlavní aktivity. Nabízí uživateli vyplnit následující data:

- **Jméno** – jde o krátký název, který se bude používat pro lehkou orientaci v nálezech. Jméno nemusí být unikátní, aplikace si totiž vytváří vlastní unikátní ID pro každý nález. Jedná se o povinný údaj.
- **Popis** – jedná se o nepovinnou položku, která uchovává poznatky o nález nebo samotném výkopu. Je to také prostor určený pro specifikaci dat, které často archeolog potřebuje (hloubka nález, druh zeminy, atd.).



- **Poloha** – uživatel je povinen zaznačit místo nálezu. Má možnost si vybrat ze způsobů zadání tohoto údaje (GPS, vlastní zeměpisná šířka a výška, určení přímo na mapě). Pro možnost určení pomocí GPS musí uživatel souhlasit s poskytnutím těchto oprávnění na jeho zařízení a dané zařízení musí být touto službou vybaveno.
- **Sdílení** – z dotazníku 3.1.1 společně s cílem mé práce jasně vyplývá, že je nutné dát uživatelům možnost nález sdílet s archeologem. Právě zaškrťovací políčko pro sdílení určí, zda jsou všechna zadaná data o nález viditelná pro spárovaného webového uživatele. Určuje hodnotu „shared“ položky v databázi. Na základě této hodnoty má spárovaný webový uživatel možnost nález číst nebo nikoliv.
- **Fotografie** – jedním z požadavků na funkčnost je možnost mít u záznamu uložené fotografie. Aplikace umožňuje připsat maximálně tři fotografie, které se nahrají i na externí úložiště. Přesná adresa těchto fotografií je uložena v databázi u nález.

### Aktivita pro spárování uživatelů

Aktivita nabízí uživateli spárování s archeologem (webovým uživatelem). Beze spárování nemá zaškrťovací pole („Sdílet s archeologem“) u nález efekt. V první fázi, kdy uživatel není spárován, aktivita nabízí pouze tlačítko pro spárování a textový popis co samotné spárování znamená. Po stisku tlačítka se do databáze zapíší uživatelská data (unikátní ID, jméno a e-mail, se kterým se přihlásil). Také se jeho unikátní ID zapíše do tabulky pro spárování. Tato tabulka je následně zobrazována archeologům ve webové aplikaci, kde můžou spárování přijmout. Do této doby uživatel čeká na spojení s archeologem, jak mu nově zobrazená hláška oznamuje. Po přijetí aktivita zobrazuje potvrzení o spárování a základní údaje o archeologovi. Tímto funkčnost aktivity končí.

### Přihlašovací aktivita

Aktivita slouží pro přihlášení uživatele a realizuje ověřovací proces. Nabízí pouze přihlášení s použitím účtu Google. K ověření byla použita služba *play-services-auth*<sup>6</sup>, která umožní aplikaci využít ověřovací server Google. Po potvrzení správnosti zadaných údajů je uživatel přeměrován na hlavní aktivitu (podsekce 6.1.3). Při dalším spuštění aplikace je služba schopná ověřit správnost minulého přihlášení, a tak se uživatel nemusí znovu přihlašovat. Výjimkou je ruční odhlášení a odhlášení pomocí správy Google účtu.

#### 6.1.4 Lokální úložiště

Nálezy jsou uloženy v externí databázi, avšak pro zaznamenané oblasti stačí využití lokálního (interního) úložiště. Tato data nepotřebují být zálohovaná ani jejich možnost sdílení (na rozdíl od nálezů) nemá význam pro samotného uživatele. Android aplikace mají různé možnosti zápisu dat na zařízení (prozkoumané v sekci 4.5). V mém případě se vhodně nabízelo využít úložiště určené pro třídu *WebView*, která zprostředkovává mapové zobrazení. Právě mapa, nikoliv samotná aplikace, potřebuje mít možnost tato data číst a upravovat. Toto úložiště s názvem *Local Storage* je možno v instanci třídy používat po přidání oprávnění pro čtení a zápis dat (`setAllowFileAccess(true)`).

Do úložiště se zapíše každá vytvořená oblast jako série souřadnic s unikátním klíčem pro vyhledávání. Při spuštění aplikace se všechny oblasti načtou a zobrazí na mapě. Jejich

<sup>6</sup><https://developers.google.com/android/guides/setup>



zobrazená vizuální reprezentace obsahuje informaci nesoucí právě jejich unikátní klíč, díky kterému je smazání oblasti uživatelem promítnuto i v úložišti.

### 6.1.5 Uživatelské rozhraní

Výsledné uživatelské rozhraní je velice podobné vytvořené drátěné kostře v návrhu (popsané v části 5.2.2). Android Studio umožňuje vývoj GUI pomocí kódu v jazyce XML nebo graficky (aplikace výsledek vygeneruje v patřičný XML kód). Pro každou aktivitu bylo vytvořeno vlastní rozložení, které se může změnit na základě uživatelských interakcí (obrázek 6.2). Rozhraní cílí na jednoduchost orientace v aplikaci. Bylo také vyzkoušeno na různých velikostech dnešních mobilních telefonů pomocí testování na virtuálních strojích (funkce je součástí nástroje Android Studio). Tím byla ověřena správnost zobrazení rozložení na různých rozlišeních.

## 6.2 Databáze

Klíčovým prvkem pro komunikaci je databáze. Umožňuje sdílení dat mezi mobilní a webovou aplikací. Poskytuje také e-mailové adresy aplikacím pro další komunikaci. Průzkum možností databází jsem provedl v sekci 5.3.

### 6.2.1 Realtime Database

Rozhodl jsem se pro řešení z balíčku služeb *Firebase*. Právě služba *Realtime Database* (podsekce 5.3.2) umožňuje přeskočit veškerou implementaci na serveru a nabízí připravené funkce pro používání. O databázi jsem požádal prostřednictvím webového rozhraní pro služby *Firebase*<sup>7</sup>. Po registraci mobilní aplikace mi služba přidělila API klíč a byla mi vytvořena databáze.

Připojení databáze k aplikaci jsem provedl pomocí importování služby *firebase-database* a použití metody `getDatabase()`. Android Studio dokonce obsahuje tutoriál<sup>8</sup>, který celý proces nastavení a připojení zjednodušuje. Důležitou funkcí při získávání dat z databáze je `getReference`. Jako parametr požaduje přesnou cestu na požadovaná data. *Firebase* data pošle zpátky, pouze pokud má daný uživatel podle databázových práv oprávnění číst. K této referenci dat je možné připojit posluchače událostí, který při změně (přidat, odebrat, změnit, přesunout) v dané části databáze dostane upozornění od serveru a provede sérii příkazů zadanou vývojářem. Data dostává pomocí snímků (*snapshot*). Zanoření a vynoření probíhá pomocí metod `child()` a `parent()`.

Implementovaná databáze se dělí na 4 hlavní tabulky (složky). Tabulka *users*, která si pod uživatelskými unikátními klíči uchovává údaje, které pak sdílí se spárovaným webovým uživatelem. Jedná se o jméno a e-mailovou adresu. Složka *webusers* si uchovává data o webovém uživateli včetně ID spárovaných mobilních uživatelů. Tabulku *finds*, kde se pod uživatelským ID nachází jednotlivé nálezy spolu s jejich daty. Poslední tabulka *binds* tvoří ID mobilních uživatelů, kteří hledají archeologa. Toto rozložení umožňuje nastavit práva (popsána v sekci 6.2.3) a celkově zjednodušit objem posílaných dat při určitých dotazech.

<sup>7</sup><https://console.firebase.google.com>

<sup>8</sup><https://firebase.google.com/docs/database/android/start>

### 6.2.2 Úložiště

Další z použitých služeb *Firebase* je služba pro úložiště (*Firebase Storage*). Využil jsem ji na ukládání a načítání fotografií. Adresa k fotografii je uložena přímo u konkrétního nálezu. Toto URL je sdíleno s archeologem, pokud k tomuto nálezu má přístup. Fotografie se načte ve webové aplikaci na stránce daného nálezu.

### 6.2.3 Oprávnění

*Firebase Realtime Database* používá pro zabezpečení vlastní pravidla pro přístup do databáze [8], které lze na každou tabulku udělit zvlášť. Celkem obsahuje čtyři druhy pravidel:

- **read** – určuje, jestli a kdy mohou uživatelé data číst.
- **write** – určuje, jestli a kdy mohou uživatelé data zapisovat.
- **validate** – definuje, jak bude hodnota formátována, jaký budou mít data typ a jestli můžou obsahovat vnořené uzly.
- **indexOn** – určuje index vnořených dat pro podporu řazení a prohledávání.

Pro vlastní specifikaci přístupů jsem využil oprávnění pro čtení a zápis. Přístup mobilních uživatelů pouze k vlastním datům nebyl obtížný. Databáze totiž podporuje použití proměnných při specifikaci cesty u zadaných pravidel. Komplikace nastala z důvodu dvou typů uživatelů. Webový uživatel má totiž oprávnění k datům o nálezech pouze v případě, pokud je s majitelem dat spojen (seznam spojení je také v databázi) a zároveň je daný nález sdílený. Řešení spočívá v prohledání tohoto seznamu spojení databázovým serverem. Po nalezení spojení mezi těmito uživateli je přístup povolen. Tato zajímavá část pravidel je ukázána v úryvku kódu 6.3.

```
1 "finds": {  
2   "$uid": {  
3     ".write": "$uid === auth.uid",  
4     ".read": "$uid === auth.uid",  
5     "$id": {  
6       ".read": "root.child('webUsers').child(auth.uid).hasChild($uid) ===  
↪     true && data.child('shared').val() === 1",  
7       }  
8     }  
9   }  
}
```

Výpis 6.3: Úryvek ukazující pravidla přístupu pro tabulku *finds* v databázi. Specifikuje pravidla pro zápis a čtení jednotlivých podtabulek. Je použita možnost využití proměnných v cestě pro specifikaci uživatele. Právě `$uid` značí název podtabulky. Přístup je povolen, pokud je ID přihlášeného uživatele (značeno `auth.uid`) shodné s ID tabulky. Přístup ke konkrétnímu nálezu (řádek 6) je udělen webovému uživateli, pokud má autora nálezu ve svém seznamu spojení (`hasChild($uid)`) a hodnota položky *shared* je 1.

## 6.3 Webová aplikace

Důvod pro implementaci tohoto rozšíření mé bakalářské práce je popsán v sekci 3.1.2. Slouží pro správu sdílených záznamů archeology. Níže popisují implementaci vycházejícího z návrhu (popsané v části 5.4).

### 6.3.1 Použité technologie

Nejčastější technologie pro vytváření webové stránky jsou HTML, CSS a JavaScript [1]. Právě HTML má na starosti vytvoření kostry a rozložení. Definuje značky (*tagy*), které se následně pomocí CSS stylizují graficky a funkční část (psaná v jazyce JavaScript) se na ně odkazuje. Zvolil jsem stejný postup. Nejzajímavější implementační části jsou proto napsány v programovacím jazyce JavaScript.

### 6.3.2 Propojení s databází

Hlavní funkcí celé aplikace je zobrazovat údaje o nálezech zaznamenaných pomocí mobilní aplikace. Ty jsou uloženy v databázi (popsané v 6.2), a bylo nutné zajistit jejich dostupnost i pro webovou aplikaci. Připojení databáze jsem provedl použitím konfigurace vygenerované službou *Firebase*. Po zavolání funkce `firebase.initializeApp()` je s touto konfigurací navázáno spojení a je možno začít posílat jednotlivé požadavky na data. Volání dotazů probíhá stejně jako v mobilní aplikaci. Je nutné si zažádat o referenci na specifická data. Databáze zkontroluje, zda data existují a jestli k nim má žadatel oprávnění přistupovat. Při úspěšné kontrole pošle žadateli snímek dat a čeká na další požadavky.

### 6.3.3 Funkce

Webová aplikace implementuje funkce pro splnění požadované funkčnosti. Většina z nich pracuje s připojenou databází a interpretuje požadovaná data uživateli. Níže popisují ty nejdůležitější implementované funkce.

#### Přihlášení

Pro přihlášení má archeolog k dispozici přihlašovací formulář s kolonkami e-mail a heslo. Po stisknutí tlačítka „Přihlásit“ se aplikace připojí k autorizačnímu serveru *Firebase* pomocí příkazu `firebase.auth()`. Ten zadané údaje porovná se svojí databází registrovaných uživatelů a při shodě povolí uživateli přístup. Stránka uživatele při úspěšném přihlášení přesměruje na stránky s nálezy. Při každé další akci je znovu provedena kontrola přihlášení. Toto opatření znemožňuje přístup po odstranění oprávnění uživatele nebo samotný neoprávněný přístup.

#### Spárování

Funkce spárování slouží pro připojení archeologa k uživateli mobilní aplikace. Toto spojení umožní archeologovi vidět sdílené nálezy spárovaného uživatele. K realizaci této funkce slouží stránka „Spojení“. Posílá dotaz do databáze pro seznam uživatelů, kteří si zažádali o spárování a seznam již spárovaných s přihlášeným archeologem. Tyto seznamy následně zobrazí s možností nové spojení přijmout nebo aktivní spojení zrušit.

## Zobrazování nálezů

Stránka si zažádá o seznam všech aktivních spojení přihlášeného uživatele z databáze. Potom přes tento seznam uživatelských spojení prochází a žádá o všechny názvy a ID jejich sdílených nálezů. Z těchto dat následně vytváří tabulku, kterou po dokončení zobrazí. Vytvoří i odkazy pro zobrazení stránky daného nálezu a tlačítka pro reakci.

## Komunikace s uživatelem

Při stisku jednoho z tlačítek u nálezu se akce projeví v databázi a následně v mobilní aplikaci pomocí změny barvy ukazatele nálezu. Toto upozorní uživatele, zda má archeolog o daný nález zájem či nikoliv. Tato komunikace ovšem nestačí. Při projeveném zájmu bude archeolog chtít nález vidět a případně převzít. Pro tuto formu komunikace slouží e-mailová korespondence. Ta je zjednodušena pomocí funkce webové aplikace. Přímo u nálezu je totiž zobrazeno jméno nálezce. Toto jméno je zároveň odkazem, který otevře vybraného poštovního klienta s předvyplněnou adresou adresáta (mobilního uživatele) a předmětem zprávy upřesňujícím daný nález. Tato e-mailová adresa je uložena v databázi, když tento uživatel zažádá o spojení prostřednictvím mobilní aplikace.

### 6.3.4 Uživatelské rozhraní

Všechny implementované funkce jsou schované za uživatelským rozhraním. Hlavní části vizuální reprezentace jsou tvořeny pomocí technologie CSS s použitým nástrojem *Bootstrap*<sup>9</sup>. Výsledný obraz vychází z náčrtu 5.7. Pro navigaci mezi jednotlivými stránkami slouží hlavní menu na vrchní liště. Zbytek stránky je určen pro zobrazení požadovaného obsahu. Ten se nezobrazí ihned, protože o veškerá data se musí požádat z databáze a následně tato data interpretovat. Výsledné uživatelské zobrazení je ukázáno na obrázcích v příloze A.

---

<sup>9</sup><https://getbootstrap.com/>

## Kapitola 7

# Testování

Testování je nedílnou součástí vývoje softwaru. Provádí se za účelem získat informace o kvalitě vyvíjeného produktu, zjištění chyb a jestli splnil stanovené požadavky na funkčnost. Kapitola se věnuje postupu a získaným informacím z testování obou implementovaných aplikací. V první části je popsáno testování mobilní aplikace a získané výsledky. Druhá část se zaměřuje na ověření správnosti funkcionality webové aplikace.

### 7.1 Mobilní aplikace

Aplikaci jsem se rozhodl nejdříve testovat sám v různých zařízeních. Stejnou verzi jsem použil i pro testování s potenciálními uživateli z cílové skupiny. Ze zpětné vazby vyplývají návrhy úprav pro budoucí vývoj. Na závěr sekce je shrnutí získaných výsledků.

#### 7.1.1 Vlastní testování

Mobilní aplikace je určená na zařízení s operačním systémem Android verze 6.0 a vyšší. Toto omezení je dané v podmínkách pro instalaci a na nepodporovaná zařízení nepůjde instalovat. Na nejpoužívanějších podporovaných verzích (9.0, 8.1, 7.0, 6.0) jsem aplikaci testoval pomocí nástroje *Android Virtual Device* pro ověření funkčnosti. Také jsem testoval správnost zobrazení uživatelského rozhraní na jiných rozlišeních displeje. To se ukázalo přínosné, jelikož odhalilo chybu při zobrazování tlačítek na hlavní obrazovce s mapou. Na menších rozlišeních se v pozadí tlačítek totiž nachází smluvní ujednání poskytovatele dané mapy. Nepřesný pokus o zmáčknutí na tlačítko může způsobit nechtěné otevření adresy ve webovém prohlížeči. Provedl jsem proto úpravu pozice tlačítek vůči spodní části obrazovky.

Hlavní funkčnost aplikace jsem testoval pomocí svého telefonu (Android 9.0, rozlišení 1520 × 720). Funkce zahrnující získávání polohy pomocí GPS jsem prováděl v terénu i simulací pomocí aplikace Fake GPS<sup>1</sup>. Při testování ve venkovních podmínkách jsem si omezil rychlost datového připojení k internetu na 50 KB/s. Chtěl jsem tímto vyzkoušet náročnost a závislost aplikace na internetovém připojení. I s touto rychlostí se jednotlivé mapové podklady načítaly v použitelném čase (maximálně 20 vteřin). Avšak na prohlížení celé mapy, rychlé změny zobrazených lokací a podkladů nebo pro časté změny úrovně přiblížení není tato rychlost dostačující. I z tohoto důvodu byla implementována i offline mapa (popsaná v sekci 6.1.2). Práce s databází (vytvoření nového nálezu, jejich načítání, apod.) nebyla nízkou rychlostí výrazně ovlivněna. Nahrávání i stahování fotografií z úložiště bylo značně

---

<sup>1</sup><https://play.google.com/store/apps/details?id=com.lexa.fakegps>

pomalejší. Aplikace ale nečeká, než se fotografie na server nahrají a umožňuje práci i během tohoto procesu. Uživatelé po úspěšném i neúspěšném nahrávání zobrazí malé vyskakovací okénko (*toast*<sup>2</sup>) s výsledkem procesu. Tímto testováním jsem si ověřil funkčnost aplikace při špatných podmínkách internetového připojení.

### 7.1.2 Uživatelské testování

Po dokončení vlastního testování jsem nabídl použití aplikace třem aktivním hledačům. Vzhledem k lokacím jejich hledání, jsem přímo u používání aplikace nebyl přítomen. Všechny zmíněné poznatky proto pocházejí pouze ze zpětné vazby. Žádný z těchto detektorářů nepoužívá jinou aplikaci pro záznam polohy nálezů. Všichni tři už v minulosti s archeologem spolupracovali v rámci detektoristiky.

Prvním bodem bylo uživatelské rozhraní. Celé rozložení jsem cílil na jednoduchou orientaci a intuitivnost zobrazených prvků. Ze zpětné vazby vyplynulo, že navigace aplikací je jednoduchá a jasná bez jakékoliv pomoci. Mapové prvky jsou bez problému srozumitelné. Nejasnost byla shledána na obrazovce pro přidání nového nálezů. Možnost zaznamenat polohu ručně není objasněna a uživatel tak neví, jak funkce pracuje. Uživatel je totiž následně přesměrován zpět do mapového zobrazení a má určit polohu na mapě kliknutím. Řešením je implementovat informační okénko s vysvětlením, jak tato funkce funguje.

Další na řadě byla funkcionalita aplikace. Implementované funkce pro záznam cesty nebo označení oblasti byly označeny jako užitečné a jednoduše použitelné. Testování uživatelé by ale ocenili možnost připnutí vlastního názvu nebo data vytvoření k oblastem. Kladně vítaná byla možnost vybrat si z různých mapových podkladů. Problém se objevil na stránce pro správu nálezů. Uživatel má možnost určovat jestli je nález sdílený, ale neví, že nejdříve musí provést samotné spárování s archeologem. Z tohoto důvodu přidám vyskakovací okénko u prvního pokusu o sdílení nálezů s informací, že sdílení nemá efekt bez aktivního spojení s archeologem.

Posledním bodem byly připomínky na zlepšení aplikace. Uživatelé by ocenili obrazovku se seznamem svých nálezů s možnostmi vyhledávání. Pro zlepšení této funkce by se k nálezům mohly přidávat i klíčová slova (např. mince, odznak, apod.). Dalším nápadem na zlepšení byla možnost zobrazovat nálezy na mapě pomocí fotografií. Tuto funkci jsem původně zvažoval při implementaci, ale zavrhl z důvodu lepší přehlednosti a menší režie na načítání mapy.

### 7.1.3 Shrnutí

Výsledkem testování je potvrzení, že aplikace funguje dobře, je přehledná a není náročná na datové připojení. Z uživatelského testování (sekce 7.1.2) vyplývají převážně drobné úpravy funkcionality a návrhy na vylepšení pro budoucí vývoj aplikace. Nebyl objeven kritický problém způsobený neočekávaným chováním uživatele nebo konfigurací telefonu. Mobilní aplikace splnila všechny uživatelské požadavky (popsané v části 3.1.1) a byla označena jako užitečná při detektorové prospekci (testovanými uživateli). Po opravení nalezených chyb a případné implementaci navržených funkcí (vyplývajících z testování) je aplikace připravená pro veřejnou distribuci. Aplikace nesplnila jeden z požadavků na funkčnost vycházející ze zadání bakalářské práce. Jedná se o práci s historickými mapami. Aplikace sice implementuje testovací promítnutí historických map pomocí obrázků, ale nejedná se o plnohodnotné načtení mapy. Implementace této funkce má vysokou prioritu a je v pokročilém stavu.

---

<sup>2</sup><https://developer.android.com/guide/topics/ui/notifiers/toasts>

## 7.2 Webová aplikace

Webová aplikace slouží převážně pro zobrazování některých dat z databáze. Testování jsem proto založil na ověření správného čtení uložených dat získaných při testování mobilní aplikace (sekce 7.1). Prováděl jsem pouze vlastní testování bez dalších uživatelů.

Spojení s databází při běžném provozu nebylo problémové. Výjimku tvoří velký počet jednotlivých požadavků na data. Ty mohou způsobit krátké zpoždění zaslání dat a stránky se načtou dříve bez požadovaného obsahu. Řešením je počkání na data před vytvořením obsahu stránky. Funkce pro spárování uživatelů s archeologem fungovala bez problému. Nálezy z nových spojení se zobrazí na příslušné stránce bez delší prodlevy. Reakce na jednotlivé nálezy fungovaly v pořádku. Stránky konkrétních nálezů zobrazovaly všechny uživatelem zadané informace. Chybně se zobrazovaly pouze fotografie, pokud nebyly právě tři.

Bohužel se mi nepodařilo aplikaci testovat s archeology jakožto cílovou skupinou. Velkou systémovou změnou může být více veřejné řešení nálezů. Spojení totiž nemusí být vázáno přímo na jednoho archeologa, ale na celou skupinu řešící danou oblast. Před dalším vývojem ale bude nejprve nutné získat zpětnou vazbu na aktuální verzi. Webová aplikace splňuje funkci čtení dat z databáze a umožňuje jednoduchou komunikaci s nálezcem prostřednictvím e-mailové korespondence.

## Kapitola 8

# Závěr

Cílem mé bakalářské práce bylo vytvořit mobilní aplikaci pro hledače pokladů s detektorem kovu. Hlavním účelem aplikace bylo umožnit uživatelům zaznamenávat údaje o nálezech na mapě. Další cíl byl usnadnit hledání archeologa pro předání informací a případně samotného nálezu.

Na základě stanovených cílů jsem prostudoval problematiku archeologie, ústavní pohled na detektorovou prospekci, zjistil požadavky cílové skupiny a kontaktoval archeology. Z těchto setkání vznikl návrh na rozšíření zadání o implementaci další specializované aplikace. Ta bude schopná zobrazovat informace o nálezech nasbíraných pomocí mobilní aplikace a usnadní případnou další komunikaci. Dále jsem prozkoumal vývoj mobilních aplikací pro systém Android, existující řešení pro záznam dat a knihovny mapových API. Ze získaných poznatků jsem vytvořil model architektury s mobilní aplikací, databází jako komunikačním prvkem a webovou aplikací. Na základě těchto informací jsem vytvořil návrh samotných aplikací včetně uživatelského rozhraní a jednotlivých funkcí. Při návrhu byl kladen důraz především na jednoduchost orientace a jednoznačnost funkcí.

V mobilní aplikaci jsem implementoval mapové zobrazení (použitím knihovny Leaflet), funkce pro práci s mapou podle uživatelských požadavků a připojení k databázi *Firestore Realtime Database*. Všechna data o nálezech jsou ukládána do uživatelské části databáze kromě fotografií, které jsou nahrané na úložiště. Webová aplikace tato data čte pouze, pokud je nález se spojeným archeologem sdílený (určuje uživatel). K přihlašování je použit Google účet uživatele. Správnost přihlášení je ověřena pomocí služby *Firestore Authentication*. Další uživatelská data (označené zóny a uložené trasy) jsou ukládána do paměti zařízení.

Vytvořená aplikace splňuje všechny uživatelské požadavky na funkčnost, byla úspěšně otestována a po realizaci navržených úprav je připravena k distribuci. Celkově dosáhla stanovených cílů. Ve vývoji aplikace budu nadále pokračovat a reagovat na další podněty ze strany uživatelů. Webová aplikace plní svoji hlavní funkci zobrazování dat a ulehčení komunikace, ale nebyla zatím otestována archeology.

Výsledkem celé práce je systém, který je veřejně prospěšný. Díky němu se totiž uchová více dat o nálezech, které jsou nosičem nenahraditelných informací. Tyto informace jsou klíčem k historii naší země a díky mé práci se jednoduše dostanou do pověřených rukou archeologů. Po dokončení delšího testování v rámci Jihomoravského kraje bude systém představen jednotlivým archeologickým střediskům ostatních krajů České republiky.



# Literatura

- [1] ALOK RANJAN, ABHILASHA SINHA, RANJIT BATTEWAD. *JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript (English Edition)*. 1. vyd. BPB Publications, 2020. ISBN 9789389328721.
- [2] ANDROID. *Codenames, Tags, and Build Numbers* [online]. 2020 [cit. 2020-4-10]. Dostupné z: <https://source.android.com/setup/start/build-numbers>.
- [3] ARCHEOLOGIE - MUŠOV. *Archeologie* [online]. 2015 [cit. 2020-4-10]. Dostupné z: <https://www.archeologiemusov.cz/archeologie>.
- [4] ARCHEOLOGIE - MUŠOV. *Detektorové hledání* [online]. 2015 [cit. 2020-4-10]. Dostupné z: <https://www.archeologiemusov.cz/hledani-detektorem/>.
- [5] BERNZWEIG, D. *The History of the Metal Detector* [online]. 2014 [cit. 2020-4-10]. Dostupné z: <https://www.metaldetector.com/learn/metal-detector-history/history-of-the-metal-detector>.
- [6] DB-ENGINES. *DB-Engines Ranking* [online]. 2020 [cit. 2020-6-10]. Dostupné z: <https://db-engines.com/en/ranking>.
- [7] DISCOVER DETECTING. *How Does a Metal Detector Work?* [online]. 2016 [cit. 2020-4-10]. Dostupné z: <https://www.discoverdetecting.com/how-does-a-metal-detector-work/>.
- [8] FIREBASE. *Understand Firebase Realtime Database Rules* [online]. 2020 [cit. 2020-6-10]. Dostupné z: <https://firebase.google.com/docs/database/security>.
- [9] GOOGLE PLAY. *My Google Maps* [online]. 2019 [cit. 2020-6-10]. Dostupné z: <https://play.google.com/store/apps/details?id=com.google.android.apps.m4b>.
- [10] JOHNSON, J. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 1. vyd. Morgan Kaufmann Publishers Inc., 2010. ISBN 9780123750303.
- [11] KOMORÓCZY, B. Obecná úvaha na téma utajení jako nástroj ochrany v archeologii. *Zprávy památkové péče*. 1. vyd. 2018, roč. 78, č. 1, s. 24–29. ISSN 1210-5538.
- [12] KOMORÓCZY, B. Revize stavu archeologických komponent v trati Drnholec „Holenická pole“ pomocí prospekčních a málo invazivních výzkumných metod. *Přehled výzkumů*. 1. vyd. 2019, roč. 60, č. 2, s. 23–31. ISSN 1211-7250.

- [13] LARDINOIS, F. *Kotlin is now Google's preferred language for Android app development* [online]. 2019 [cit. 2020-4-10]. Dostupné z: <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/>.
- [14] MANCHANDA, A. *Why Kotlin is the Future of Android Application Development* [online]. 2020 [cit. 2020-6-10]. Dostupné z: <https://www.netsolutions.com/insights/why-kotlin-is-the-future-of-android-app-development/>.
- [15] MAPBOX. *Vector tiles* [online]. 2020 [cit. 2020-4-10]. Dostupné z: <https://docs.mapbox.com/vector-tiles/reference/>.
- [16] MARCIN MOSKALA, I. W. *Android Development with Kotlin*. 1. vyd. Packt Publishing Limited, 2017. ISBN 1787123685.
- [17] MINISTERSTVO KULTURY ČESKÉ REPUBLIKY. *Ochrana archeologického dědictví* [online]. 2019 [cit. 2020-4-10]. Dostupné z: <https://www.mkcr.cz/ochrana-archeologickeho-dedictvi-275.html>.
- [18] MODELSKI, T. *The Polish Contribution to the Ultimate Allied Victory in the Second World War*. 1. vyd. Tadeusz Modelski, 1988. ISBN 09-511-1710-6.
- [19] RAGHAVENDRA RAO K. *Kotlin for Enterprise Applications using Java EE*. 1. vyd. Packt Publishing, 2018. ISBN 9781788997270.
- [20] RAHMAN, M. *Android Version Distribution statistics will now only be available in Android Studio* [online]. 2020 [cit. 2020-4-10]. Dostupné z: <https://www.xda-developers.com/android-version-distribution-statistics-android-studio/>.
- [21] STATCOUNTER. *Mobile Operating System Market Share Worldwide* [online]. 2020 [cit. 2020-6-10]. Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- [22] STERLING QUINN, JOHN A. DUTTON E-EDUCATION INSTITUTE, COLLEGE OF EARTH AND MINERAL SCIENCES, THE PENNSYLVANIA STATE UNIVERSITY. *Why tiled maps?* [online]. 2020 [cit. 2020-4-10]. Dostupné z: <https://www.e-education.psu.edu/geog585/node/706>.
- [23] TYSON, J. *How Metal Detectors Work* [online]. 2001 [cit. 2020-4-10]. Dostupné z: <https://electronics.howstuffworks.com/gadgets/other-gadgets/metal-detector.htm>.

## Příloha A

# Uživatelské rozhraní

Nové

Uložené

Chci vidět

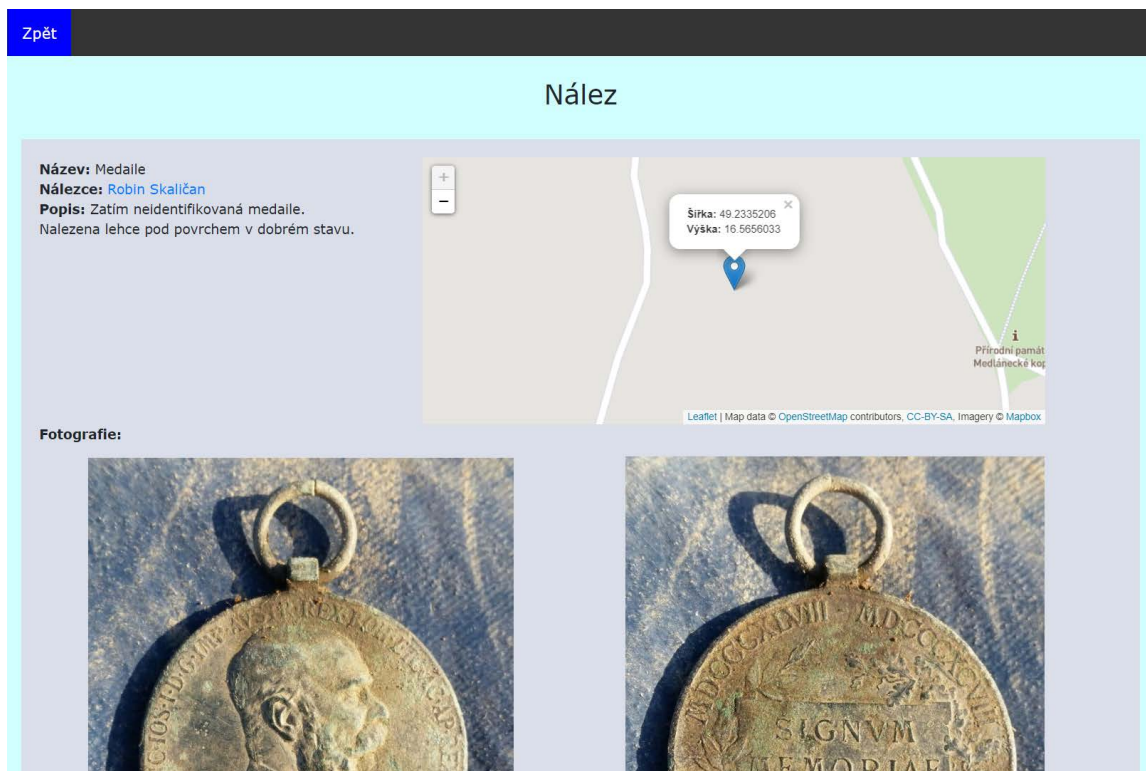
Spojení

Odhlásit se

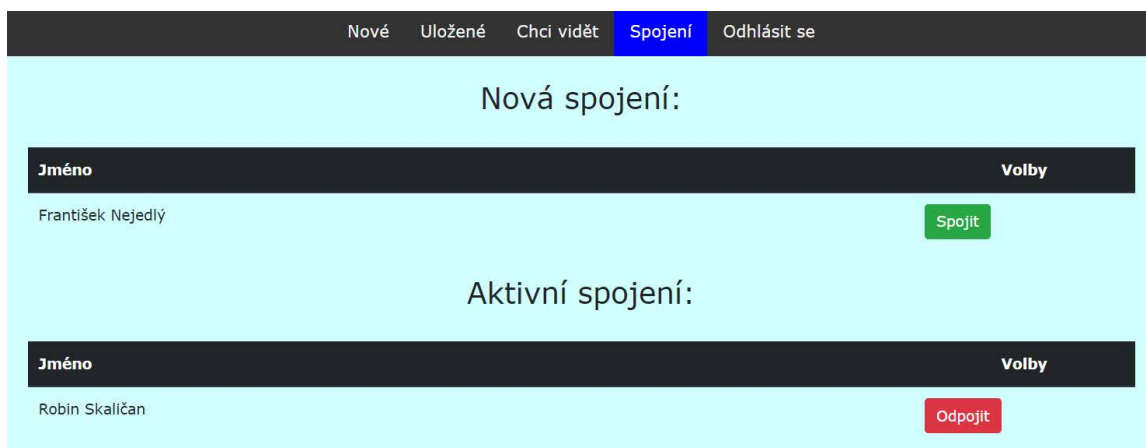
Nové nálezy:

Název nálezu	Jméno nálezce	Volby		
Rezavý šíp	Robin Skaličan	Chci vidět	Uložit	Nezajímá mě
Odznáček	Robin Skaličan	Chci vidět	Uložit	Nezajímá mě
Zlatá mince	Robin Skaličan	Chci vidět	Uložit	Nezajímá mě

Obrázek A.1: Stránka slouží pro zobrazení nových nálezů. Uživatel má možnost pomocí jména nálezu být přesměrován na stránku zobrazující veškeré údaje o nálezu, napsat e-mail nálezci (odkaz v jeho jméně) a reagovat na nález pomocí tlačítek. Pouze sdílené nálezy spárovaných uživatelů bez předešlé reakce jsou zobrazovány.



Obrázek A.2: Obrázek ukazuje vzhled stránky pro konkrétní nález. Zobrazuje uživateli všechna data z databáze a případné fotografie z úložiště. Zeměpisná šířka a výška je zobrazena pomocí mapy.



Obrázek A.3: Stránka sloužící pro správu spojení. Zobrazuje tabulku s uživateli mobilní aplikace, kteří se chtějí s archeologem spojit a tabulku aktivních spojení. Archeolog má možnost nová spojení přijmout a aktivní zrušit.

<a href="#">Nové</a> <a href="#">Uložené</a> <a href="#">Chci vidět</a> <a href="#">Spojení</a> <a href="#">Odhlásit se</a>		
Nové nálezy:		
Název nálezu	Jméno nálezce	Volby
Bronzová spona	František Nejedlý	<a href="#">Chci vidět</a> <a href="#">Uložit</a> <a href="#">Nezajímá mě</a>
Rezavý šíp	Robin Skaličan	<a href="#">Chci vidět</a> <a href="#">Uložit</a> <a href="#">Nezajímá mě</a>
Odznáček	Robin Skaličan	<a href="#">Chci vidět</a> <a href="#">Uložit</a> <a href="#">Nezajímá mě</a>
Zlatá mince	Robin Skaličan	<a href="#">Chci vidět</a> <a href="#">Uložit</a> <a href="#">Nezajímá mě</a>

Obrázek A.4: Obrázek ukazuje stejnou stránku jako obrázek A.1, ale navíc s nálezem od nově spojeného uživatele pomocí funkce spojení (obrázek A.3).

## Příloha B

# Obsah přiloženého DVD

- **/app/app.apk** – Instalační balíček mobilní aplikace pro systém Android.
- **/app/app.zip** – Komprimované zdrojové soubory mobilní aplikace pro *Android Studio*.
- **/app/app-javadoc.zip** – Komprimovaná dokumentace zdrojového kódu mobilní aplikace (javadoc).
- **/app/manual.pdf** – Manuál k používání mobilní aplikace.
- **/text/BP.zip** – Komprimované zdrojové soubory technické zprávy (L<sup>A</sup>T<sub>E</sub>X).
- **/text/xskali19-BP.pdf** – Technická zpráva.
- **/text/xskali19-BP-print.pdf** – Technická zpráva ve formátu pro tisk.
- **/web/web.zip** – Komprimované zdrojové soubory webové aplikace.
- **/web/manual.pdf** – Manuál k používání mobilní aplikace.
- **/readme.txt** – Soubor s popisem obsahu DVD.