

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra geoinformatiky

**NÁVRH A VÝVOJ
PASPORTIZAČNÍHO KLIENTA PRO MOBILNÍ
PLATFORMU ANDROID**

Bakalářská práce

Jan ROLLER

Vedoucí práce Mgr. ROSTISLAV NÉTEK, Ph.D.

Olomouc 2020

Geoinformatika a geografie

ANOTACE

Tato bakalářská práce je zaměřena na vývoj pasportizačního klienta pro mobilní platformu Android převážně v programovacím jazyce Java. Cílem je vytvořit univerzální pasportizační mapovou aplikaci pro mobilní zařízení, která splňuje veškeré zásady vývoje těchto druhů softwaru.

Tyto zásady vývoje aplikací pro mobilní zařízení jsou v teoretické části této práce definovány stejně tak jako technologické aspekty a zásady tvorby mapových aplikací. Teoretická část této práce dále obsahuje obecné informace o pasportizaci a pasportech a je zde také popsán současný stav mapových aplikací vhodných k zobrazení dat z pasportů.

Hlavním výsledkem práce je univerzální mapový pasportizační klient pro systém Android a výrobní postup tvorby tohoto klienta.

KLÍČOVÁ SLOVA

pasportizace; pasport; smartphone; mobilní zařízení; Android; Java

Počet stran práce: 64

Počet příloh: 7 (z toho 2 volné a 2 elektronické)

ANOTATION

This bachelor thesis is focused on development of an evidence map client for mobile devices running on Android platform coded mainly in Java programming language. The aim of this bachelor thesis is to create a universal evidence map application for mobile devices which meets all the principles needed for development of such kinds of software.

These principles of development of applications for mobile devices are defined in theoretical part of this thesis as well as the technological aspects and principles for development of map applications. Apart from that, the theoretical part of this thesis contains general information on gathering evidence data, and the current status of map applications suitable for displaying inventory data from evidence is described in this thesis as well.

The main result of this thesis is the universal evidence map client for Android and the description of the process of development of this client.

KEYWORDS

evidence; mobile device; smartphone; Android; Java

Number of pages: 64

Number of appendixes: 7

Prohlašuji, že

- bakalářskou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu,

- jsem si vědom, že na moji bakalářskou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevydělečně, ke své vnitřní potřebě, bakalářskou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk bakalářské práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé bakalářské práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne

Jan Roller

Děkuji vedoucímu práce Mgr. Rostislavu Nétkovi, Ph.D. za poskytnutá data, podněty a připomínky při vypracování práce.

Dále děkuji všem testerům, zejména Lukáši Pospíšilovi a Ondřeji Hubáčkovi za jejich spolupráci a zpětnou vazbu při výkonnostním testování aplikace.

Za odborné rady týkající se pasportizace v zákoně děkuji Mgr. Janu Kopeckému.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan ROLLER**

Osobní číslo: **R16405**

Studijní program: **B1301 Geografie**

Studijní obor: **Geoinformatika a geografie**

Název tématu: **Návrh a vývoj pasportizačního klienta pro mobilní platformu Android**

Zadávací katedra: **Katedra geoinformatiky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je sestavit univerzální pasportizační klient pro mobilní platformu Android. Student provede rešerši stávajících mobilních mapových klientů, a specifikuje hardwarové, softwarové, geoinformatické a kartografické požadavky. Na základě výběru vhodného vývojového prostředí a vhodně zvolených metod vývoje softwaru, bude hlavním cílem práce návrh a vývoj mobilní aplikace umožňující online i offline přístup, pochopitelně s důrazem na dodržování zásad UX/UI mobilních řešení. Pilotní studie bude obsahovat data dodané vedoucím práce, na jejíž implementaci student provede uživatelské i výkonostní testování. Vedle vlastního vývoje aplikace se student v teoretické části zaměří na specifikaci vývojových postupů a metod vývoje softwaru pro mobilní aplikace (nativní/hybridní/PWA, vodopádový vs. agilní model, iterace, prototypování, verzování apod).

Student vyplní údaje o všech datových sadách, které vytvořil nebo získal v rámci práce, do Metainformačního systému katedry geoinformatiky a současně vytvoří zálohu údajů ve formě validovaného XML souboru. Celá práce (text, přílohy, výstupy, zdrojová a vytvořená data, XML soubor) se odevzdá v digitální podobě na CD (DVD) a text práce s vybranými přílohami bude odevzdán ve dvou svázaných výtiscích na sekretariát katedry. O diplomové práci student vytvoří webovou stránku v souladu s pravidly dostupnými na stránkách katedry. Práce bude zpracována podle zásad dle Voženílek (2002) a závazné šablony pro diplomové práce na KGI. Povinnou přílohou práce bude poster formátu A2.

Rozsah grafických prací: dle potřeby
Rozsah pracovní zprávy: max. 50 stran
Forma zpracování bakalářské práce: tištěná
Seznam odborné literatury:

- Allen, G. **Android 4: průvodce programováním mobilních aplikací**. 1. vyd. Překlad Jakub Mužík. Brno: Computer Press, 2013, 656 s. ISBN 978-80-251-3782-6.
- Goldman R., Gabriel R.P. (2005) **How To Do Open-Source Development: In Innovation Happens Elsewhere: Open Source as Business Strategy**. Z. Mednieks, L. Dornin, G.B. Meike, M. Nakamura. **Programming Android**. O'Reilly Media, 2011.
- Murphy, Mark L. **The busy coder's guide to Android development** [online]. 2nd ed. United States: CommonsWare, 2009, c2008-2009., xix, 443 p. [cit. 2015-03-09]. ISBN 978-098-1678-009.
- Schildt, Herbert. **Mistrovství - Java**. 1. vyd. Brno: Computer Press, 2014, 1224 s. ISBN 978-80-251-4145-8.
- Sharma, S., Sugumaran, V. & Rajagopalan, B. (2002). **A framework for creating hybrid-open source software communities**. *Information Systems Journal* 12 (1), 7-25.
- Voženílek, Vít. **Diplomové práce z geoinformatiky**. Univerzita Palackého, 2002.

Vedoucí bakalářské práce: **Mgr. Rostislav Néték, Ph.D.**
Katedra geoinformatiky

Datum zadání bakalářské práce: **11. května 2018**
Termín odevzdání bakalářské práce: **13. května 2019**

doc. RNDr. Martin Kubala, Ph.D.
děkan

L.S.

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA GEONFORMATIKY
17. listopadu 50, 771 46 Olomouc

prof. RNDr. Vít Voženílek, CSc.
vedoucí katedry

V Olomouci dne 15. května 2018

OBSAH

SEZNAM POUŽITÝCH ZKRATEK	10
ÚVOD	11
1 CÍLE PRÁCE.....	12
2 METODY A POSTUPY ZPRACOVÁNÍ.....	13
3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	15
3.1 Pasport a pasportizace	15
3.1.1 Typy pasportů	16
3.1.2 Proces pasportizace	18
3.2 Mobilní platforma Android	21
3.2.1 Java.....	21
3.2.2 Android Studio.....	22
3.3 Datové formáty JSON a GeoJSON	22
3.3.1 JSON	22
3.3.2 GeoJSON	22
3.4 Existující mobilní mapové aplikace.....	23
3.5 Webová řešení	24
3.5.1 HTML a CSS	24
3.5.2 PHP	25
3.5.3 JavaScript.....	25
3.6 Vývojové postupy a metody vývoje softwaru pro mobilní zařízení	26
3.6.1 Vývojové postupy při tvorbě mobilních aplikací	26
3.6.2 Metody vývoje softwaru	28
3.6.3 Životní cyklus	29
3.6.4 Shrnutí.....	31
4 MOTIVACE	32
5 ASPEKTY TVORBY MOBILNÍCH MAPOVÝCH APLIKACÍ	34
5.1 Výběr aspektů	35
5.1.1 Hardwarový a softwarový aspekt.....	35
5.1.2 Datový aspekt	36
5.1.3 Uživatelský aspekt	37
5.1.4 Koncepční aspekt.....	37
5.1.5 Geoinformační aspekt	38
5.1.6 Vizualizační a estetický aspekt	38
5.2 Zásady UX/UI mobilních řešení	40
5.2.1 Charakteristika	40
5.2.2 Srovnání	41
6 PILOTNÍ STUDIE – APLIKACE PASPORTY.....	42
6.1 Návrh aplikace.....	42
6.2 Vývoj aplikace.....	43
6.2.1 Soubor AndroidManifest.xml	44
6.2.2 Soubor build.gradle.....	44

6.2.3	Úvodní obrazovka.....	45
6.2.4	Hlavní menu	46
6.2.5	Online část – Online Editor.....	47
6.2.6	Offline část – Offline Prohlížeč	50
6.2.7	Ostatní	52
6.3	Webové rozhraní pro upload dat	52
7	TESTOVÁNÍ.....	54
7.1	Výkonnostní testování.....	54
7.1.1	Výsledky výkonnostního testování	56
7.2	Uživatelské testování.....	56
7.2.1	Struktura dotazníku a otázky	56
7.2.2	Výsledky uživatelského testování	57
8	VÝSLEDKY	60
9	DISKUSE	63
10	ZÁVĚR	64
	POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE	
	PŘÍLOHY	

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
API	Application Programming Interface
APK	Android Application Package
CSS	Cascading Style Sheets
ČÚZK	Český Úřad Zeměměřičský a Katastrální
GIS	Geografický Informační Systém
GIT	Geoinformační Technologie
GNSS	Global Navigation Satellite System
GNU	GNU's Not Unix!
GNU GPL	GNU General Public License
GPS	Global Positioning System
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
OSM	OpenStreetMap
PHP	PHP: Hypertext Preprocessor
PWA	Progresivní Webová Aplikace
px	pixel
QA	Quality Assurance
RAM	Random Access Memory
RC	Release Candidate
SDK	Software Development Kit
UI	User Interface
USGS	United States Geological Survey
UX	User Experience
W3C	World Wide Web Consortium
WGS84	World Geodetic System 1984
WMS	Web Map Service
XML	Extensible Markup Language

ÚVOD

Pasporty jsou velmi důležité dokumenty úzce spjaté s informačními systémy o území. Jsou využívány všemi obcemi v České republice, protože jsou některé pasporty ze zákona povinné. Obce byly zvyklé vést si pasport svého majetku v papírové podobě, ovšem s nástupem moderních technologií, GIT a GIS se tvorba pasportů provádí digitálně a jsou takto i vedeny. Zhotovení pasportu a zpracování jeho případných aktualizací je finančně a technologicky náročné a na trhu operuje mnoho firem zabývajících se touto problematikou.

Existuje nespočet mobilních mapových aplikací, v nichž by se pasport dal vytvořit nebo upravit, některé z těchto aplikací jsou používány samotnými společnostmi, které pasporty vytvářejí. Tyto aplikace jsou ovšem příliš odborné, podrobné a složité na ovládání pro obvyčejného uživatele.

Z tohoto důvodu si autor této práce stanovil za cíl navrhnout jednoduchý, uživatelsky přívětivý mobilní mapový pasportizační klient pro obvyčejné uživatele, v nichž se pasport bude dát zobrazit a bude možné jej upravovat bez nutnosti využívat komerčních služeb a zbytečně tak utrácet finanční zdroje.

1 CÍLE PRÁCE

Hlavním cílem bakalářské práce je sestavit univerzální pasportizační klient pro mobilní platformu Android. Práce sestává z několika částí rozčleněných do deseti kapitol. Nejprve je zjištěn a popsán současný stav řešené problematiky a poté je na základě různých technologických aspektů a zásad realizován samotný vývoj mapového pasportizačního klienta pro mobilní platformu Android a je také provedeno jeho testování.

V teoretické části je provedena rešerše stávajících mobilních mapových klientů a specifikace hardwarových, softwarových, geoinformatických a kartografických požadavků. Taktéž jsou zde specifikovány vývojové postupy a metody vývoje softwaru pro mobilní zařízení (nativní/hybridní/PWA, vodopádový vs. agilní model, iterace, prototypování, verzování apod.).

V praktické části je hlavním cílem práce, na základě výběru vhodného vývojového prostředí a vhodně zvolených metod vývoje softwaru, návrh a vývoj mobilní aplikace umožňující online i offline přístup s důrazem na dodržování zásad UX/UI mobilních řešení. Pilotní studie obsahuje data dodaná vedoucím práce, na jejichž implementaci je prováděno výkonnostní a uživatelské testování.

Vytvořená mobilní mapová pasportizační aplikace umožní uživatelům zobrazit, prohlížet a jednoduše editovat atributy polygonových, liniových a bodových vrstev rozličných pasportů v reálném čase online i offline.

2 METODY A POSTUPY ZPRACOVÁNÍ

Na začátku vypracovávání bakalářské práce a před volbou dalšího postupu a metod je důležitá rešerše odborné literatury a dalších zdrojů týkajících se tématu. Tyto byly zaměřeny na tvorbu a programování mobilních aplikací, Android a jazyk Java. Důležité bylo také seznámit se s pojmem pasportizace, což autor této bakalářské práce podstoupil během povinné praxe. Teprve poté byl stanoven postup celé práce, metody a byl také vybrán software využitý při tvorbě výsledné aplikace.

Použité metody

Po provedení rešerše, jejíž součástí bylo i studium odborné literatury, kde se autor seznámil s jazykem Java a programováním aplikací pro systém Android, bylo přistoupeno k samotné tvorbě. Byla *navržena a naprogramována hybridní mobilní aplikace* pro mobilní zařízení se systémem Android.

Současně s tvorbou aplikace bylo prováděno jednak *výkonnostní testování aplikace* jakožto součást agilního přístupu vývoje softwaru a za druhé bylo provedeno také *uživatelské testování* aplikace, kde bylo využito *dotazníkového šetření*, v němž respondenti po otestování aplikace odpovídali na autorovy otázky týkající se této aplikace.

Použitá data

V bakalářské práci byly použity dva druhy dat. Testovací data ve formátu JSON byla poskytnuta vedoucím práce, Mgr. Rostislavem Nétkem, PhD., byla nahrána do aplikace a bylo na nich provedeno testování. Tato data pochází z projektu Inovačního Voucheru Olomouckého kraje pro obec Příkazy v okrese Olomouc.

Mimo tato tematická data bylo v samotné aplikaci použito několik podkladových vrstev z různých zdrojů, mezi nimiž si uživatel může přepínat:

- „OpenStreetMap“ od OpenStreetMap Foundation
- „Mapy.cz“ – základní mapa od Mapy.cz
- „Letecká“ podkladová mapa od Mapy.cz
- „Letecká ČÚZK“ – od ČÚZK
- „Základní ČÚZK“ – základní mapa ČR z ČÚZK
- „OpenTopo“ – topografická mapa od serveru OpenTopoMap
- „USGS Satelitní“ – satelitní snímek od USGS

Jako volitelná překryvná vrstva byla použita „Katastrální mapa“ od ČÚZK.

Použité programy

Mobilní mapová aplikace (pasportizační klient) byla napsána převážně v programovacím jazyce Java, ve vývojovém prostředí Android Studio. Dílčí části aplikace byly naprogramovány ve skriptovacích jazycích JavaScript a PHP a značkovacím jazyce HTML se styly CSS ve vývojovém prostředí Visual Studio Code. Grafika v aplikaci, loga a ikony byly vytvořeny v programech GIMP a Inkscape. „Testovací data 2“ byla vytvořena v programu ArcMap 10.6. Grafy a diagramy byly vytvořeny v programech balíku Microsoft Office.

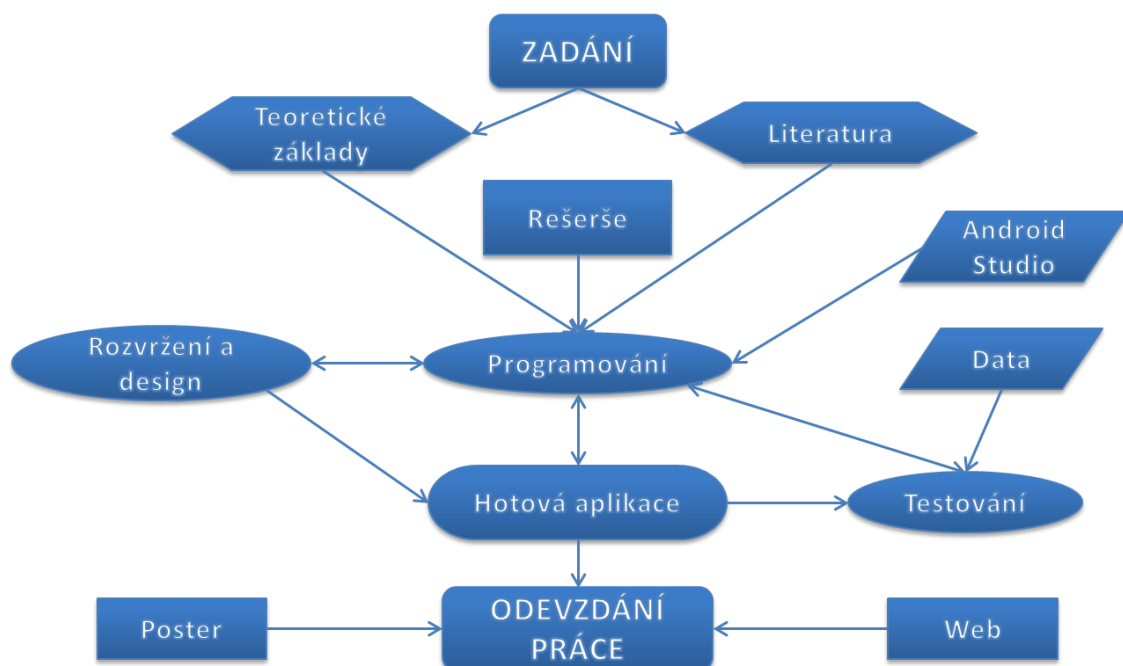
Postup zpracování

Nejprve bylo nutné popsat a vysvětlit jednotlivá klíčová slova jako Android či pasportizace, bez nichž by se tato bakalářská práce neobešla. Dále bylo potřeba prozkoumat trh a zjistit, jaké podobné mapové aplikace schopné zobrazit a editovat data ve formátu JSON existovaly. Bylo provedeno jejich testování a zhodnocení.

Následně byly v teoretické části definovány jednotlivé technologické aspekty těchto mapových aplikací – aspekty zařízení, tedy aspekty hardwarové (rozlišení a orientace obrazovky, výkon a úložiště zařízení, vybavenost zařízení GPS a přístupem k Internetu), pak také aspekty softwarové, datové, geoinformatické, kartografické, uživatelské apod. Tyto aspekty poté byly zhodnoceny a byl na ně kladen důraz při tvorbě vlastního pasportizačního klienta.

Před samotnou tvorbou pasportizační aplikace bylo však ještě nutné specifikovat jednotlivé vývojové postupy a metody vývoje softwaru pro mobilní zařízení a definovat určité pojmy a přístupy při vývoji softwaru a jejich rozdíly. Až poté bylo na základě výše zmíněných poznatků přistoupeno ke tvorbě – tedy k programování a kódování – mapového pasportizačního klienta pro platformu Android. Byl vyřešen požadovaný online a offline přístup k zobrazovaným datům. Během tohoto procesu byl popsán postup tvorby tohoto mapového pasportizačního klienta.

Následně bylo provedeno uživatelské a výkonnostní testování na poskytnutých datech a na základě tohoto testování a zpětné vazby uživatelů byla aplikace opravována a vylepšována. Bylo sepsáno zhodnocení testování a závěr práce. Poté byly vytvořeny webové stránky, které mimo samotných informací o bakalářské práci obsahují i přístup do serverové databáze vytvořené aplikace a je zde uživatelům umožněno nahrávat data. Nakonec byl vytvořen informační poster o bakalářské práci.



Obr. 1 Schéma postupu práce

3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

3.1 Pasport a pasportizace

Pasportizace je velmi široký pojem a neexistuje tedy jednoznačná definice slov pasport ani pasportizace. Obecně je pojem *pasportizace* používán v mnoha různých významech.

V geoinformatické sféře je pojem pasport chápán jako velmi podrobná datová sada vyjadřující zvolenou tematiku. Tato datová sada zpravidla obsahuje polohová data a dále atributy týkající se jednotlivých entit této sady. Mezi příklady patří pasport komunikací, pasport dopravního značení, pasport zeleně apod.

V případě pasportu dopravního značení bude tedy kromě samotné polohy jednotlivých dopravních značek evidován také typ této dopravní značky (např. „P 4 *Dej přednost v jízdě!*“), a také atributy týkající se například stavu dopravní značky, jejího umístění, výrobce dopravní značky atd.

Tato databáze obsahující polohová data a atributy se nazývá pasport. Proces tvorby pasportů, tedy tvorbu databáze a samotné zaevidování jednotlivých entit do této databáze, můžeme označit termínem pasportizace. Výstupem pasportu je většinou mapa jednotlivých entit a tabulka vedených atributů, popř. i fotografie jednotlivých entit. V dnešní době jsou pasporty tvořeny hlavně digitálně, a proto jsou výstupy objednavateli dodávány zpracovatelem i v této formě.

Pasporty, a tedy i proces pasportizace jsou dnes úzce spjaty s informačními systémy o území, jež jsou využívány obcemi. Jelikož jsou některé pasporty pro obce ze zákona povinné (viz následující podkapitola *Pasportizace v zákoně*), existuje mnoho geografických či geoinformatických firem, které jsou schopny pro obec vybrané pasporty zpracovat a dokážou také obci nabídnout vlastní systém, v němž si lze pasporty prohlížet či s nimi dále pracovat nebo je editovat. V dnešní době je trendem přechod z desktopových aplikací k webovým řešením, kde je nespornou výhodou to, že samotný systém funguje prakticky odkudkoliv, kde je připojení k Internetu a dostupnost dat je tedy okamžitá.

Mezi nesmírné výhody pasportů patří například aktuálnost, obsáhlost a přesnost zpracování. V případě dobře zvolených atributů, které pasport obsahuje, může sloužit i jako dokonalá evidence stavu majetku. Majitel či správce pak snadněji může koordinovat případné opravy. Tento pasport pak lze v případech jeho pečlivého zpracování využít jako podklad pro žádosti o dotace z fondů Evropské unie, jako podklad pro energetický audit apod. Pasport může také posloužit např. technickým pracovníkům, úklidovým službám obce či policii.

Mezi nevýhody pasportizace patří jejich vysoké pořizovací náklady, což souvisí s moderními zásadami zpracování pasportů, a tedy s použitým geodetickým zařízením a softwaru GIS. Malé obce proto moderní pasporty zpravidla nevedou a mají jen nezbytné a povinné údaje v analogové („papírové“) verzi.

Pasportizace v zákoně

Obce mají ze zákona povinnost vést si určité pasporty a evidenci svého majetku. Asi nejdůležitějším zákonem týkajícím se evidence obecního majetku je zákon 563/1991 Sb., konkrétně § 29 a § 30, kde je uvedeno, že účetní jednotky (tedy obce) „zjišťují skutečný stav veškerého majetku“ a provádí inventarizaci majetku a skutečné stavy majetků a závazků zaznamenávají v inventurních soupisech. Tento zákon tedy ukládá obcím povinnost evidovat obecní majetek.

Pasportizace se týká i § 5 vyhlášky Ministerstva dopravy a spojů č. 104/1997 Sb., kterou se provádí zákon o pozemních komunikacích. Zde je uložena správcům pozemních komunikací povinnost vést jejich pasport, přičemž rozsah a způsob vedení pasportu stanoví vlastník této komunikace. Konkrétní parametry jsou stanoveny ve třetím odstavci, kde je uvedeno, že nejmenší rozsah evidence (pasportu) místních komunikací „zahrnuje délku místních komunikací I. až III. třídy v km, počet a celkovou délku mostů na nich v km a objem finančních prostředků vynaložených na jejich výstavbu a zvláště na jejich údržbu“.

Mimo toto minimum již zákon neuvádí žádné další atributy, které by měl pasport komunikací obsahovat. Z tohoto důvodu mohou různé pasporty komunikací kromě zmíněných povinných atributů obsahovat i atributy jiné, například typ povrchu komunikace, stav komunikace apod.

Účelové komunikace, stejně tak jako komunikace nižší než III. třídy nepodléhají speciální evidenci. Obce však musí podle § 63 odst. 1 zákona o ochraně přírody a krajiny (114/1992 Sb.) vést přehled o veřejně přístupných účelových komunikacích, stezkách a pěšinách v obvodu své územní působnosti.

Dále zákony pojednávají například o evidenci technické infrastruktury (tedy například veřejné osvětlení), kde jsou vlastníci této infrastruktury „povinni vést o ní evidenci, která musí obsahovat polohové umístění a ochranu, a v odůvodněných případech, s ohledem na charakter technické infrastruktury, i výškové umístění“. Paragraf 161 zákona 183/2006 Sb., o územním plánování a stavebním řádu dále uvádí, že tyto informace mohou být poskytnuty v digitální podobě.

Jak již bylo poznamenáno, neexistuje jasný a ucelený soupis atributů, které by měly být u pasportů evidovány. Zákony stanovují pouze nezbytné parametry, které vybrané pasporty obsahovat musí. Jaké další parametry budou v pasportu vedeny, záleží pak na domluvě mezi objednavatelem a zpracovatelem pasportu.

3.1.1 Typy pasportů

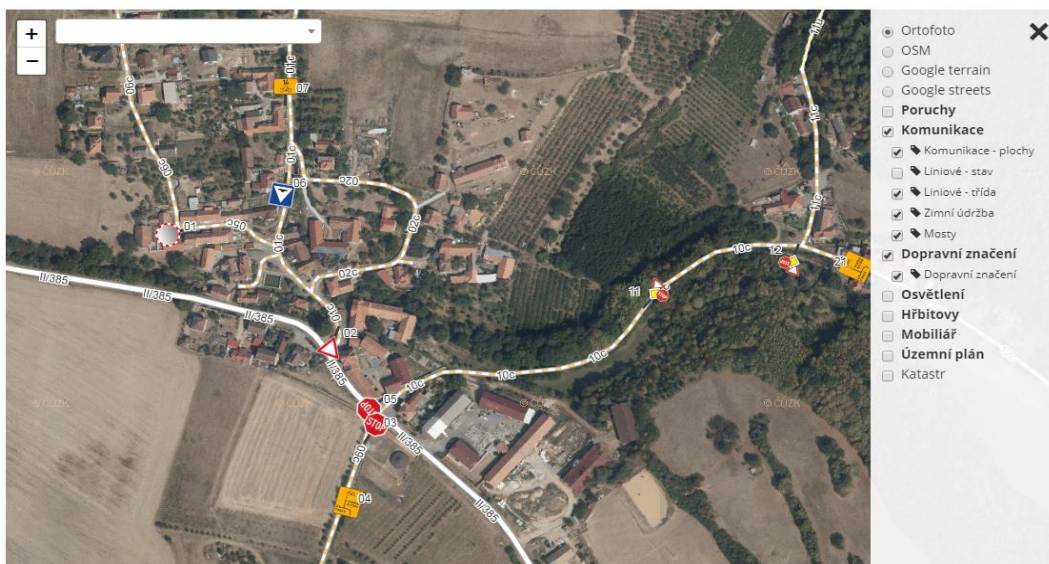
Existuje nespočet typů pasportů. Mezi nejčastější příklady patří pasport místních komunikací, pasport dopravního značení, pasport veřejného osvětlení, pasport hřbitova, pasport zeleně a mnoho dalších. S trochou nadsázky lze říct, že lze zhotovit pasport jakéhokoliv (obecního) majetku.

Pasport místních komunikací

Pasport místních komunikací zahrnuje ze zákona povinně vedenou délku místních komunikací I. až III. třídy v km, počet a celkovou délku mostů na nich v km a objem finančních prostředků vynaložených na jejich výstavbu a zvláště na jejich údržbu. Mimo tyto pasport místních komunikací obsahuje polohová data jednotlivých komunikací v liniové třídě prvků a dále zpravidla typ jejich povrchu, jejich stav, dále pak například informace o stavu krajnice či počet jízdních pruhů. V pasportu místních komunikací bývají vedeny také například účelové komunikace, chodníky a parkoviště (jako polygony).

Pasport dopravního značení

Pasport dopravního značení doplňuje pasport místních komunikací. Sestává většinou z polohových údajů jednotlivých dopravních značení v bodové třídě prvků, dále typu, druhu a popř. velikosti dopravní značky, jejím stavu, jejím umístění (vlastní sloup, budova, sloup veřejného osvětlení, ...) a fotografie této značky. Pasport dopravního značení pak slouží například jako podklad k opravě dopravního značení nebo může být využit policií při řešení dopravních nehod.



Obr. 2 Náhled na webovou aplikaci s pasportem dopravního značení (zdroj: <https://atomicon.cz/>)

Pasport veřejného osvětlení

Veřejné osvětlení spadá do technické infrastruktury obce, a tedy podléhá již zmíněnému § 161 zákona 183/2006 Sb. o územním plánování a stavebním řádu. Pasport veřejného osvětlení je tedy zákonem vyžadován a jeho součástí musí být polohové umístění a ochrana; u pasportu veřejného osvětlení se také uvádí i výškové umístění jednotlivých světelných bodů veřejného osvětlení.

Mimo polohy světelných bodů v bodové třídě prvků je zvykem v pasportu uvádět také typ a druh svítidla, typ výbojky, výkony svítidel, jejich umístění (vlastní stožár, stožár elektrického vedení, budova) apod. V liniové třídě prvků bývají evidovány i kabely (jak nadzemní, tak podzemní) vedoucí od rozvaděčů k jednotlivým světelným bodům, a jako body bývají evidovány také samotné rozvaděče veřejného osvětlení a jejich parametry. Zpravidla bývají jednotlivá svítidla a rozvaděče zfotodokumentovány.

Takový pasport veřejného osvětlení lze využít jako podklad k energetickému auditu a je tedy nezbytný k žádostem o dotace na veřejné osvětlení (např. Státní program na podporu úspor energie – „program EFEKT“, který vyhlašuje Ministerstvo průmyslu a obchodu). Obsahuje-li pasport nepovinná, ale užitečná, výše zmíněná data, např. výkony svítidel, je možné na jeho základě navrhnout energeticky úsporná opatření. Na základě polohových dat zobrazených v mapě lze také navrhnout nová umístění světelných bodů například tam, kde osvětlení není dostatečné.

Pasport hřbitova

Pasportizace hřbitovů je v dnešní době velmi častá. Některé firmy dokážou obcím či správcům hřbitovů nabídnout jednoduchý a přehledný systém, kde lze na mapě například pozorovat, kterému z hrobů na hřbitově končí nájemní smlouva.

U hrobů se dále mimo jejich polohových dat (v polygonové třídě prvků) eviduje jejich stav, typ hrobu (urnový, klasický), kdo je zde pohřbený a také např. fotografie náhrobního kamene.

Pasport zeleně

Provádí se také pasportizace veřejně přístupné obecní či městské zeleně. Jednotlivé stromy či keře jsou v pasportu evidovány jako bodové prvky. Skupiny stromů a travnaté plochy jsou vedeny jako polygonové třídy prvků. K těmto polohovým datům jsou v pasportu dále zaznamenávány další atributy jako např. výška stromu, výměra travnaté plochy apod. Odborné atributy zpravidla stanovuje a vyplňuje dendrolog. Jedná se např. o druh stromu/keře, jejich zdravotní stav aj.

Na základě pasportu zeleně je možné provádět další dendrologické průzkumy nebo navrhovat plány údržby stromů, keřů a zelených ploch.

3.1.2 Proces pasportizace

Pasportizace je náročný a komplexní proces, během něhož je potřeba uskutečnit několik důležitých dílčích kroků, bez nichž by samotný finální pasport nevznikl. Samotnou práci a tvorbu pasportu je v první řadě potřeba důkladně promyslet a je nutné být obeznámen se skutečnostmi, s nimiž se bude pracovat. Proto je nesmírně důležitá komunikace mezi zadavatelem a zpracovatelem pasportu.

Analýza

Prvním krokem tvorby pasportu bývá schůzka zpracovatele se zadavatelem – většinou obcí či majitelem nebo správcem jednotlivých entit (sloupů veřejného osvětlení, stromů, komunikací apod.). Zpracovatel se na schůzku zpravidla připravuje tak, že se dopředu seznámí s prostředím, kde bude mapovat a evidovat požadované entity. Zpracovatel si nastuduje mapu daného území, dále si např. prohlédne dostupné pohledy v Google Street View či Panorama na Mapy.cz a na základě tohoto si zaznamená možné problémy, které by mohly při jeho práci nastat.

Na samotné schůzce zadavatel budoucímu zpracovateli sdělí základní údaje – tedy co se bude evidovat a kolik toho bude. Dále přednese jeho vizi, jaké atributy by v pasportu měly být uvedeny, co bude chtít navíc apod. Zpracovatel se zadavatelem projedná možné problémy, které si předtím zaznamenal (např. „Silnice vypadá jako obecní, ale patří státu – bude v pasportu?“) a tyto spolu vyřeší. Je nutné poznamenat, že dvě protistrany spolu musí vyřešit i finanční stránku věci – kolik to bude stát? Pokud se na všem shodnou, zpracovatel (odborník/geoinformatik) se může dát do práce.

Sběr dat

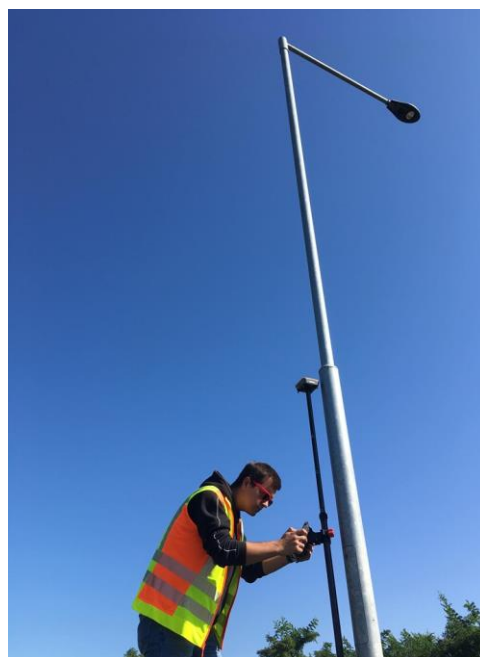
Snad nejdůležitějším krokem pasportizace je práce odborníka v terénu. Sběr dat je časově a technicky náročný. Pro přesný sběr geografických dat v terénu potřebuje odborník-geoinformatik odborná technická zařízení jako např. přesný GNSS/GPS přijímač s vhodným přenosným počítačem (tabletem), kolečkový hodometr na měření vzdálenosti (např. šířky komunikací), laserové měřiče aj. Výběr těchto zařízení záleží na zpracovávaném tématu a typu evidovaných entit. V neposlední řadě si musí odborník dávat pozor také na technické aspekty pasportizace (viz kapitola 5).



Obr. 3 Kolečkový hodometr

Jako příklad bude uveden sběr dat pro pasport veřejného osvětlení. Zde je důležité disponovat kvalitním GNSS/GPS přijímačem s držákem na tablet a také samotným tabletem s nainstalovanou mapovou aplikací vhodnou k pasportizaci (viz podkapitola 3.4).

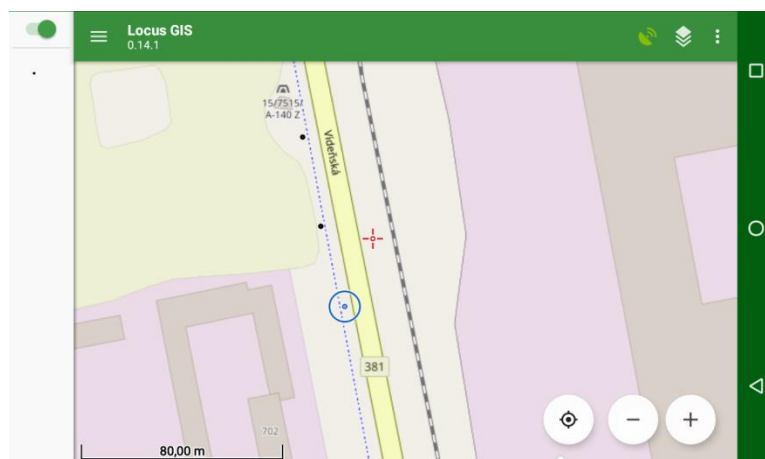
Zaměření bodové vrstvy sloupů veřejného osvětlení probíhá na tabletu spárovaném s GNSS přijímačem ve vhodné aplikaci. Výsledkem je bodová vrstva. Dále je potřeba do vrstvy doplnit atributy týkající se druhu sloupu, druhu svítidla apod., které odborník na tuto tematiku zjistí pozorováním a zapíše do databáze. Přibližnou výšku světelného bodu lze zjistit laserovým měřidlem. Sloup i svítidlo bývají poté i vyfoceny.



Obr. 4 a 5 Průběh zaměřování sloupu veřejného osvětlení

V neposlední řadě je také nezbytné zmapovat a zaevidovat do pasportu rozvaděče veřejného osvětlení, kde zaměření bodové vrstvy probíhá stejně jako u jednotlivých sloupů se svítidly. Fotodokumentace je běžná i u rozvaděčů veřejného osvětlení.

Nakonec je důležité zanést do pasportu kabely vedoucí od rozvaděčů k svítidlům a od svítidel k svítidlům dalším. Nadzemní kabely, vedoucí od sloupu ke sloupu, lze zaznamenat poměrně snadno, ovšem problémem jsou kabely podzemní. Zde je potřeba zkontaktovat zadavatele pasportu či samotného zhotovitele stavby veřejného osvětlení a v ideálním případě od něj získat geodeticky zaměřená liniová data podzemního kabelu. U kabelů bývají vedeny atributy týkající se typu, druhu a materiálu.

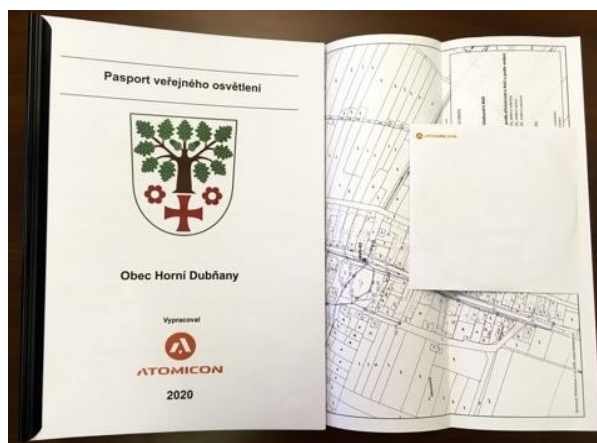


Obr. 6 – Screenshot z aplikace Locus GIS při zaměřování sloupů veřejného osvětlení

Po nasbírání všech těchto dat a jejich zaznamenání do databáze je možné pokročit k poslednímu kroku a tím je zpracování dat.

Zpracování dat

Zpracování nasbíraných a naměřených dat a vyhotovení finálního pasportu je poté zpravidla dokončeno v kanceláři na stolním PC. Z databáze je vytvořen textový pasport s tabulkami, fotografiemi apod. Z naměřených dat je v GIS softwaru vytvořen mapový výstup – mapa pasportu. Tento výstup může být i v digitální podobě, a to například ve formě webové mapové aplikace. Samotná data bývají zadavateli poskytnuta ve vhodném výměnném formátu, většinou jako Shapefile.



Obr. 7 Jedna z možných podob finálního pasportu – textová část, mapové výstupy a CD s daty (zdroj: <https://www.facebook.com/atomicon.cz>)

3.2 Mobilní platforma Android



Nejprve je nutné uvést, že mimo platformu Android existuje mnoho dalších platform. Mezi největší a nejpoužívanější systémy patří kromě Androidu také iOS od společnosti Apple, dále pak Windows Phone od společnosti Microsoft a například BlackBerry. Vzhledem k tomu, že téma této bakalářské práce se týká platformy Android, bude zde diskutována primárně tato platforma.

Android je platforma a mobilní operační systém vyvíjený firmou Google pod hlavičkou konsorcia *Open Handset Alliance*. Toto konsorcium tvoří 84 společností zabývajících se technologiemi a mobilními zařízeními a jejich společným cílem je nabídnout spotřebitelům bohatší, levnější a lepší zážitek s mobilními zařízeními. Z tohoto důvodu vyvinuli Android – první kompletní, otevřenou a bezplatnou platformu pro mobilní zařízení.

Systém Android je založený na Linuxovém jádře, je dostupný jako open-source a mimo smartphony a tablety jej v dnešní době používají například chytré televize, chytré ledničky, autonavigace apod. Ačkoliv je operační systém Android postaven na Linuxovém jádře a byl naprogramován převážně v jazyce C, vývojáři mohou pro vývoj aplikací pro tento systém používat několik různých programovacích jazyků. Nejpoužívanějším z nich je Java (viz podkapitola 3.2.1). Vedle něj začal Google v roce 2017 prosazovat programovací jazyk Kotlin. Dále je možné aplikace doplnit kódem z jazyků C/C++. V neposlední řadě lze do nativní Android aplikace zakomponovat dokonce i prvky webové.

Co se týče statistik, tak dle údajů z roku 2019 je Android nejpoužívanějším mobilním operačním systémem na světě se 76,67 %. Následuje iOS od Applu s 22,09 %. Ostatní operační systémy měly v tomto roce velmi zanedbatelný podíl na trhu. Z tohoto důvodu jsou nejnovější aplikace vyvíjeny zejména pro zařízení s Androidem, či systémem iOS.

3.2.1 Java

Java je objektově orientovaný programovací jazyk vyvinutý firmou Sun Microsystems roku 1995. Jde o jeden z nejpoužívanějších programovacích jazyků na světě.

Společnosti Google a Android zvolily použití Javy jako základ pro aplikace mobilního operačního systému Android. SDK Androidu tak používá jazyk Java jako základ pro tvorbu aplikací.



¹ Obr. 8 Logo mobilní platformy Android (zdroj: <https://www.android.com/>)

² Obr. 9 Logo programovacího jazyka Java (zdroj: <https://www.oracle.com/java/>)

3.2.2 Android Studio

Android Studio je vývojové prostředí určené pro programování aplikací pro platformu Android v jazyce Java (a Kotlin). Bylo představeno v roce 2013 a od té doby se prosadilo jako hlavní vývojové prostředí pro vývoj aplikací pro Android.

Má přívětivé uživatelské rozhraní a disponuje mnoha skvělými funkcemi, jako například integrovaným konfigurovatelným emulátorem systému Android.



3

3.3 Datové formáty JSON a GeoJSON

3.3.1 JSON

„Formát JSON (JavaScript Object Notation) původně vznikl pro předávání dat mezi serverovou a klientskou částí webové aplikace. Jedná se o podmnožinu jazyka JavaScript, která dovoluje reprezentovat základní datové struktury a umožňuje jejich přímočaré použití v prohlížeči. Postupem času se však JSON stal rozšířeným datovým formátem a knihovny umožňující jeho použití existují pro všechny používané jazyky. (...)“

JSON umožňuje reprezentovat čtyři jednoduché datové typy (řetězec, číslo, logická hodnota a null) a dva strukturované (objekty a pole). Pole a objekty mohou jako své prvky obsahovat libovolný další datový typ – jednoduchý i strukturovaný. JSON je tak velmi flexibilní a lze v něm reprezentovat v podstatě jakoukoliv datovou strukturu.“ (Holubová et al., 2015)

3.3.2 GeoJSON

GeoJSON je výměnný geoprostorový formát založený na datovém formátu JSON. Definuje několik typů JSON objektů a způsob jejich kombinace, kterým reprezentují geografické prvky, jejich vlastnosti a prostorový rozsah. GeoJSON využívá souřadnicový systém WGS84 a jako jednotky desetinné stupně (RFC 7946 - The GeoJSON Format, 2016). Souřadnicový systém nelze definovat, a proto se předpokládá, že se jedná o WGS84 (World Geodetic System 1984) (GISMentors, 2017).

GeoJSON se skládá z následujících částí:

- Objekt *Geometry*: Jedná se buď o bod, linii nebo polygon. Obsahuje informace o poloze.
- Objekt *Feature*: Sestává z objektu *Geometry* a jemu přiřazených libovolných dat – vlastností (*properties*).
- *FeatureCollection*: Je to pole prvků obsahující objekty *Feature*.

To znamená, že typický soubor či datová sada formátu GeoJSON se skládá z *FeatureCollection*, jež obsahuje jednotlivé prvky s jejich přiřazenými daty.

³ Obr. 10 Ikona prostředí Android Studio (zdroj: <https://developer.android.com/studio>)

Ukázka kódu GeoJSON

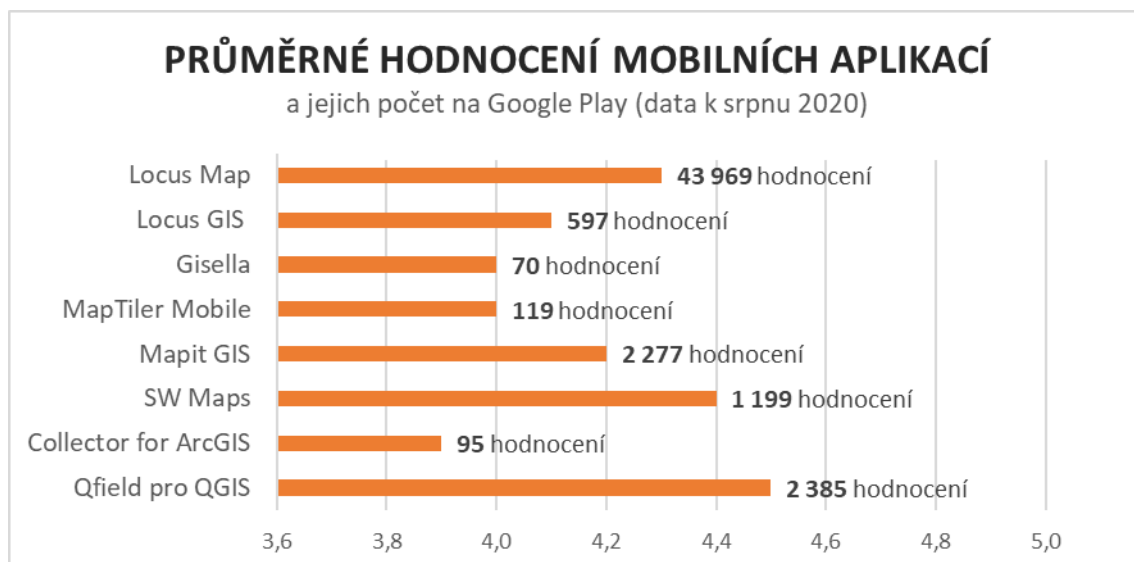
```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [ -90.0715, 29.9510 ]
      },
      "properties": {
        "jmeno": "Adam",
        "pohlavi": "muz"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [ -92.7298, 30.7373 ]
      },
      "properties": {
        "jmeno": "Eva",
        "pohlavi": "zena"
      }
    }
  ]
}
```

3.4 Existující mobilní mapové aplikace

V následující části budou popsány vybrané existující mobilní mapové aplikace pro systém Android. Na základě doporučení známých z oboru geoinformatiky, vyhledávání na Google Play podle klíčových slov „GIS, mapping, maps, map“ apod. a na základě příspěvků z internetových fór a diskusí bylo vybráno osm aplikací.

Jde o *Locus Map* a *Locus GIS*, což jsou aplikace vyvinuté českými vývojáři, dalším českým zástupcem byla aplikace *Gisella*. Vývojáři známí tvorbou MapTileru vydali v roce 2019 aplikaci *MapTiler Mobile*. Dále byla testována anglická *Mapit GIS* a aplikace od společnosti Esri – *Collector for ArcGIS*. Byla otestována také freeware aplikace *SW Maps* od nepálských vývojářů. Všechny doposud vyjmenované aplikace mají uzavřený kód. Co se týče svobodného softwaru, existuje aplikace *QField pro QGIS*, která má otevřenou GNU Public licenci a nesměla tedy ve výběru chybět. Všechny aplikace byly staženy na Google Play a testovány na smartphonu Huawei P20 Pro.

Před testováním byl vytvořen graf z průměrného hodnocení uživatelů na Google Play (v srpnu 2020). Dle těchto hodnocení z výše uvedených aplikací vychází nejlépe *QField pro QGIS* se 4,5 body z 5. Nejhůře naopak *Collector for ArcGIS* od Esri.



Obr. 11 Graf průměrného hodnocení (a jejich počtu) u vybraných mobilních aplikací

Samotné testování vybraných aplikací obsahující jejich charakteristiku – základní parametry, výpis podporovaných formátů a obecné informace – a jejich klady a zápory, vše přehledně sepsané, se nachází jako vázaná příloha č. 1 – *Test existujících mobilních mapových aplikací* na konci této bakalářské práce.

Co se výsledků testování a subjektivního názoru autora týče, tak autor zastává podobný názor jako recenzenti na portále obchodu Google Play a mezi nejlepší mobilní mapové aplikace řadí open-source aplikaci *Qfield pro QGIS*; dále autor práce oceňuje český *Locus GIS*. Produkt firmy Esri *Collector for ArcGIS* vyžaduje velké vylepšení a mnoho oprav, a proto jej autor práce v testování řadí mezi ty horší. Uživateli nejvíce hodnocenou (a v tomto případě asi i nejpoužívanější) aplikací je *Locus Map*, která sice nabízí mnoho funkcí, ale pro tematiku pasportizace je podle autora nedostatečná.

3.5 Webová řešení

Mimo nativních Androidových aplikací existují i webová řešení. Tyto bývají zpravidla napsány v různých programovacích, značkovacích a skriptovacích jazycích. Mezi nejpoužívanější jazyky, bez nichž se dnes skoro žádný web či webová aplikace neobejde, patří *HTML* a *CSS*, jazyk *PHP* a *JavaScript*.

3.5.1 HTML a CSS

HTML, neboli *HyperText Markup Language*, je značkovací jazyk pro hypertext, tedy pro strukturovaný, nelineární text obsahující odkazy (hyperlinky). Nejznámější systém propojující hypertextové dokumenty v rámci Internetu je World Wide Web.

Vývoj *HTML* byl ovlivněn vývojem webových prohlížečů. Vznikl v roce 1990 a současná stabilní verze je 5.2, přičemž standardem je dnes podle W3C (2019) verze *HTML5*.

Jazyk *HTML* je jazyk značkovací, protože obsahuje a využívá značek, mezi které se uzavírají části textu dokumentu. Značky jsou obvykle párové a mohou obsahovat vlastnosti, které funkci značky dále doplňují. Přípona souborů *HTML* je zpravidla *.html*.



¹

¹ Obr. 12 Logo *HTML5* (zdroj: <https://www.w3.org/>)

CSS (Cascading Style Sheets)

CSS je zkratkou pro *Cascading Style Sheets*. Jedná se o jazyk definující vzhled jednotlivých elementů na webové stránce. Umožňuje nastavit takzvaný styl (druh a velikost písma, odsazení, zarovnání textu, barvu apod.) a to buď přímo v textu souboru HTML anebo nepřímo, pomocí tzv. stylesheetu, souboru, který má příponu `.css` a je s HTML souborem propojen.

3.5.2 PHP

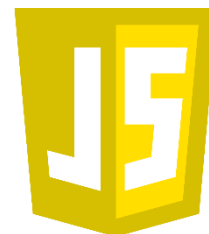
PHP: Hypertext Preprocessor je skriptovací programovací jazyk, který je určený pro programování dynamických webových stránek a webových aplikací. Lze použít také pro tvorbu desktopových aplikací.



PHP vznikl v roce 1994 a podle W3Techs (2020) je nejrozšířenějším skriptovacím jazykem pro web. PHP vychází se syntaxe jazyka C a běží na straně serveru.

3.5.3 JavaScript

JavaScript je objektově orientovaný skriptovací jazyk, který vznikl v roce 1995. Spolu s HTML a CSS je jednou ze základních částí dynamických webových aplikací. Na rozdíl od PHP JavaScript běží na straně uživatele/klienta.



V JavaScriptu je naprogramováno nespočet populárních knihoven a frameworků, jako například *jQuery*, *React.js*, *Node.js*, *AngularJS*, nebo také hojně využívané mapové knihovny jako *Leaflet*, *OpenLayers* nebo *Cesium*.

Leaflet

Mapovými knihovnami se zabývali Néték a Burian (2018) v publikaci *Free and open source v geoinformaticce* a *Leaflet* zde popsali jako „široce užívanou open source JavaScript knihovnu určenou k tvorbě webových mapových aplikací“. Dále uvedli, že „Leaflet může být rozšířen o velké množství pluginů, a že knihovna podporuje většinu mobilních a desktopových platforem podporujících formáty HTML5 a CSS3.“ (s. 65, 66)

„Její rozhraní API je velmi dobře zdokumentované a má jednoduchý a čitelný zdrojový kód“. Zde je ukázka kódu s podkladovou vrstvou OSM a bodovým prvkem:

```
var map = L.map('map').setView([51.505, -0.09], 13);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">
  OpenStreetMap</a> contributors'
}).addTo(map);

L.marker([51.5, -0.09]).addTo(map)
  .bindPopup('A pretty CSS3 popup.<br> Easily customizable.')
  .openPopup();
```

² Obr. 13 Logo jazyka PHP (zdroj: <https://www.php.net/>)

³ Obr. 14 Neoficiální logo jazyka JavaScript (zdroj: <https://seeklogo.com/>)

3.6 Vývojové postupy a metody vývoje softwaru pro mobilní zařízení

Vývoj mobilních aplikací vyžaduje mnoho času, energie a know-how. S pokrokem technologií jsou kladeny stále vyšší a vyšší nároky na programátory a designéry, kteří aplikace vytváří, zejména ve zmíněném know-how, tedy v přístupu, jak aplikace efektivně tvořit. Z tohoto důvodu byly definovány určité vývojové postupy a metody vývoje softwaru pro mobilní zařízení, jimiž se následující část této práce bude zabývat, a budou zde stručně popsány a specifikovány.

3.6.1 Vývojové postupy při tvorbě mobilních aplikací

Jedním z prvotních kroků při návrhu a tvorbě mobilní aplikace je určit, jaký vývojový postup bude použit. Dnes existuje více přístupů a některé mohou být lepší než jiné. Záleží jak na požadavcích tvůrce, tak cílového uživatele aplikace.

Vývojovými postupy při tvorbě mobilních aplikací se zabýval Aversente (2019), kdy vývojové postupy rozdělil na *nativní aplikaci*, *hybridní aplikaci* a *progresivní webovou aplikaci* (PWA). Václavík (2015) místo PWA uvádí *mobilní weby*, jelikož byl termín PWA definován až trochu později v tomtéž roce Russelem (2015) ze společnosti Google.

Nativní

Nativní vývoj je dnes standardní postup při vývoji aplikací pro mobilní zařízení. Pokud tvůrce aplikace jde především o optimalizaci a výkon aplikace, měl by zvolit tento postup. Jeho další výhodou je, že nativní aplikace je pak vzhledově přizpůsobená operačnímu systému, na kterém běží. Obsahuje jeho UI (User Interface) prvky – například v aplikaci pro Android bývají tlačítka nebo lišty v aplikaci stejné, jako kdekoliv jinde v telefonu. Totéž platí u nativních aplikací tvořených pro iOS nebo Windows Phone. Všechna funkcionalita je u nativní aplikace zaručena sadou SDK (Software development kit) dané platformy.

Nevýhodou při tvorbě nativní aplikace pro více různých platform je rozhodně fakt, že musí být použito rozdílných SDK, tedy bude-li tvůrce mít za úkol vytvořit nativní aplikaci jak pro systém Android, tak pro systém iOS, bude muset vytvořit ve skutečnosti aplikace dvě. Každou v jiném vývojovém prostředí a naprogramovanou v jiném programovacím jazyce. Podobně musí dávat tvůrce pozor i při optimalizaci aplikace pro zařízení s různým hardwarem a verzí operačního systému, tedy pro rozdílně výkonná zařízení. Zde se bude potýkat se zdlouhavým testováním s cílem toho, aby aplikace na každém zařízení běžela stejně. Od toho se odvíjí vyšší finanční, časové a personální náklady.

Tyto dva zmíněné mínusy tvorby nativní aplikace jsou jedny z mnoha, ovšem jedná se o ty nejnámější. Chce-li se jim tvůrce vyhnout, je doporučeno vytvořit aplikaci méně náročným přístupem zahrnující webové technologie – tedy hybridním anebo PWA postupem.

Hybridní

Tvorba aplikace hybridním postupem nabízí určité pohodlí, protože je zpravidla založen na webových technologiích. Odpadá tak nutnost tvořit dvě a více aplikací úplně „od nuly“, chceme-li tvořit aplikaci pro více operačních systémů.

Aplikace, tedy spíše její část, je založena na webových technologiích: HTML5, CSS, JavaScriptu a například PHP. Takto vytvořený „web“ je „zabaleno“ a ukryto tak, že se

nainstaluje a chová se jako klasická aplikace. Tento „web“ tedy není otevírán v internetovém prohlížeči, ale používá se zde většinou tzv. WebView. Allen (2013) uvedl, že WebView je webový prohlížeč vestavěný jako widget. Tento prohlížeč umí vykreslovat kód jazyka HTML a lze jej používat k plnohodnotnému procházení Internetu. Podle vývojové příručky Android Developers (2020) WebView umožňuje zobrazit webovou stránku jako součást samotné aplikace.

Další výhodou je, že během testování aplikace stačí otestovat vytvořený web ve webovém prohlížeči a není nutné používat fyzické zařízení nebo emulátor. Tvůrce tímto přístupem oproti nativnímu šetří čas i peníze. Hybridní aplikace stejně jako nativní mohou využívat všech senzorů a funkcí, které zařízení nabízí – například fotoaparátu, geolokace pomocí GPS, technologií Bluetooth a NFC; umožňují posílat uživateli notifikace a upozornění atd.

I tento přístup má ale své nevýhody. U hybridních aplikací je výkon horší než u aplikací nativních. Nativní aplikace jsou tvořeny na míru pro daný operační systém, ovšem toto zde neplatí. WebView představuje pro slabší zařízení vysokou zátěž. Je proto vhodné uvádět u aplikací hardwarové požadavky. Hybridní aplikace rovněž postrádá UI prvky, tedy vzhled tlačítek a ikon, typické pro daný systém, protože tyto byly naprogramovány pro oba systémy zároveň. U Androidu i u iOS tak bude aplikace vypadat stejně, ale uživateli zkrátka „nebude ladit se zbytkem systému“. Další nevýhodou může (ale nemusí) být nutnost připojení k Internetu, používá-li aplikace serverová řešení.

Dnes existuje také velké množství frameworků pro tvorbu hybridních multiplatformních aplikací. Na jedné straně stojí například React Native a Flutter, které se, co se UI týče, blíží spíše nativní aplikaci. Na straně druhé je k dispozici Ionic Framework, jenž používá technologii Apache Cordova. Tyto aplikace bývají zpravidla jednodušší a blíží se PWA.

PWA – Progresivní Webová Aplikace

PWA, celým názvem Progresivní Webová Aplikace (v angličtině Progressive Web Application) není v jádru nic jiného nežli responzivní webová stránka vytvořená primárně pro mobilní zařízení. Slovo „aplikace“ ovšem napovídá, že jde o trochu něco víc. Progresivní webová aplikace používá přístup k funkcím zařízení, které bývají dostupné jen pro nativní aplikace. I u progresivní webové aplikace je tedy možné využívat informací ze senzorů zařízení nebo posílat uživateli notifikace, jak bylo zmíněno u přístupu hybridního a nativního výše. Dokonce je možné odkaz na PWA aplikaci umístit na hlavní plochu zařízení. Navozuje to pocit plnohodnotné aplikace.

Co se týče výhod, progresivní webové aplikace jsou, podobně jako aplikace hybridní, méně nákladné jak finančně, tak časově – zjednodušeně řečeno tvůrci stačí pouze vytvořit responzivní web.

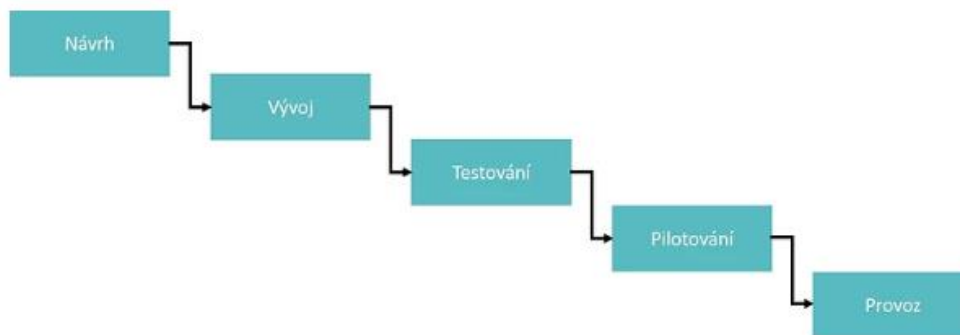
Nevýhody jsou také podobné jako u hybridních aplikací: chybí zde pocit, který má uživatel u nativní aplikace – UI je zkrátka pro každý systém jedinečný a ani hybridní aplikace, ani PWA ho nedokáže plnohodnotně replikovat. Další nevýhodou, podobně jako u aplikace hybridní, je výkon aplikace – webové technologie zatím nenabízejí takové možnosti, jako aplikace nativní.

3.6.2 Metody vývoje softwaru

Při samotné tvorbě softwaru se uplatňují různé metody (nebo metodiky) vývoje. Mezi nejznámější patří vodopádový model a agilní model vývoje softwaru. Obě metody mají svá pro i proti, a i když se jedná pouze o teoretické modely a dnes je v čisté formě určitě nenajdeme, obě tyto metody jsou v modifikovaných verzích široce využívány.

Vodopádový model

Vodopádový, v angličtině Waterfall, je podle Šimůnka (2019) „systém řízení projektů, který rozděluje projekt na jednotlivé „výrobní“ fáze, které se snaží striktně oddělovat. Druhá fáze nezačne, dokud první fáze neskončila apod. Typické fáze Waterfall projektů jsou *analýza, návrh řešení, vývoj řešení, testování, pilotování a provoz.*“ V závislosti na typu projektu a metodiky mohou fáze nabývat i jiných podob. Fáze se zpravidla neopakují a návrh se realizuje pro celé řešení najednou. Na začátku jsou nastavena striktní pravidla, jež pokrývají všechny oblasti projektu a kterých se při vývoji všichni drží.



Obr. 15 Schéma vodopádového modelu (autor: David Šimůnek)

Zde jsou uvedeny výhody a nevýhody vodopádové metody podle Šimůnka (2019):

Výhody

- Lépe se plánuje, řídí a kontroluje
- Dle některých zapadá lépe do našeho kulturního prostředí – je pro nás přirozenější
- Umožňuje jasně určit zodpovědnosti
- Jste schopni doručit projekt i s méně kvalitním týmem

Nevýhody

- Jde relativně o neefektivní způsob řešení
- Je statisticky méně úspěšný
- Neposkytuje dostatek zpětné vazby od zákazníka

Agilní model

Agilní metoda (v angličtině Agile) je modernější přístup vývoje softwaru. Snaží se reagovat na slabé stránky vodopádového modelu. Jeho princip spočívá v krátkých opakovaných (iterovaných) cyklech, které se nazývají sprinty. Během sprintů se u vybraných požadavků navrhne řešení, které se následně vytvoří a proběhne jeho testování. Tím se docílí neustálého zlepšování, jelikož se tvůrce vždy poučí z chyb, které byly objeveny v předchozí iteraci a opraví je.



Obr. 16 Schéma agilního modelu (autor: David Šimůnek)

Zde jsou uvedeny výhody a nevýhody použití agilní metody podle Šimůnka (2019):

Výhody

- Vyšší úspěšnost projektů
- Více zpětné vazby od zákazníka
- Větší zaměření na spokojenost týmu
- Podporuje neustálé zlepšování

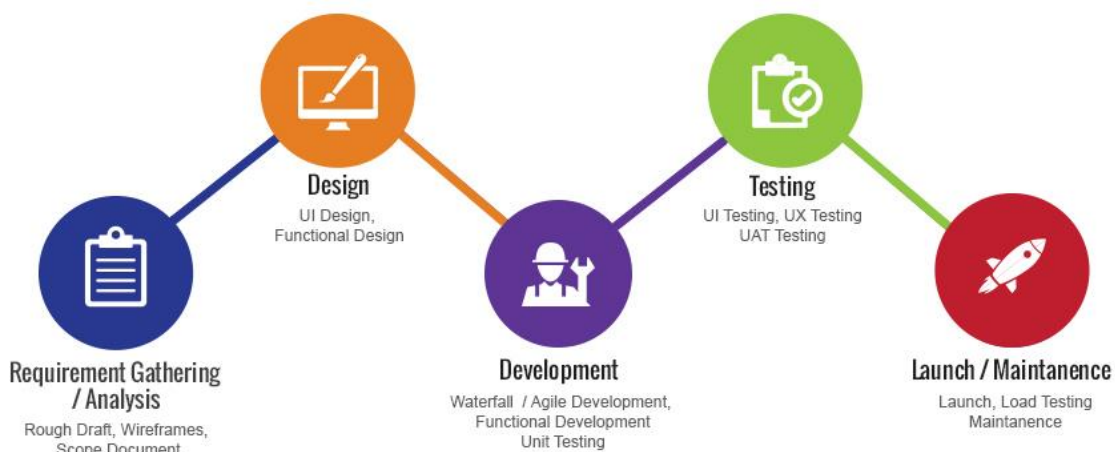
Nevýhody

- Počítá s předpoklady, které nejsou vždy reálné (seniorní tým, agilní okolí – tým, přistoupení top managementu na agilní způsob řízení, ...)
- Je obtížné Agile úspěšně školovat a řídit skrze něj velké celky
- Agile dobře pokrývá práci v týmu, ale selhává ve spolupráci mezi týmy
- Dle některých obtížně zapadá do naší mentality a je pro nás nepřírodní

3.6.3 Životní cyklus

Životní cyklus vývoje mobilního softwaru se podle článku na webu Microsoft Docs (2016) nijak zvlášť neliší od webového nebo desktopového softwaru a má obvykle pět hlavních částí:

1. *Počátek*: Všechny aplikace začínají nápadem. Tento nápad je poté převeden na návrh aplikace.
2. *Návrh*: Fáze návrhu se skládá z definice user experience (UX), jako je například hlavní rozložení aplikace, jak funguje atd., a také z transformace UX do správného návrhu a designu uživatelského rozhraní (UI – user interface), obvykle s využitím pomoci grafického návrháře.
3. *Vývoj*: Vývoj je většinou nejnáročnější a nejnákladnější část životního cyklu. Zde probíhá samotné programování a tvorba aplikace.
4. *Stabilizace*: Pokud je vývoj v pokročilejší fázi, do procesu se začleňuje fáze QA (anglicky: quality assurance; česky: zajištění jakosti), kde probíhá testování a opravy chyb. Často dochází k tomu, že aplikace přejde do omezené beta verze, kterou má možnost širší skupina uživatelů použít, otestovat a poskytnout zpětnou vazbu tvůrci.
5. *Vypuštění aplikace*



Obr. 17 Jedno z možných schémat životního cyklu vývoje mobilního softwaru (anglicky)
(zdroj: <https://www.infoicontechnologies.com/mobile-application-development>)

Softwarové prototypování

Softwarové prototypování (anglicky Software prototyping) souvisí s životním cyklem aplikace. Podobně jako v jiných oblastech života mimo informatiku se zde jedná o prototyp, jak je obecně známý – prototyp nového vozu, nového výrobku apod. před uvedením do výroby. Podle Nývlt (2012) se prototyp při vývoji softwaru typicky zaměřuje pouze na některé z funkcionalit konečného řešení. Proces tvorby prototypu se nazývá prototypování a jeho testování je nedílnou součástí procesu vývoje softwaru (viz podkapitola 3.6.2). Tento pojem souvisí také s verzováním.

Verzování softwaru

Každé vydání nějakého softwaru je zpravidla číslováno nebo označeno nějakým kódovým či slovním označením. Tomuto se říká verzování softwaru. Podle Chytila (2005) se číslování verzí softwaru skládá většinou ze tří čísel, např.: 1.0.2 nebo 7.5.1. První číslo je hlavní, druhému se říká vedlejší a třetímu se říká číslo revize.

Programy s číslem verze nižším než jedna (např. 0.95) jsou programy, které mohou být plně funkční, ale autor se ještě nerozhodl pro jejich plné vypuštění. Tento program chce autor zpravidla ještě zdokonalit.

Verze 1.0.0 je první vydaná verze programu. Při revizích se verze čísluje 1.0.x atp. Pokud už je na programu velké množství zásahů (revizí), bývá vydána verze 1.1.0. A podobně se verzuje dále. Velmi používané jsou také slovní označení verzí, jako například *alpha*, což znamená verzi vytvořenou během vývoje určenou pro testy; dále je používáno označení *beta*, což je další testovací verze po alpha verzi. Program je stále ve vývoji, ale už je umožněno běžným uživatelům použít jej. Dalším hojně používaným termínem je RC (Release Candidate), což označuje téměř finální verzi – „kandidáta na vydání“.

Dnešní vývojová prostředí již disponují nástroji a pluginy pro automatické verzování. Zároveň také existuje systém správy verzí Git a oblíbený portál Github, který nabízí bezplatný webhosting pro open-source projekty. Touto tematikou se zabývali Nėtek a Burian v publikaci *Free and open source v geoinformaticce* a nebude zde dále vysvětlována.

3.6.4 Shrnutí

Aversente (2019) tvrdí, že neexistuje žádný správný nebo špatný postup, jakou metodu vývoje softwaru použít. Neexistuje tak ani odpověď na otázku, jak ideálně postupovat. Vše podle něj závisí na několika faktorech, které ovlivní tvůrcovo rozhodnutí:

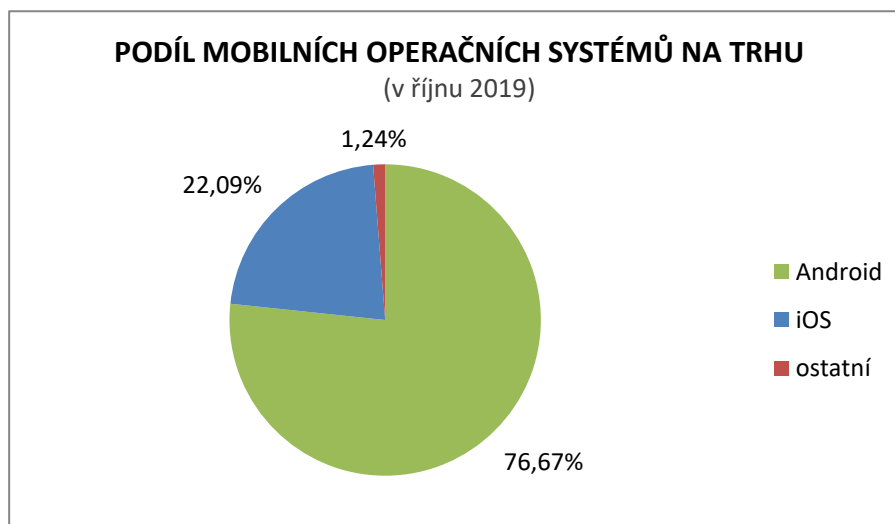
- *Velikost vývojářského týmu:* Tým o jednom člověku bude potřebovat daleko více času, pokud bude tvořit nativní aplikaci pro více platforem, zatímco pro tým dostatečně velký tento postup nebude problémem.
- *Náklady:* Tento bod přímo koreluje s bodem předchozím. Čím početnější vývojářský tým (nebo více týmů), tím vyšší náklady na vývoj aplikace.
- *Zkušenosti:* Pokud má vývojář (nebo vývojářský tým) mnoho zkušeností například v nativním vývoji, bude efektivnější a produktivnější při tvorbě nativních aplikací. Pro někoho, kdo má malé zkušenosti (se třemi metodami vývoje mobilních aplikací výše), bude nejsnazší vývoj progresivní webové aplikace nebo aplikace hybridní.
- *„Time to go-live“:* Tento bod by se dal charakterizovat jako doba, která uběhla od začátku vývoje aplikace do jejího publikování. Zde jsou zkušenosti opět klíčovým faktorem. Čas potřebný k vytvoření PWA by měl být daleko menší, než čas strávený vývojem nativní aplikace a jejím publikováním v obchodech.
- *Osobní preference:* Nejvíce záleží na samotném tvůrci, co upřednostňuje **on sám**.

4 MOTIVACE

Vývoj mobilních aplikací je v současné době velmi lákavá inženýrská disciplína. Mobilní zařízení, ať už se jedná o chytré telefony nebo tablety, má dnes snad každý a množství, které nám tato zařízení nabízí, je nespočet. Porovnáme-li mobilní zařízení s desktopem, tedy například s klasickým stolním počítačem, první obrovskou výhodou oproti počítači, která už je v samotném názvu mobilních zařízení, je *mobilita*. Zeměměřiči nebo zpracovatelé pasportů disponují profesionálními mobilními měřicími zařízeními, která jim ulehčují práci – oproti dřívějším dobám dokážou vytvářet digitální data přímo v terénu. V podkapitole 3.1 bylo uvedeno, že tato hotová data zpracovatelé pasportů poté poskytují zadavatelům.

Chce-li si ovšem zadavatel, například postarší starosta malé obce, tento pasport upravit sám na místě – vymění-li mu elektrikář sloup veřejného osvětlení za nový, proběhne-li rekonstrukce povrchu komunikace z betonu na asfalt nebo mu na obecním hřbitově přibude nový pohřbený – bude to pro něj takřka nemožné. Pro starostu dnes neexistuje intuitivní, jednoduchá (ale taky specializovaná) mobilní aplikace pro tento účel (viz podkapitola 3.4). Starosta bude muset vynaložit finanční prostředky a znovu zkontaktovat zpracovatele pasportu, ať mu pasport upraví. Z tohoto důvodu bylo rozhodnuto v této práci vytvořit jednoduchý, uživatelsky přívětivý prohlížeč geografických dat z pasportizace a zároveň editor jejich atributů.

Samozřejmě existuje více operačních systémů a více možností tvorby aplikací (viz podkapitola 3.6.2), v této práci byl ale vybrán operační systém Android, který je dle údajů z roku 2019 nejpoužívanějším mobilním operačním systémem na světě se 76,67 % podílu trhu.



Obr. 18 Graf podílu mobilních operačních systémů na trhu (zdroj dat: <https://gs.statcounter.com/os-market-share/mobile/worldwide>)

Co se týče vývojových postupů a metod vývoje softwaru, aplikace bude tvořena agilním přístupem, který se autorovi této práce jevil nejlépe. Samotná aplikace bude aplikací hybridní – na jedné straně bude obsahovat nativní části, na straně druhé budou dílčí části využívat technologie WebView. Z pohledu studenta geoinformatiky je toto řešení ideální, jelikož budou zúročeny zkušenosti s tvorbou responzivních webů (konkrétně webových map) a zároveň budou využity poznatky nové, nabyté

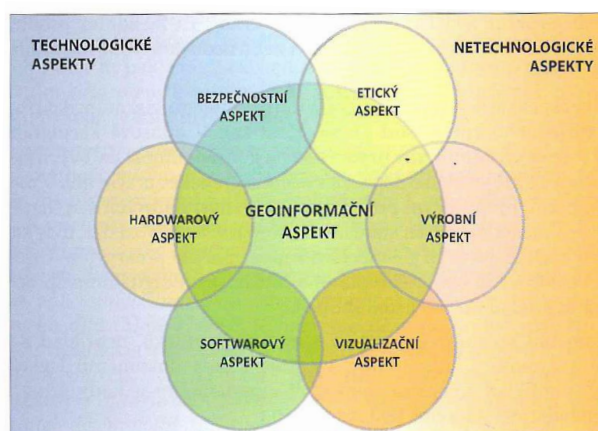
samostudiem jazyka Java a vývojového prostředí Android Studio (viz podkapitoly 3.2.1, 3.2.2 a 3.5). Webová část aplikace bude vystavěna na upravené open-source knihovně Leaflet, jelikož se podle Nětka a Buriana (2018) jedná o jednu z nejpoblárnějšich mapových knihoven jazyka JavaScript. Dalším důležitým bodem jsou data. Aplikace bude pracovat s jednoduchými formáty JSON a GeoJSON (viz podkapitola 3.3). V těchto formátech byla autorovi vedoucím poskytnuta pilotní data pocházející z projektu Inovačního voucheru pro obec Příkazy. Aplikace bude dále využívat GPS senzoru zařízení a přístupu k Internetu.

Proto, aby byla finální mobilní mapová pasportizační aplikace použitelná, musí být kromě výše zmíněných technických aspektů řešeny i geoinformatické a kartografické aspekty.

5 ASPEKTY TVORBY MOBILNÍCH MAPOVÝCH APLIKACÍ

Stejně tak jako při tvorbě klasické papírové mapy je i při tvorbě mapové aplikace důležité definovat a brát ohled na aspekty mapové tvorby. Tyto aspekty se dělí na technologické (resp. technické) a netechnologické. Jednotlivé aspekty v obou kategoriích vymezila Vondráková (2014) ve své publikaci *Netechnologické aspekty mapové tvorby*. Viz obrázek 19.

V následujících částech budou z těchto aspektů vybrány ty, které jsou pro tvorbu mobilní mapové aplikace nejdůležitější, budou popsány a budou aplikovány na proces tvorby mobilních mapových aplikací. Nakonec budou definovány zásady UX/UI mobilních řešení.



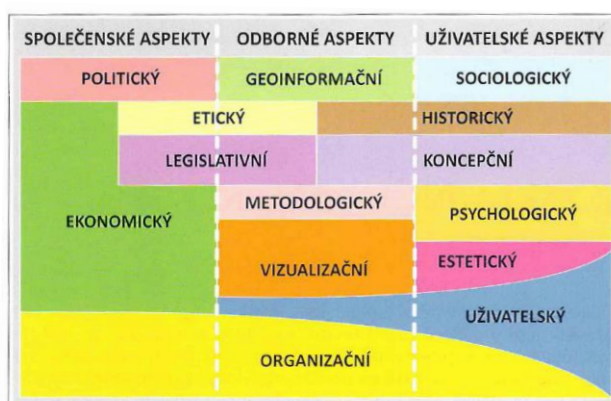
Obr. 19 Prolínání aspektů s aspektem geoinformačním (autor: Vondráková, 2014)

Technologické aspekty podle Vondrákové

Jednotná definice technologických aspektů neexistuje. Tento pojem se v různých variacích objevuje napříč všemi vědními obory. Vondráková (2014) jako technologické aspekty vymezila následující: bezpečnostní, datový, geoinformační, hardwarový, matematický, mapovací, softwarový, standardizační, vizualizační a výrobní aspekt.

Netechnologické aspekty podle Vondrákové

Netechnologických aspektů Vondráková (2014) vymezila skutečně mnoho – viz obrázek 20. Pro potřebu této práce budou vybrány pouze některé z nich.



Obr. 20 Klasifikace netechnologických aspektů mapové tvorby (autor: Vondráková, 2014)

Aspekty podle Nétka

S Vondrákovou spolupracoval Nétka (2015) ve své publikaci *Rich Internet Application pro podporu rozhodovacích procesů Integrovaného záchranného systému*, kde uzpůsobil tyto aspekty tvorbě mapové aplikace určené pro nasazení v oblasti krizového řízení. Viz tabulka 1.

Tab. 1 Tabulka aspektů uzpůsobena mapové aplikaci vytvořené Nétkem (2015)

Skupina technologických aspektů	Skupina netechnologických aspektů
<ul style="list-style-type: none">• Technologické<ul style="list-style-type: none">○ Hardwarové + softwarové○ Standardizační○ Vizualizační (kartografické, geo-informační)	<ul style="list-style-type: none">• Uživatelské<ul style="list-style-type: none">○ Použitelnosti○ Koncepční + organizační○ Vizualizační (psychologické, estetické)
<ul style="list-style-type: none">• Datové	<ul style="list-style-type: none">• Politické, legislativní
<ul style="list-style-type: none">• Bezpečnostní	
<ul style="list-style-type: none">• Participativní	

5.1 Výběr aspektů

Po pečlivém promyšlení, které aspekty důležité pro tuto práci s důrazem na aspekty, jimiž se zabýval Nétka (2015), vybrat, byly z technologických a netechnologických aspektů vybrány následující:

- Hardwarový a softwarový aspekt
- Datový aspekt
- Uživatelský aspekt
- Koncepční aspekt
- Geoinformační aspekt
- Vizualizační a estetický aspekt

5.1.1 Hardwarový a softwarový aspekt

- „**Hardwarový aspekt** je souhrnem vlivů hardwarového vybavení na proces mapové tvorby. Z pohledu tvůrce kartografického díla se jedná především o vybavení pracoviště, dostupnosti potřebných zařízení k mapové tvorbě (...) Z pohledu uživatele se jedná především o nároky na technické vybavení uživatele při práci s mapovým dílem. Klíčovou oblastí hardwarového aspektu je i zajištění integrity jednotlivých technických prostředků včetně zajištění ochrany proti přetížení, zničení nebo poškození.“ (Vondráková 2014, s. 23).
- „**Softwarový aspekt** představuje souhrn programových prostředků používaných pro tvorbu kartografických děl. Zahrnuje geografické informační systémy, grafické programy (...), nástroje pro internetové aplikace a další. Jádrou technologií je programové vybavení.“ (Vondráková 2014, s. 25).

Hardwarový a softwarový aspekt je nutné brát ze dvou pohledů. Jednak z pohledu tvůrců (návrháře, programátora, designéra apod.) mobilní mapové aplikace a také z pohledu uživatelů výsledné mobilní mapové aplikace.

Co se týče hardwarového a softwarového aspektu z pohledu tvůrce aplikace, tak u něj je důležité především disponovat hardwarem, jenž bude bez problému podporovat software nezbytný pro vývoj mobilních aplikací. V případě, že vyvíjí aplikaci pro systém Android, tímto vývojovým prostředím je software Android Studio (viz podkapitola 3.2.2). Hardwarové požadavky Android Studia jsou podle Android Developers (2020) následující:

- *Operační paměť RAM:* minimální 4 GB; doporučené 8 GB
- *Místo na disku:* minimální 2 GB; doporučené 4 GB (500 MB pro vývojové prostředí + 1,5 GB pro Android SDK a systémový obraz pro emulátor)
- *Rozlišení obrazovky:* minimálně 1280 × 800 px

Tyto hardwarové požadavky jsou pro všechny počítačové operační systémy stejné. Co se týče softwarových požadavků, tak jsou podporovány všechny využívané operační systémy: Microsoft Windows® od verze 7, Mac® OS X® 10.10 (Yosemite) a vyšší, až do 10.14 (macOS Mojave), Linux GNOME nebo KDE desktop a GNU c knihovna 2.19 nebo pozdější, posledním podporovaným operačním systémem je ChromeOS. Tyto hardwarové a softwarové požadavky jsou důležité zejména při práci s emulátorem, který je velmi náročný na výkon počítače.

U hardwarového a softwarového aspektu z pohledu uživatele aplikace je řešeno téměř totéž jako u tvůrce, a tedy hardwarové a softwarové vybavení zařízení, na něž je aplikace tvořena. Za toto zodpovídá tvůrce aplikace. Ať už se jedná o obyčejnou textovou aplikaci, nebo o náročnou aplikaci pracující například s 3D grafikou, vývojář by se měl snažit aplikaci tvořit s cílem pokrýt nejširší spektrum hardwaru a softwaru (v případě platformy Android se jedná o verzi operačního systému Android) a aplikaci pak při jejím výkonnostním testování (viz podkapitola 7.1) pro tato všechna zařízení optimalizovat. Čím širší spektrum podporovaných zařízení, tím vyšší počet stažení, uživatelů a v případě zpeněžení aplikace i vyšší finanční zisk. Tvůrce aplikace by poté měl o hardwarových a softwarových požadavcích své aplikace informovat cílovou skupinu uživatelů.

Při tvorbě mobilní mapové aplikace je také důležité vzít v úvahu polohová čidla a služby zařízení. Dnes téměř všechna mobilní zařízení disponují polohovými senzory GNSS/GPS, připojením k mobilní síti a Internetem, tedy prostředky, pomocí nichž lze poloha s velkou přesností určit. Oproti profesionálním zařízením, která se používají například při geodetických zaměřeních nebo pasportizaci (viz podkapitola 3.1), jsou však běžná zařízení méně přesná (metry vs. milimetry).

5.1.2 Datový aspekt

- *„**Datový aspekt** zahrnuje vliv vlastností použitých prostorových vrstev (geodatabází) při tvorbě kartografického díla. Jedná se například o formát dat, jejich kvalitu, návaznost na další datové sady a další. Součástí datového aspektu je i způsob práce s daty, použité datové modely a struktury, včetně souvisejících formálních náležitostí (např. licenční ujednání). Nezbytnou součástí datového aspektu jsou i technologie zajišťující důvěrnost programových prostředků, především pak odolnost proti chybným vstupním datům (např. odolnost proti buffer overflow), zajištění ochrany proti parazitním kódům, proti podvržení identity a další.“* (Vondráková 2014, s. 21).

V případě mobilní mapové aplikace by měl být kladen důraz na to, s jakými daty bude aplikace umět pracovat. V podkapitole 3.4 je u zkoumaných mobilních mapových aplikací uveden výčet podporovaných datových formátů. Zde opět záleží na tvůrci – chce-li dělat aplikaci odbornou a komplexní, umožní podporu většího množství formátů. Chce-li tvořit aplikaci jednoduchou a univerzální, bude podporovat formátů méně nebo umožní uživateli konverzi dat do univerzálního výměnného formátu.

V případě dat, s nimiž pracují GIS je takovým formátem Shapefile. V případě webových produktů je ovšem pro geografická data velmi populární formát GeoJSON (viz podkapitola 3.3.2). Podle Nětka (2015) je GeoJSON formát nezávislý na platformě, který lze číst a upravovat v libovolném programovacím jazyce, podporující sémantickou strukturu dat. Z tohoto pohledu se tento formát s jednoduchou strukturou jeví jako ideální formát pro (mobilní) mapové aplikace. Tomuto nahrává i fakt, že, jak Nětka (2015) uvedl, se na GeoJSON adaptoval i Google, jeho potenciál vycítila firma Esri a také je tento formát podporován celou řadou nezávislých odborníků.

5.1.3 Uživatelský aspekt

- „Uživatelé kartografických děl jsou nejdůležitějším článkem v procesu mapové tvorby. Bez uživatelů by nebylo třeba kartografická díla produkovat a tím by kartografie jako věda zanikla. Proto je třeba na uživatelské potřeby a preference brát zvláštní důraz a měly by být určujícím prvkem pro tvorbu koncepce kartografického díla a postupovat celým procesem mapové tvorby. Všechny tyto faktory zahrnuje **uživatelský aspekt**. Existují obecně hodnocené aspekty kartografických děl, které jsou vztaženy k potřebám uživatelů map (např. optické aspekty), a tyto mohou být v rámci uživatelského testování dále hodnoceny.“ (Vondráková 2014, s. 153).

Uživatelský aspekt je z kategorie aspektů netechnologických. Jak je uvedeno výše, uživatel je na prvním místě, i co se týče tvorby mobilních mapových aplikací. Tvůrce aplikace by měl tvořit aplikaci pro uživatele – musí mít vytyčenou nějakou cílovou skupinu nebo mít objevenou „díru na trhu“. Tohle je podobný případ jako u hardwarového a softwarového aspektu, kde by měl tvůrce pokrýt široké spektrum zařízení, aby aplikaci mohlo využívat co největší počet uživatelů.

Dále je také důležitá uživatelská zpětná vazba. Zde hraje velkou roli uživatelské testování, důležitá část softwarového vývojového procesu. Uživatelé zpravidla beta verzi aplikace otestují a dávají pak podněty tvůrci, který pak na jejich základě upravuje funkce a design (UX a UI prvky) aplikace tak, aby uživatelům vyšel vstříc.

5.1.4 Koncepční aspekt

- „**Koncepční aspekt** kartografických děl se vztahuje především k obsahu map a je zaměřen na způsob prezentování informací a výsledků analýzy a syntézy vstupních informací v mapě. V souvislosti s tvorbou tematických map existují koncepce informační, komunikační, systémová, poznávací, matematicko-kartografické modelování, jazyková a další. Obecnějším pojetím koncepčního aspektu je dělení map podle jejich koncepce na mapy analytické, komplexní a syntetické.“ (Vondráková 2014, s. 96).

Koncepční aspekt také spadá do kategorie netechnologických aspektů. Jak napovídá poslední věta v úvodu, tvůrce mobilní mapové aplikace by měl rozhodnout, zda jeho aplikace bude zobrazovat analytické, komplexní nebo syntetické mapy. Koncepční aspekt zohledňuje především cíl a účel mobilní mapové aplikace. Zde opět záleží na cílové skupině uživatelů (uživatelském aspektu) a „díře na trhu“.

5.1.5 Geoinformační aspekt

- *„**Geoinformační aspekt** zahrnuje vliv použitého hardwaru a softwaru na tvorbu kartografických děl. Aspekt zahrnuje různé nástroje pro zpracování, správu a vizualizaci prostorových dat. Součástí geoinformačního aspektu jsou různé metodiky pro zpracování a vizualizaci prostorových dat, různé formy realizace prostorových analýz a způsoby vyhodnocování dat, včetně možností aplikace vybraných metod kartografického vyjadřování pro tvorbu výsledného mapového díla. Nezbytnou technologií geoinformačního aspektu jsou geografické informační systémy.“* (Vondráková 2014, s. 22).
- *„V netechnologickém pojetí představuje **geoinformační aspekt** především využití znalostí a dovedností geografických informačních systémů – využití různých geoinformačních nástrojů například pro realizaci analýzy a syntézy zpracovávaných dat, které mohou vést k získání dosud neznámých informací, prezentovaných následně prostřednictvím mapové tvorby.“* (Vondráková 2014, s. 81).

Geoinformační aspekt můžeme zkoumat ve dvou kategoriích (viz výše), jak v technologickém, tak netechnologickém pojetí mapové tvorby. U tvorby mobilních mapových aplikací geoinformační aspekt úzce souvisí s hardwarovým a softwarovým aspektem. Při tvorbě mobilní mapové aplikace je potřeba služeb odborníka – geoinformatika (a kartografa), aby bylo docíleno geoinformační a kartografické správnosti map v aplikaci, a také zpracování a zobrazení geografických dat.

Jedině po konzultaci tvůrce s odborníky bude mobilní mapová aplikace správná a nedojde k matení uživatele – laika, například špatně zvoleným kartografickým zobrazením a souřadnicovým systémem.

5.1.6 Vizualizační a estetický aspekt

- *„**Vizualizační aspekt** je z hlediska technologického pojetí souhrnem různých vlivů fyzikálních veličin na tvorbu kartografických vizualizací, a to nejen s ohledem na formu zpracování (digitální nebo tištěná kartografická díla), ale také v souvislosti se smyslovou percepcí map uživateli. Rozhodující jsou technologie využité při tvorbě a prezentaci výsledného kartografického díla.“* (Vondráková 2014, s. 26).
- *„**Vizualizační aspekt** zahrnuje vytváření kompletní mapové kompozice se všemi doplňkovými kompozičními prvky mapy a především úpravou mapového pole. Do vizualizačních aspektů patří výběr použitých metod kartografického vyjadřování, použití znakového klíče, kompozice mapy, její design a vzhled. Do vizualizačního aspektu patří také forma prezentace výsledné mapy.“* (Vondráková 2014, s. 165).

- **„Estetický aspekt kartografických děl má vliv na celkovou kvalitu a tzv. uživatelskou vstřícnost kartografického díla. Zahnuje výtvarné provedení kartografického díla, celkový design, použité metody kartografické vizualizace a další. Z hlediska uživatelské percepce může mít estetika mapy vliv na emoce uživatele a jeho budoucí preference.“** (Vondráková 2014, s. 71).

Vizualizační aspekt spadá do obou kategorií – technologických i netechnologických aspektů mapové tvorby, estetický aspekt je aspektem netechnologickým.

Co se týče vizualizačního aspektu z pohledu tvůrce mobilní mapové aplikace, je nutné nejprve zvolit vhodnou mapovou knihovnu pro zobrazování podkladové mapy a překryvných tematických dat. Zde se nabízí využít buď nativní cestu (podkapitola 3.6.1) a knihovnu nativní nebo cestu hybridní či PWA a knihovnu webovou (JavaScriptovou). Nepoužívanějšími nativními knihovnami jsou Maps SDK for Android od společnosti Google, ArcGIS runtime for Android od Esri anebo také open-sourcová knihovna *OSMDroid*. Dalšími nativními knihovnami jsou například Mapbox, CARTO, MapsForge nebo Scout.

Mezi webové JavaScriptové knihovny patří například open-sourcové knihovny OpenLayers nebo *Leaflet* (viz podkapitola 3.5.3). Webové knihovny jsou v dnešní době velmi populární a například Néték (2015) dokonce vidí budoucnost v tzv. webových terminálech, kdy uživatelská koncová zařízení budou obsahovat jen a pouze webový prohlížeč. Proto se nabízí využít webového nebo hybridního řešení a využít open-source JavaScriptovou knihovnu. V případě velmi zkušeného programátora volba knihovny odpadá, jelikož si toto vše může naprogramovat sám, jedná se ovšem o zdlouhavý a technicky náročný proces.

Po volbě vhodného přístupu k vizualizaci geografických dat v mobilní aplikaci je dále důležité zvolit pro jednotlivé prvky tematických mapových vrstev vhodný znakový klíč a design a barvy tak, aby byly jak v harmonii s podkladovou mapou a zároveň aby nepůsobily rušivě a vhodně vyjadřovaly zobrazovanou tematiku. Vizualizační aspekt úzce souvisí s metodami kartografické vizualizace. Těmito se zabývali Voženílek a Kaňok (2011) v publikaci *Metody tematické kartografie: vizualizace prostorových jevů* a také například Vondráková (2014) v publikaci *Netechnologické aspekty mapové tvorby*. Nebudou zde proto dále rozebírány.

Dále je při návrhu a tvorbě mobilní mapové aplikace důležité zvolit vhodné kompoziční prvky. Na displeji mobilního zařízení by měl uživatel při používání aplikace vidět mimo ovládacích tlačítek a mapového pole také měřítko a legendu. Tiráž a název mapy na displeji mobilní mapové aplikace během používání (podle autora této práce na základě prozkoumaných mapových aplikací v podkapitole 3.4) nejsou potřeba – lze je skrýt například do výsuvných informačních nabídek nebo do informací o aplikaci. Totéž platí i u legendy, ale u ní se nabízí skrytí přímo pod ovládací tlačítko přepínání vrstev.

Nadstavbové kompoziční prvky jsou pro uživatele mapové aplikace také důležité. Tvůrce aplikace může například využít směrovku při pohybu uživatele (v případě sledování jeho polohy). Samozřejmostí jsou také textová pole a informační tabulky, jež se v aplikaci zobrazí například po kliknutí na mapový prvek.

Všechny tyto zmíněné body týkající se vizualizace by měly být tvořeny s důrazem na estetiku, a tedy i na estetický aspekt. Ten má vliv na celkovou kvalitu mobilní mapové aplikace.

5.2 Zásady UX/UI mobilních řešení

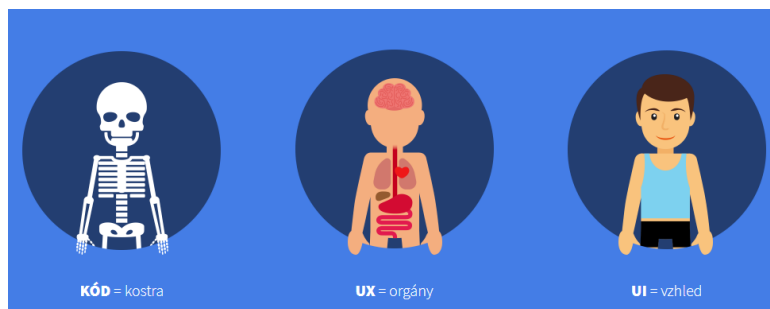
User Experience a User Interface jsou dva moderní termíny, které se používají při tvorbě webových stránek nebo aplikací. Čorič (2017) napsal, že společný cíl UX a UI je produkt, který budou uživatelé používat ke své maximální spokojenosti. Je nutné také zmínit, že v praxi je řešení UX a UI jeden jediný proces a celý se poté nazývá UX/UI design.

5.2.1 Charakteristika

UX je zkratkou pro *User Experience*. Česky by se tento termín dal nejlépe přeložit jako *uživatelský dojem*. UI je zkratkou pro *User Interface*, což se česky překládá jako *uživatelské rozhraní*. Oba tyto prvky jsou důležité pro vytvoření kvalitní mobilní aplikace a v praxi se často překrývají.

Stručně vysvětleno na příkladu mobilní mapové aplikace se jako User Interface dá označit vše, co uživatel vidí na obrazovce – mapové pole, ovládací tlačítka, legenda, barvy, rozložení prvků apod. Na druhou stranu to, co je v pozadí, jak funguje vyhledávací pole, jakým způsobem probíhá editace atributů, jak fungují ovládací tlačítka atd., se označuje jako User Experience.

UX design se tedy snaží poskytnout co nejlepší uživatelskou zkušenost, zatímco UI design dbá na tvorbu graficky přívětivého a efektivního designu aplikace. Článek „*Co je UX/UI design (webů a aplikací)?*“ od agentury bpromotion přirovnává UX a UI k lidskému tělu (viz obrázek 21) a je zde uvedeno, že „*Spojením pevných kostí (kód), zdravých orgánů (UX) a příjemného vzhledu (UI) vznikne funkční, smysluplný a estetický organický stroj (web či aplikace).*“



Obr. 21 Přirovnání UX a UI k lidskému tělu (zdroj: <http://cojeuxui.cz/>)

User Experience

UX je podle Čoriče (2017) způsob, jak promýšlet a plánovat aplikace tak, aby uživatel měl co nejlepší zkušenost z interakce s aplikací.

UX je především o průzkumu trhu a cílové skupiny uživatelů, analýze a plánování. Jde zde spíše o přemýšlení a předcházení problémům, které by mohly během používání aplikace nastat než o samotné designování aplikace.

Konkrétně se jedná o proces tvorby a vylepšování interakce mezi uživatelem aplikace a aplikací samotnou. Během návrhu a tvorby mobilní aplikace se UX designer vždy drží zásad uživatelského a koncepčního aspektu. Všechny prvky aplikace tedy tvůrce přizpůsobuje potřebám uživatele. Uživatel musí ze své interakce s aplikací získat určitou hodnotu.

Konečný výsledek UX procesu je něco, co nejde na první pohled postřehnout. Perfektní UX design je takový, že uživatel aplikace nenarazí na žádné chyby, které by mu bránily v jejím plnohodnotném používání.

User Interface

UI je oproti UX jiný typ procesu. Řeší se v něm design, vzhled a dojem z aplikace. UI proces provádí většinou zkušený grafik. Grafik neboli UI designér typicky řeší:

- Přívětivý a efektivní design aplikace
- Design, který správně nasměruje pozornost uživatele a dovede ho k cíli
- Interaktivní prvky aplikace

Cílem každého UI designéra by měla být atraktivní a uživatelsky použitelná aplikace, v níž by mělo být pomocí designových prvků umožněno uživateli plnohodnotně a bez problému využít veškerých jejich funkcí. Účelem UI designu je podle Čoriče (2017) usnadnit komunikaci mezi uživatelem aplikace a aplikací samotnou.

5.2.2 Srovnání

Podle článku „Co je UX/UI design (webů a aplikací)?“ od agentury bpromotion jsou rozdíly mezi UX a UI následující:

Tab. 2 Rozdíly mezi UX a UI designem

UX (User Experience)	UI (User Interface)
Na první pohled neviditelný	Na první pohled viditelný
Návrh produktu/služby tak, aby splnil uživatelské požadavky	Návrh prvků, celku, animací a interakcí tak, aby uživatele vizuálně provedl a naplnil tak UX
Je zaměřen na uživatele, jejich potřeby a cíle	Je zaměřen na uživatele, jejich potřeby a cíle
Netýká se jen webů a aplikací	Týká se pouze webů a aplikací

UX a UI jsou si v mnoha směrech podobné. Při tvorbě ať už mobilních aplikací, klasických aplikací, webů nebo jiných druhů softwaru je velmi důležité dbát na všechny jejich zásady.

6 PILOTNÍ STUDIE – APLIKACE PASPORTY

Poté, co byla provedena rešerše, byly vysvětleny důležité pojmy, byly prozkoumány různé přístupy a metodiky vývoje softwaru a byly definovány aspekty tvorby mobilních mapových aplikací a zásady UX a UI mobilních řešení, lze konečně přejít k samotné tvorbě mobilního pasportizačního klienta. V následujících podkapitolách této pilotní studie bude navržena a naprogramována aplikace s názvem Pasporty. Tento postup bude také orientačně vysvětlen a popsán.

6.1 Návrh aplikace

Na základě rešerše a autorovy Motivace uvedené v kapitole 4 bylo rozhodnuto vytvořit jednoduchý, uživatelsky přívětivý prohlížeč geografických dat z pasportizace a zároveň editor jejich atributů. Aplikace bude hybridní a při její tvorbě bude použit agilních metod vývoje softwaru. Aplikace bude využívat Internetu a polohových služeb ze zabudovaných senzorů GNSS. Název aplikace bude „**Pasporty**“ a sloganem aplikace bude věta „*Atributy máte ve svých rukou...*“.

Aplikace **Pasporty** se bude skládat ze dvou částí:

- **Online Editor** – jedná se o onlinovou část zabudovanou v nativní aplikaci. Bude zde možné prohlížet a editovat pasporty v rozhraní založeném na open-source knihovně Leaflet. Tato knihovna bude upravena pro potřeby aplikace, webově hostována a s nativní částí aplikaci propojena přes WebView.
- **Offline Prohlížeč** – tato nativní část aplikace bude založena na open-sourcové nativní knihovně OSMDroid. Bude umožňovat uživateli zobrazit datovou vrstvu staženou přes rozhraní Online Editoru a také zde uživatel bude moci zobrazit atributy jednotlivých prvků.

Tyto dvě části budou přístupné v jednom společném nativním rozhraní, jehož součástí bude *úvodní obrazovka*, *hlavní menu aplikace* a pomocné obrazovky, na nichž bude mít uživatel možnost *vybrat pasport ke zobrazení*. Dále bude aplikace obsahovat položku *O Aplikaci*, kde budou informace o použitých mapových zdrojích, knihovnách apod. a také bude v hlavním menu *odkaz na web* k této bakalářské práci.

Co se týče datového aspektu, tak aplikace bude pracovat s jednoduchými formáty dat JSON a GeoJSON. Data pro *Online Editor* budou umístěna na serveru spolu s onlinovou částí aplikace. Na webu k bakalářské práci bude položka *Nahrát data*, kde bude mít uživatel po přihlášení do systému možnost nahrávat do aplikace další data pasportů a také je mazat. Pro testování aplikace budou použita pilotní data pocházející z projektu Inovačního vouchery pro obec Příkazy, která autorovi poskytli vedoucí práce.

V *Online Editoru* **nebude** možné editovat geometrii, ale pouze atributy jednotlivých prvků. Je pro to více důvodů: Cílovou skupinou (viz uživatelské a koncepční aspekty tvorby mobilní mapové aplikace, podkapitoly 5.1.3 a 5.1.4) této aplikace jsou starostové malých obcí a tito podle autorových zkušeností požadují jednoduchost – příklady byly uvedeny v kapitole 4. Další důležitý důvod, který hraje v neprospěch editace geometrie je nízká přesnost zabudovaných GPS senzorů v obyčejných mobilních zařízeních. Oproti profesionálním měřicím přístrojům nemají GNSS senzory v mobilních zařízeních dostatečnou kvalitu, přesnost a korekční data. Z tohoto důvodu nepřesnosti je editace

geometrie zbytečná, jelikož pasport by po editaci geometrie s použitím nepřesných senzorů ztratil hodnotu a smysl. Dalším důvodem, proč aplikace nebude umožňovat editaci geometrie prvků je samotné mobilní dotykové zařízení, na dotykovém displeji nelze spolehlivě zajistit dostatečnou přesnost kliknutí.

Uživatel (starosta malé obce) tak může výslednou aplikaci používat pro prohlížení pasportu v terénu za pomoci vestavěných GNSS senzorů do jeho zařízení, a také pro editaci atributů, nastane-li mu v pasportu nějaká změna. Přesně zaměřený pasport mu bude dodán zpracovatelem (viz podkapitola 3.1) ve vhodném datovém formátu, který si bude moci nahrát do aplikace *Pasporty*.

Poznámky:

Následující podkapitoly jsou srozumitelnější pro čtenáře se zkušenostmi s programováním, zejména s vývojovým prostředím Android Studio a programovacím jazykem Java. Dále jsou pro lepší srozumitelnost doporučeny zkušenosti s tvorbou webů (jazyky HTML, PHP a JavaScript).

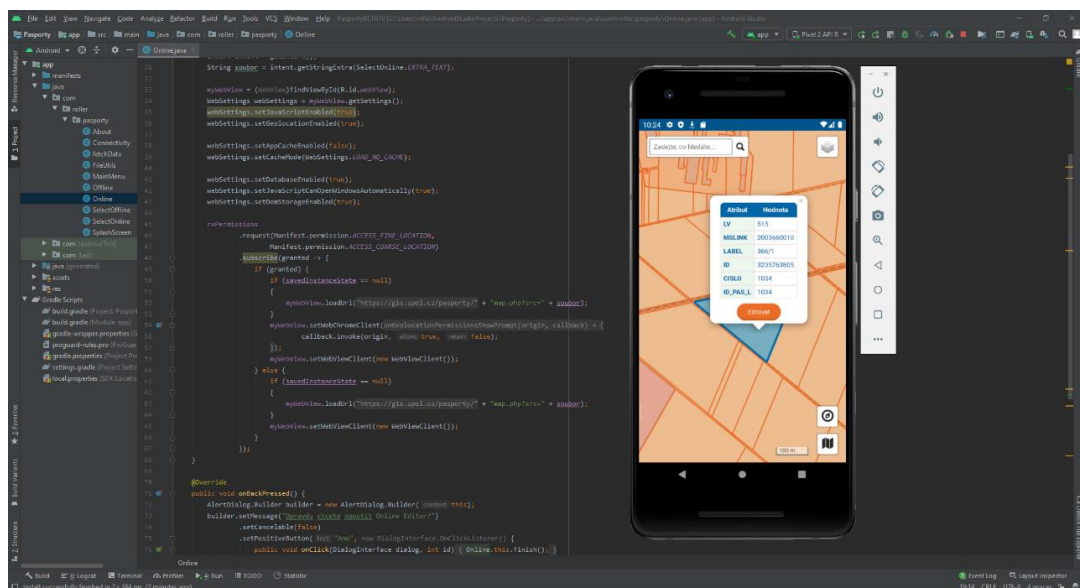
V následujících podkapitolách také nebude uveden a vysvětlen celý programový kód. Budou zde spíše vysvětleny metody a postupy tvorby aplikace. Celé znění programového kódu (místy okomentovaného autorem) je dostupné jako příloha na CD.

Dále je také nutno poznamenat, že autor neměl doposud s tvorbou mobilních aplikací žádnou zkušenost. Při tvorbě vycházel z oficiální dokumentace, uživatelských příruček, videonávodů, internetových fór a z knihy *Android 4: průvodce programováním mobilních aplikací* od Granta (2013). Z tohoto důvodu mohou zkušení programátoři nalézt v postupu určité chyby nebo odchylky od zažitých standardů.

6.2 Vývoj aplikace

Aplikace byla vyvíjena v několika verzích vývojového prostředí Android Studio (viz podkapitola 3.2.2). Bylo využito programovacího jazyka Java a XML. Dílčí části aplikace byly naprogramovány ve skriptovacích jazycích JavaScript a PHP a značkovacím jazyce HTML se styly CSS ve vývojovém prostředí Visual Studio Code. Grafika v aplikaci, loga a ikony byly vytvořeny v programech GIMP a Inkscape.

V průběhu tvorby aplikace bylo vytvořeno celkem 43 testovacích verzí aplikace, která má 3567 řádků, s pomocnými funkcemi a komentáři dokonce 5504 řádků kódu.



Obr. 22 Vývojové prostředí Android Studio s emulátorem, v němž je spuštěná aplikace *Pasporty*

6.2.1 Soubor `AndroidManifest.xml`

Soubor `AndroidManifest.xml` je soubor XML popisující sestavovanou aplikaci a komponenty dodávané touto aplikací (aktivity, služby atd.) (Allen, 2013).

V souboru `AndroidManifest.xml` u aplikace Pasporty byly definovány povolení, (anglicky permissions), které bude aplikace ke správnému fungování potřebovat. Jedná se o povolení přistupovat k lokaci zařízení (povolení pro GPS), dále pak povolení k přistupování k úložišti mobilního zařízení a také povolení týkající se připojení k Internetu. Zde jsou jednotlivé `android.permission` vypsány:

```
ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION,  
ACCESS_BACKGROUND_LOCATION, WRITE_EXTERNAL_STORAGE,  
READ_EXTERNAL_STORAGE, ACCESS_NETWORK_STATE, INTERNET
```

Dále se v souboru `AndroidManifest.xml` nachází definice ikony aplikace, jejího názvu apod., a také výčet všech aktivit v aplikaci, jejich vlastnosti a další nastavení, například nastavení orientace obrazovky.

6.2.2 Soubor `build.gradle`

Soubor `build.gradle` obsahuje definici použitého SDK pro kompilaci, podpis APK (Android Application Package) a také definici použitých knihoven.

Minimální verze SDK byla zvolena (na základě výkonnostního testování) jako 23, což odpovídá verzi Android 6.0 Marshmallow.

V aplikaci Pasporty bylo použito mnoho knihoven, a zde je jejich přehled:

- **Lottie** – open-source knihovna od společnosti Airbnb umožňující animace (zdroj: <https://github.com/airbnb/lottie-android>)
- **Android About Page** – open-source knihovna autora medyo, v níž byla vytvořena část *O Aplikaci* (zdroj: <https://github.com/medyo/android-about-page>)
- **CoordinatorLayout** – kontejner pro potřebu Offline Prohlížeče
- **RxPermissions** – open-sourcová knihovna autora tbruyelle využívána pro správu povolení (využívat například internetové připojení, přistupovat k úložišti nebo polohovým službám) (zdroj: <https://github.com/tbruyelle/RxPermissions>)
- **OSMDroid** – open-source mapová knihovna, na které běží Offline Prohlížeč (zdroj: <https://github.com/osmdroid/osmdroid>)
- **OSMBonusPack** – open-source knihovna autora MKergall rozšiřující knihovnu OSMDroid; knihovna přidává mimo jiné podporu pro formát GeoJSON a je využita v Offline Prohlížeči (zdroj: <https://github.com/MKergall/osmbonuspack>)

6.2.3 Úvodní obrazovka

První věcí, kterou uživatel po nainstalování aplikace spatří je aktivita úvodní obrazovky, takzvaný splash screen.



Obr. 23 Úvodní obrazovka aplikace (tzv. splash screen)

XML – splash_screen.xml

Vzhled úvodní obrazovky byl v XML vytvořen pomocí kontejneru *ConstraintLayout*, do něhož bylo vloženo předem vytvořené logo aplikace jako prvek *ImageView*. Loga a ikony se typicky nachází v projektové složce *drawable*. Pod logo byly umístěny dvě instance prvku *TextView*. V té první je vepsán název aplikace a je zde použit font Gurajada.

Tento font byl dostupný zdarma na Google Fonts a byl použit i v dalších částech aplikace. Instalace fontu je jednoduchá, stačí vložit font ve formátu *.ttf* do projektové složky *res* a poté se na font odkázat v XML, v tomto případě řádkem

```
android:fontFamily="@font/gurajada_regular".
```

Ve druhém prvku *TextView* je napsán menším písmem uvítací text. Tento text byl uložen ve zvláštním souboru *strings.xml* ve složce *values* pod názvem *popis_aplikace*. Odkázáno na něj bylo řádkem

```
android:text="@string/popis_aplikace".
```

Úplně dole se nachází tlačítko s nápisem **ZAČÍT**. Jedná se o prvek *Button*, kterému byla nastavena požadovaná velikost a barva. Barva byla nastavena odkázáním na předem vytvořený XML soubor *bgbutton*, kde byla definována barva a poloměr kulatých rohů tlačítka.

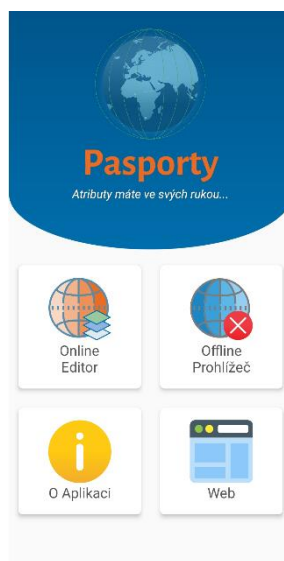
Java – SplashScreen.java

V Java kódu byly použity animace loga, textů a tlačítka, aby aktivita úvodní obrazovky zaujala uživatele. Tyto animace se nacházejí v projektové složce *anim* a mají názvy *forimg.xml* a *frombottom.xml*.

Bylo zde také nutné nastavit tlačítko **ZAČÍT**, aby aplikace po kliknutí přešla do další aktivity a to do hlavního menu. Toto bylo nastaveno pomocí metody *onClick*.

6.2.4 Hlavní menu

Další vytvořenou aktivitou bylo hlavní menu aplikace. Zde se nachází čtyři tlačítka, která „vedou“ do Online Editoru, Offline Prohlížeče, do sekce O Aplikaci a poslední tlačítko otevře v prohlížeči webovou stránku k bakalářské práci.



Obr. 24 Hlavní menu aplikace

XML – main_menu.xml

V XML kódu hlavního menu bylo nejprve využito kontejneru *ConstraintView* a poté byla obrazovka rozdělena kontejnery *FrameLayout* na dvě části. V první části nahoře se nachází animace zeměkoule přidaná pomocí knihovny Lottie a prvku *LottieAnimationView*. Animace byla vytvořena uživatelem Nicky Santoro a byla volně dostupná ke stažení na webových stránkách LottieFiles (zdroj: <https://lottiefiles.com/14178-globe>). Soubor animace bylo nutné vložit do složky assets v projektu.

Pod animací se podobně jako v úvodní obrazovce nachází dva prvky *TextView*. V nich je uveden název aplikace Pasporty a slogan „Atributy máte ve svých rukou...“. Poté už následuje druhá část obrazovky rozdělená kontejnerem *FrameLayout*.

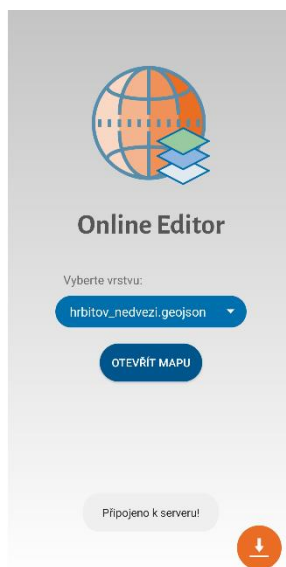
Zde je využito kombinace kontejneru *GridLayout* (tedy mřížky) a prvků *CardView*. *GridLayout* má nastaven počet řádků a sloupců na 2 × 2 (`android:columnCount="2"` a `android:rowCount="2"`), čímž bylo zajištěno designově pohledné rozložení čtyř ovládacích tlačítek. Tlačítka jsou vedena jako karty – prvky *CardView* – které obsahují kontejner *LinearLayout* a prvky *ImageView* spolu s *TextView*. Mají tak nastaveny svoje ikony, které byly vytvořeny autorem, a svůj text označující části aplikace, kam dané tlačítko „vede“.

Java – MainMenu.java

Hlavní menu je opět animované podobně jako úvodní obrazovka aplikace. Bylo zde využito animace *frombottom*. A také zde bylo nastaveno jednotlivým tlačítkům, jakou aktivitu mají otevřít. V případě tlačítka OnlineEditoru po kliknutí probíhá kontrola internetového připojení. Nemá-li uživatel k dispozici rychlé připojení k Internetu (400 kbps a více), aktivita se nespustí a místo ní se objeví chybová hláška „Pomalá rychlost připojení! Nelze otevřít Online Editor!“. Kontrola je zajištěna třídou *Connectivity* vytvořenou Emilem Davtyanem (zdroj: <https://gist.github.com/emil2k/5130324>).

6.2.5 Online část – Online Editor

Nejdůležitější a nejkomplexnější částí aplikace je Online Editor. Tento byl totiž řešen hybridně. Jeho první, nativní, částí je úvodní obrazovka Online Editoru, kde si uživatel zvolí datovou vrstvu pasportu, kterou chce zobrazit v mapě. Druhá část, samotná mapa, je pak naprogramována v jazycích HTML, PHP a JavaScript s využitím open-source mapové knihovny Leaflet a je s úvodní obrazovkou Online Editoru propojena přes novou aktivitu s kontejnerem *WebView*.



Obr. 25 Úvodní obrazovka Online Editoru (dole upozornění, že bylo navázáno spojení se serverem)

XML – `activity_select_online.xml`

Obsah XML kódu aktivity úvodní obrazovky Online Editoru, kde probíhá výběr dat k zobrazení v mapě, je opět celkem jednoduchý k pochopení. Opět je zde využito kontejneru *ConstraintLayout*. Nahoře na obrazovce se nachází prvek *ImageView* zobrazující ikonu Online Editoru. Pod ním je prvek *TextView*, který slouží jako popis prvku *Spinner*.

Spinner je rozbalovací nabídka, v níž je možné vybírat vrstvu. Na *Spinner* byl použit již zmíněný `@drawable/bgbutton`, čímž bylo docíleno stejného vzhledu, který mají vlastní vytvořená tlačítka. Tento *Spinner* po kliknutí zobrazuje všechny dostupné datové vrstvy ze serveru.

Pod *Spinnerem* se nachází tlačítko – prvek *Button* s textem OTEVŘÍT MAPU, jímž se vstupuje do mapové části Online Editoru. Design tlačítka je opět stejný jako u jeho předchozích instancí.

V pravém dolním rohu je tlačítko s ikonou stahování (ikona byla dostupná v Android Studiu), toto tlačítko je ve skutečnosti *ImageView*. Pomocí tohoto tlačítka je vybraná vrstva stažena do zařízení.

Java – `SelectOnline.java`

V Java kódu aktivity `SelectOnline` je toho podobně jako u následující aktivity `Online` opravdu mnoho a je doporučeno prostudovat si celý zdrojový kód s komentáři. Zde bude popsán pouze základní princip.

Je-li uživatel připojen k Internetu a je-li jeho připojení dostatečně rychlé, bude z menu vpuštěn do aktivity `Online Editoru`. Zde se zavolá metoda `addItemOnSpinner()`, která naplní *Spinner* aktuálními soubory (vrstvami pasportu) přítomnými na serveru.

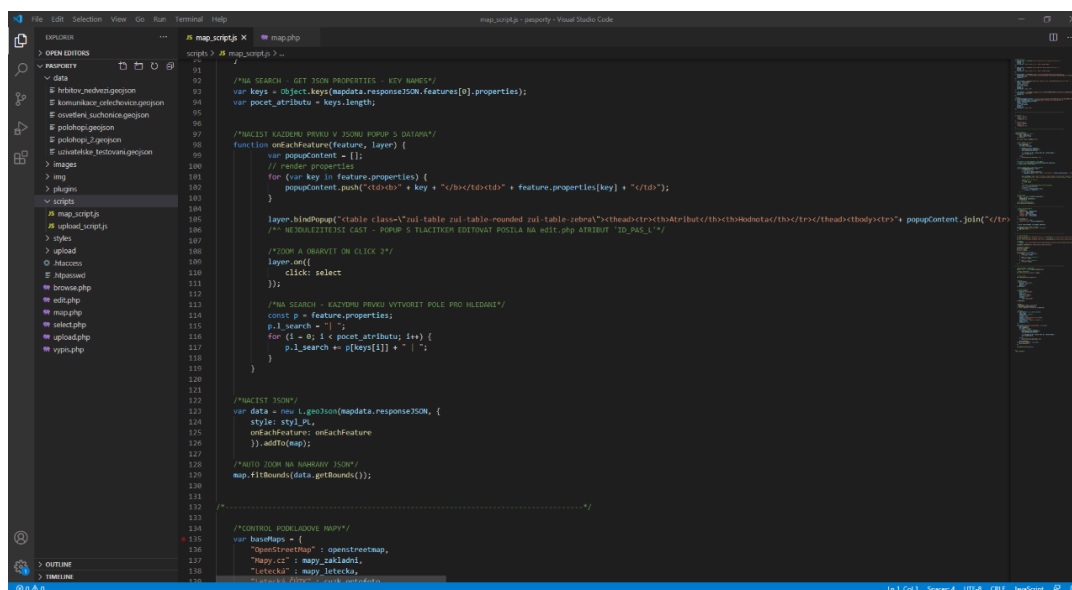
K získání dat ze serveru byla naprogramována třída *fetchData*, která funguje tak, že v pozadí načte obsah webové stránky do proměnné. Vedle této třídy byl na serveru vytvořen PHP skript *vypis.php*, který vypisuje aktuální data ve složce oddělené středníkem. Tento textový řetězec obsahující názvy vrstev pasportů je tedy pomocí třídy *fetchData* uložen do proměnné, tato je podle středníků rozdělena do pole jednotlivých souborů. Těmito je pak naplněn *Spinner*.

Tlačítko stažení vrstvy funguje na podobném principu. Nejprve je ve *Spinneru* vybrána vrstva a na základě jejího názvu je poté přes metodu *startDownloading* stažen ze serveru. Všude v těchto zmíněných případech samozřejmě v kódu figuruje cesta na samotný server, ta je napevno zapsána v souboru *strings.xml* v projektové složce *values* jako následující řádek:

```
<string name="web">https://gis.upol.cz/pasporty/</string>
```

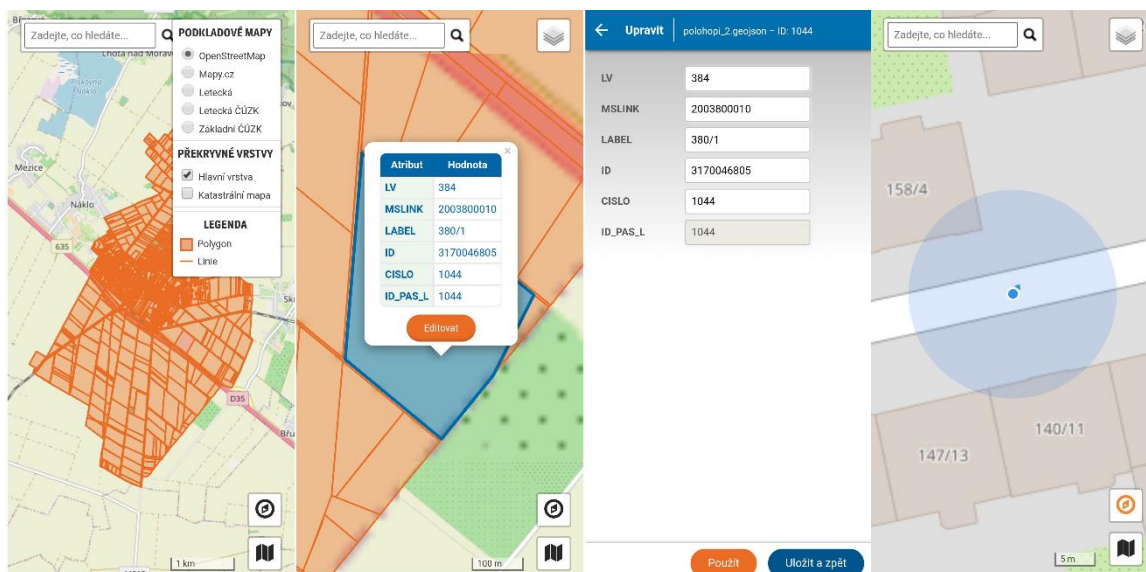
K tomuto odkazu jsou pak na základě požadované operace dynamicky přidávány další části plné cesty.

Zmíněné řešení funguje i v případě, že se uživatel rozhodne po výběru vrstvy tlačítkem OTEVŘÍT MAPU mapu skutečně otevřít. Aplikace předá spojenou kompletní webovou adresu další aktivitě, kterou je samotná aktivita *Online*, v ní je v kontejneru *WebView* tato adresa otevřena a zobrazena tak, že vypadá jako součást nativní aplikace, ale nejedná se o ni.



Obr. 26 Vývojové prostředí Visual Studio Code

Jádro aktivity *Online Editoru* bylo naprogramováno jako responzivní webová mapová aplikace založená na open-source knihovně *Leaflet*. Zdrojový kód webových částí hybridní aplikace *Pasporty* je také k dispozici jako příloha na CD. Programování probíhalo ve vývojovém prostředí *Visual Studio Code* (viz obrázek 26) od společnosti *Microsoft*.



Obr. 27 Náhled na funkce Online Editoru – mapové pole, vyskakovací atributová tabulka, rozhraní pro editaci atributů a lokalizace uživatele

Využití knihovny Leaflet

Hlavní částí tohoto webu je stránka *map.php*, která obsahuje JavaScriptový soubor *map_script.js*, kde jsou naprogramovány jednotlivé prvky knihovny Leaflet pro potřebu Online Editoru aplikace Pasporty.

Po potvrzení výběru vrstvy ve *Spinneru* v předchozí aktivitě se v této aktivitě přejde například na odkaz **.../pasporty/map.php?src=hrbitov_nedvezi.geojson**, kde za **src=** se nachází název souboru – název vybrané vrstvy pasportu. Pomocí skriptu na řádcích 56–68 v souboru *map.php* je do knihovny Leaflet tato vrstva nahrána.

Poté je vektorová vrstva ve formátu GeoJSON zobrazena v mapové knihovně Leaflet (viz zdrojový kód souboru *map_script.js* a komentáře autora aplikace). U každého mapového prvku byla vytvořena vyskakovací bublina (popup) s atributy, která se zobrazí po kliknutí. Pro pohledně zobrazení bylo využito CSS kódu tabulek „zui“ (nachází se v souboru *style_map.css*) (zdroj: <https://zource.github.io/zui/table.html>) – viz druhý snímek obrazovky zleva z koláže na obrázku 27.

Byly zde dále pomocí open-source pluginů *Leaflet.Locate* (odkaz: <https://github.com/domoritz/leaflet-locatecontrol>) a *L.EasyButton* (odkaz: <https://github.com/CliffCloud/Leaflet.EasyButton>) vytvořeny tlačítka lokalizace uživatele a návratu zpět na mapu.

Také bylo nastaveno pět podkladových map a jedna mapa překryvná (katastrální), které lze zvolit tlačítkem vpravo nahoře. Vlevo od tohoto tlačítka byl přidán vyhledávač, jenž umí vyhledávat ve všech prvcích zobrazené vrstvy pasportu. Byly také přidány další kompoziční prvky mapy – grafické měřítko vpravo dole a do výběru vrstev byla pomocí jQuery zabudována také legenda. Tlačítka byla dostatečně zvětšena, aby byla uzpůsobena používání na dotykovém mobilním zařízení. Bylo také zrušeno zoomování pomocí tlačítek, jelikož dnešní mobilní zařízení disponují funkcí multitouch.

Co se týče editace atributů, tak zde se vrátíme k vyskakovací tabulce s atributy. Pod tabulkou se nachází tlačítko Editovat. To uživatele pošle do editačního rozhraní nacházejícího se na stránce *edit.php*. Jeho vzhled byl tvořen s cílem připodobnit se nativní části aplikace (viz třetí snímek obrazovky zleva z koláže na obrázku 27) ať už typem a vzhledem horní navigační lišty nebo vzhledem tlačítek. Po provedení editace

atributů je potřeba potvrdit změny tlačítkem Použít, a až se objeví potvrzovací hláška „Změny provedeny!“, lze poté zmáčknout tlačítko Uložit a zpět pro návrat zpět na mapu, kde se již bude nacházet upravená vrstva.

Proces editace je velmi složitý a pro úplné pochopení je potřeba nastudovat zdrojový kód části *edit.php*. Je v něm využito několik funkcí pro práci s textem a formátem JSON v PHP a JavaScriptu. GeoJSON je zde převáděn na text za účelem zobrazení v editační tabulce a po úspěšné editaci je opět převáděn zpět a ukládán/nahráván na server jako GeoJSON. Je zde také využito mnoho cyklů a kombinací několika programovacích a skriptovacích jazyků dohromady.

XML – activity_online.xml

Vrátíme-li se zpět do prostředí Android Studia, tak XML soubor *activity_online.xml* je řešen celkem jednoduše, sestává totiž pouze z kontejneru *ConstraintLayout*, v němž je přes celou obrazovku nastaven kontejner *WebView*, kde se odehrává vše zmíněné v podkapitole výše.

Java – Online.java

Totéž platí i u Java kódu této aktivity, při pohledu na její zdrojový kód zde probíhá pouze předávání odkazu na webovou stránku zobrazenou v kontejneru *WebView* z předchozí aktivity, kde byla na *Spinneru* vybrána vrstva pasportu. Správnost propojení těchto dvou aktivit je velmi důležitá a musí být naprogramována správně, jinak by vůbec nedošlo k otevření mapy.

Dále jsou tu opět pomocí knihovny *RxPermissions* kontrolována uživatelská povolení přístupu k poloze zařízení a k Internetu. Při stisknutí tlačítka zpět na zařízení se Aplikace vyskakovacím oknem uživatele dotáže, zda chce Online Editor opustit.

6.2.6 Offline část – Offline Prohlížeč

Pro tvorbu Offline části byl zvolen nativní přístup. Bylo využito open-source knihovny *OSMDroid* spolu s jejím rozšířením *OSMBonusPack*, které umožňuje práci se geografickými daty v souborech typu GeoJSON. Offline Prohlížeč je obdobně jako Online Editor rozdělen na dvě aktivity.

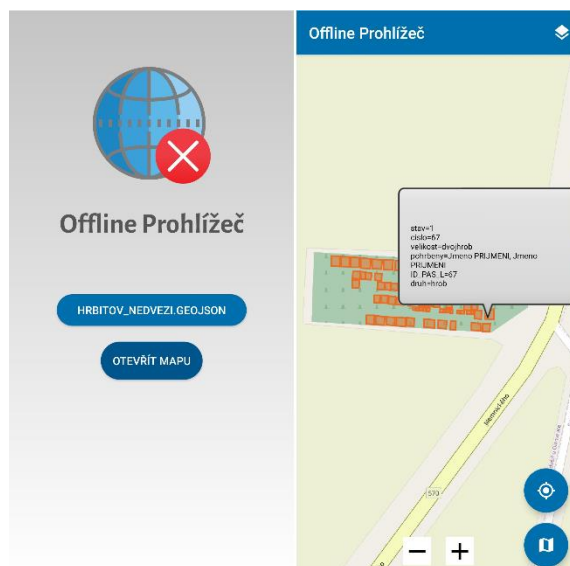
XML – activity_select_offline.xml

Aktivita *activity_select_offline.xml* vypadá podobně jako aktivita výběru u Online Editoru a i její XML kód vypadá podobně. Jediný rozdíl je ve způsobu výběru vrstvy. Místo *Spinneru* se zde nachází prvek *Button*, jemuž se po zvolení vrstvy změní popisek na vybranou vrstvu pasportu.

Java – SelectOffline.java

Kód Java aktivity výběru Offline Prohlížeče je také obdobný jako u Online Editoru. Rozdíl je zde po kliknutí na tlačítko výběru vrstvy. Nejprve uživateli vyskočí okno s upozorněním, že doporučená velikost vybrané vrstvy GeoJSON je 1 MB. To je dáno chováním knihovny *OSMDroid*, která má s většími daty problém.

Po kontrole uživatelských povolení přístupu k úložišti zařízení se uživateli objeví okno s výběrem vrstvy z paměti zařízení. Zde si uživatel vybere vrstvu a její cesta se potvrzení tlačítkem OTEVŘÍT MAPU předá do další aktivity, podobně jako u Online Editoru. Pro získání absolutní cesty k souboru zde byla využita třída *FileUtils* od Satyawana Hajareho (zdroj: <https://stackoverflow.com/a/55469368>).



Obr. 28 Úvodní obrazovka (nalevo) Offline Prohlížeče a jeho prostředí – mapové pole s vyskakovací bublinou obsahující atributy (napravo)

XML – activity_offline.xml

XML kód aktivity samotného Offline Prohlížeče se skládá z kontejneru *CoordinatorLayout*, který umožňuje použít takzvané *FloatingActionButtony*, tedy plovoucí akční tlačítka (na obrázku 28 na pravém snímku obrazovky se nacházejí vpravo dole). Tato tlačítka je nutné v XML umístit jako první, jelikož se musí zobrazit nad samotným mapovým polem. Tlačítkům byly nastaveny ikony (dostupné přímo v Android Studiu), zvoleny vhodné barvy a určena jejich pozice. Cílem je, aby jejich vzhled, umístění a velikost odpovídaly tlačítkům v Online Editoru.

Pod *FloatingActionButtony* byl poté vložen samotný kontejner *MapView* z knihovny OSMDroid. Tomuto bylo nastaveno, aby zaplnil celou obrazovku zařízení.

V horní části obrazovky se také nachází menu s tlačítkem výběru vrstvy, toto bylo nadefinováno v souboru *AndroidManifest.xml* a v souboru *menu_layers.xml*, který se nachází ve složce menu v projektu. V souboru *menu_layers.xml* je nadefinováno i ovládání podkladových map. Barva menu je definována v souboru *styles.xml*, jenž se nachází v projektové složce values.

Java – Offline.java

Java kód Offline Prohlížeče byl převzat z dokumentace knihovny OSMDroid a byl upraven pro potřeby aplikace Pasporty (odkaz: <https://osmdroid.github.io/osmdroid/How-to-use-the-osmdroid-library.html>).

Bylo zde nastaveno otevření vrstvy pasportu z cesty, která byla aktivitě předána aktivitou výběru a to pomocí nových funkcí knihovny OSMDroidBonusPack, viz úryvek kódu níže.

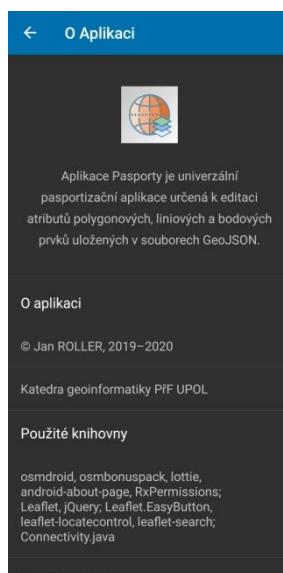
```
KmlDocument kmlDocument = new KmlDocument();
Bundle extras = getIntent().getExtras();
souborUri = Uri.parse(extras.getString("vybranySoubor"));
soubor = new File(souborUri.getPath());
kmlDocument.parseGeoJSON(soubor);
```

Po přidání vybrané vrstvy do mapy byl nastaven zoom na tuto vrstvu stejně jako u Online Editoru. Dále byl nastaven znakový klíč a barvy tak, aby odpovídaly těm v Online Editoru (viz řádek 103). Byla také zprovozněna tlačítka *FloatingActionButton*, která fungují stejně jako v Online Editoru, a také nabídka volby podkladových map. Hláška při stisknutí tlačítka zpět je také stejná a po potvrzení dojde k opuštění aktivity Offline Prohlížeče.

6.2.7 Ostatní

Poslední funkce mobilní mapové aplikace Pasporty se nachází pod dvěma spodními tlačítka v hlavním menu. Právě tlačítko spustí webový prohlížeč a otevře v něm webové stránky k této bakalářské práci. Tato funkce je nadefinována v souboru MainMenu.java na řádce 62.

Levé tlačítko skrývá nabídku O Aplikaci, která nabízí informace o aplikaci, o použitých knihovnách, mapové zdroje a kontakt na autora. Tato byla vytvořena pomocí open-source knihovny Android About Page, která je určena přímo pro tvorbu těchto aktivit



Obr. 29 Sekce O Aplikaci

XML – activity_about.xml

Co se XML kódu activity_about.xml týče, nenachází se zde nic kromě kontejneru *ConstraintLayout*. Ten je roztažen, aby pokryl celou obrazovku. Je zde ovšem přidáno také menu s tlačítkem zpět. To je nadefinováno v souboru AndroidManifest.xml.

Java – About.java

Nastavení Android About Page proběhlo podle pokynů na stránkách tvůrce (<https://github.com/medyo/android-about-page>). Viz zdrojový kód aplikace *Pasporty*.

6.3 Webové rozhraní pro upload dat

Vedle samotné mapové aplikace pro systém Android bylo vytvořeno také webové rozhraní pro nahrání dat – jednotlivých vrstev ve formátu GeoJSON – na server. Tato data se poté zobrazí v Online Editoru v aplikaci Pasporty.

Bylo naprogramováno v jazycích HTML5, PHP a JavaScript a disponuje stejným CSS designem jako web k bakalářské práci. Jeho zdrojový kód lze opět nalézt v přílohách na CD.

Rozhraní pro upload dat do aplikace Pasporty se nachází na adrese <https://gis.upol.cz/pasporty/upload.php> nebo se na něj lze dostat přes hlavní nabídku webu k této bakalářské práci. Funkce uploadu je z bezpečnostních důvodů uzamčena přístupovým jménem a heslem.

- **Přístupové jméno:** admin
- **Heslo:** pasporty

Toto rozhraní dále nabízí ke stažení pilotní data, která byla použita při testování aplikace, a také je zde odkaz na převodník dat Online Geodata Converter, kde si nezkušení uživatelé mohou data (v omezené míře) převést do požadovaného formátu GeoJSON.

Data lze ze serveru také stáhnout a smazat, a to pomocí upravené a do češtiny přeložené knihovny Simple PHP File Manager, kterou vytvořil John Campbell (zdroj: <https://github.com/jcampbell1/simple-file-manager>). Tato knihovna je také chráněna heslem uvedeným výše.

The screenshot shows the 'PASPORTY' web interface. At the top, there is a blue header with the title 'PASPORTY' and a navigation link 'Zpět NAHRÁT DATA'. The main content area is titled 'NAHRÁT DATA' and contains instructions: 'Zde lze do pasportizační aplikace nahrát data ve formátu GeoJSON. A nebo můžete data stáhnout či smazat zde.' Below this are two buttons: 'Vybrat GeoJSON' and 'Nahrát'. The second part of the interface is titled 'DŮLEŽITÉ ODKAZY A TESTOVACÍ DATA' and features four cards:

- Online Geodata Converter:** Online převodník geodat, kde je možné převést různé formáty do požadovaného GeoJSONu. Omezení je na 3 datosety nebo 5 MB dat / měsíc. Je důležité nastavit cílový souřadnicový systém na WGS 84 (EPSG: 4326). Button: Přejít.
- Geojson.io:** Nástroj k editaci geodat v souborech typu GeoJSON. Button: Přejít.
- Testovací Data 1:** Pilotní data 1 ve formátu GeoJSON pochází z projektu inovačního Voucheru Olomouckého kraje pro obec Příkazy. Jedná se o data parcel. Soubory polohapi a polohapi_2 v archivu ZIP (633 kB). Button: Stáhnout.
- Testovací Data 2:** Pilotní data 2 ve formátu GeoJSON byla vytvořena autorem práce. Jedná se o 3 úkázkové vrstvy možných pasportů hřbitova, komunikací a veřejného osvětlení. 3 vrstvy (SHP a GeoJSON) v archivu ZIP (21 kB). Button: Stáhnout.

The footer contains author information: 'AUTOR: Jan ROLLER, info@jan.horaz@gmail.com', 'VEDOUČÍ PRÁCE: Mgr. Rostislav NĚTEK, Ph.D., rostislav.netek@upol.cz', logos for 'Katedra geoinformatiky' and 'Univerzita Palackého v Olomouci', and a copyright notice: '© Jan ROLLER, Olomouc 2019/20 | This template is made with by Colorlib'.

Obr. 30 Rozhraní pro upload dat na server (do aplikace) jako součást webu k bakalářské práci

7 TESTOVÁNÍ

Neopomenutelnou částí tvorby jakékoliv aplikace, ať už se jedná o desktopovou, webovou, či mobilní, je testování. Testováním aplikace se tvůrci a programátoři snaží docílit eliminace nežádoucích chyb a bugů, které se během tvorby aplikace chtě nechtě objeví skoro vždy. Více v podkapitole 3.6.

Martinák (2012, s. 3) uvedl, že “na testování je (...) možné nahlížet z mnoha pohledů (od analytiků až po koncové zákazníky a uživatele) a jejich pohledy se liší už z principu“ a popsal testování jako „jako zkoumání softwaru za účelem odhalení chyb“.

Testování v této práci bylo tedy rozděleno na *výkonnostní*, kde byly vyhledávány a odhalovány chyby, a na *uživatelské*, kde byla výsledná hotová aplikace poskytnuta uživatelům, kteří ji vyzkoušeli, ohodnotili a poskytli autorovi práce zpětnou vazbu.

7.1 Výkonnostní testování

V průběhu procesu tvorby aplikace Pasporty a také po jejím úspěšném vytvoření a vyexportování instalačního balíčku aplikace bylo prováděno výkonnostní testování. V rámci výkonnostního testování bylo hlavním úkolem *hledat a opravovat chyby*.

Martinák (2012, s. 3) ve své diplomové práci dále uvedl, že softwarové chyby se „dají nazývat různě (bugy, defekty, problémy, selhání, atp.) a každý člověk může daným názvem myslet něco jiného. To už je potom na domluvě zainteresovaných osob (návrhář, vývojář, tester, zákazník, atd.), aby používali společný slovník. Obecně je chyba softwaru faktorem, který snižuje hodnotu/kvalitu softwaru“. Podle Pattona (2000) se za chybné chování dají označovat situace, kdy:

- Software nedělá něco, co by podle specifikace dělat měl.
- Software dělá něco, co by podle specifikace dělat neměl.
- Software dělá něco, o čem se specifikace nezmiňuje.
- Software nedělá něco, o čem se specifikace sice nezmiňuje, ale měla by se zmiňovat.
- Software je obtížně srozumitelný, těžko se s ním pracuje, je pomalý nebo (podle názoru testera) jej koncový uživatel nebude považovat za správný.

Martinák (2012, s. 3) také přišel s několika body, s nimiž je potřeba počítat při testování, zde jsou uvedeny některé z nich:

- Ať se testuje jakkoli dlouho, vždycky tam zůstanou nějaké chyby.
- Není možné pokrýt úplně všechny případy užití.
- Některé chyby je velice obtížné odhalit a jejich odhalování se třeba ani nevyplatí.
- Testování vhodné pro jeden produkt nemusí být vhodné pro jiný.

Průběžné výkonnostní testování aplikace úzce souvisí s agilní metodou vývoje softwaru (viz podkapitola 3.6.2). Je-li nalezena v průběžné verzi nějaká chyba, musí být odstraněna a je potom vydána další verze s více funkcemi atd. Tento proces je opakován tak dlouho, dokud aplikace jednak neobsahuje veškeré funkce, které jsou v návrhu aplikace vymyšleny a navrženy, a za druhé neobsahuje žádné chyby. Toto ovšem nelze nikdy spolehlivě zajistit, viz první bod od Martináka (2012) v předchozím odstavci.

Pro potřebu výkonnostního testování mobilní aplikace je nezbytné disponovat širokou škálou různých zařízení. Tato by měla mít jednak odlišné hardwarové vybavení (procesor, operační paměť, velikost displeje), tak odlišné softwarové vybavení (v tomto případě různé verze operačního systému Android). Na výkonnostním testování se z tohoto důvodu podíleli kromě autora ještě dva testéři, aby bylo docíleno největšího počtu různých zařízení.

Výkonnostní testování probíhalo celkem na pěti zařízeních: na výkonném smartphonu vyšší třídy Huawei P20 Pro, smartphonech střední třídy Xiaomi Redmi 4X a Sony Xperia Z5, dále na smartphonu nižší třídy Huawei Y6 a také na tabletu Huawei MediaPad M5. Zařízení odpovídala požadavkům uvedeným v předchozím odstavci – měla rozdílný hardware a také byla vybavena různými verzemi operačního systému Android. Podrobné specifikace jsou uvedeny v tabulce 3 níže.

Tab. 3 Zařízení použita k výkonnostnímu testování

Výrobce	Modelová řada	Rok	Velikost a rozlišení displeje	Procesor	RAM	Verze OS Android
Huawei	P20 Pro	2018	6,1" 1080 × 2240 px	Osmijádrový; HiSilicon Kirin 970 (4 × 2,4 GHz a 4 × 1,8 GHz)	6 GB	10 (EMUI 10.0.0)
Xiaomi	Redmi 4X	2017	5,0" 720 × 1280 px	Osmijádrový; Snapdragon 435 (1,4 GHz)	3 GB	7.1.2 (MIUI Global 11.0.2)
Sony	Xperia Z5	2015	5,2" 1080 × 1920 px	Osmijádrový; Snapdragon 810 (4 × 1,5 GHz a 4 × 2,0 GHz)	3 GB	7.1.1
Huawei	Y6	2015	5,0" 720 × 1280 px	Čtyřjádrový; Snapdragon 210 (1,1 GHz)	1 GB	5.1.1
Huawei	MediaPad M5 10	2018	10,8" 2560 × 1600 px	Osmijádrový; HiSilicon Kirin 960 (4 × 2,1 GHz a 4 × 1,8 GHz)	4 GB	10 (EMUI 9.1.0.332)

Poznámky:

Pro porovnání Huawei P20 Pro, Xiaomi Redmi 4X a Sony Xperia Z5 navštivte tento odkaz vedoucí na web *GSMarena*: <https://bit.ly/3ibgz8D>

Také je nutné poznamenat, že zařízení Huawei Y6 nesplňuje minimální požadavky na verzi operačního systému Android (API level 23; Android 6.0 Marshmallow), takže do něj výslednou aplikaci nebylo možné nainstalovat. Byla na něm nicméně testována a laděna onlinová část aplikace – Online Editor.

7.1.1 Výsledky výkonnostního testování

Jelikož byla aplikace vyvíjena agilní metodou vývoje, tak výkonnostní testování probíhalo během tvorby aplikace. Stručně řečeno byla po každé nové přidané funkci vytvořena testovací verze a tato byla otestována autorem a testery na zmíněných pěti zařízeních. Během tvorby aplikace bylo postupně vytvořeno celkem 43 testovacích verzí. Na základě pozorování a testování bylo provedeno vychytávání a ladění chyb. Tento proces probíhal iterativně, dokud nebyla aplikace hotová a bez jakýchkoliv viditelných chyb. Byla tak vydána plně funkční verze aplikace *Pasporty 1.0*.

Poté byla aplikace nejprve rozeslána dalším, neevidovaným testerům. Zde je nutné poznamenat, že i když byla aplikace nainstalována na nespočet různých zařízení testerů, objevily se dva případy, kdy aplikace nefungovala, i přesto, že zařízení splňovalo minimální požadavky. Jednalo se o smartphone ASUS Zenfone Max Plus (ASUS_X018D) s operačním systémem Android 8.1 a o smartphone Xiaomi Redmi 6A (M1804C3CG) opět s OS Android 8.1. Z tohoto důvodu bylo provedeno ladění aplikace v emulátoru se stejnými nastaveními a verzí OS Android, jako u dvou zmíněných případů a *nedošlo k žádné chybě*, aplikace *bez problémů fungovala*. Je tedy velmi pravděpodobné, že chyba byla na straně testerů.

Po úspěšném výkonnostním testování, jehož výsledkem byla hotová aplikace, bylo přistoupeno k testování uživatelskému.

7.2 Uživatelské testování

Uživatelské testování aplikace *Pasporty* bylo řešeno pomocí dotazníkového šetření. Zde byla výsledná hotová aplikace poskytnuta uživatelům, kteří ji vyzkoušeli, ohodnotili a poskytli autorovi práce zpětnou vazbu.

Podle UX agentury sherpas „uživatelské testování hodnotí především použitelnost a pochopení stránky nebo aplikace. Ověřuje, zda uživatelé pracují s rozhraním efektivně a správně rozumí instrukcím a informacím“.

Dotazníkové šetření bylo provedeno na platformě Google Forms, kde byl vytvořen jednoduchý dotazník pokrývající všechny oblasti testované aplikace. Autorem práce bylo vybráno deset respondentů, kdy jedna polovina měla geoinformatické nebo kartografické vzdělání a druhá polovina nikoliv. Tím byl zajištěn pohled z obou stran a bylo tak docíleno lepšího otestování aplikace lidmi z různých oblastí života.

7.2.1 Struktura dotazníku a otázky

Vytvořený dotazník se skládal z několika částí. V úvodu byly, kromě představení autora a tématu bakalářské práce, obsaženy také informace o průběhu dotazníkového šetření.

V další části byly zjištěny základní informace respondentovi: pohlaví a věk, dále zde byly otázky, zda má respondent geoinformatické nebo kartografické vzdělání, zda má zkušenosti s tvorbou mobilních aplikací a také zda ví, co znamenají pojmy pasport a pasportizace. Odpověděl-li respondent na tuto otázku odpovědí NE, byl pak krátkým shrnutím a popisem těchto pojmů informován.

Zkoumán zde byl také hardwarový a softwarový aspekt (viz podkapitola 5.1.1) a tedy otázkami na typ respondentova zařízení (smartphone nebo tablet), název jeho výrobce a jeho modelovou řadu. Respondent byl taktéž dotázán na verzi OS Android v jeho mobilním zařízení.

V následující části dotazníku byl respondentovi poskytnut odkaz na stažení testované aplikace *Pasporty*, byl požádán, aby si ji nainstaloval a s aplikací byl také

v pár odstavcích seznámen. Před vyplňováním další části dotazníku, zabývající se samotným hodnocením aplikace, byl respondent také požádán o důkladné otestování aplikace a prozkoumání všech jejích částí a funkcí. Pro potřebu editace byla autorem vytvořena a do aplikace nahrána vrstva *uzivatelske_testovani.geojson*. Jednalo se o autorem vytvořenou polygonovou vrstvu imaginárního pasportu hřbitova.

V další části dotazníku byl pokryt Online Editor. Byly zde hodnoceny jednotlivé funkce a aspekty aplikace na škále 1–5, kde 5 bylo nejvíce a 1 nejméně:

- *Rychlost* (Rychlost načítání Online Editoru, rychlost a svižnost editace apod.)
- *Intuitivnost* (Je-li vše srozumitelné a pochopitelné, navozují-li ikony pocit toho, jakou akci vykonávají apod.)
- *Srozumitelnost znakového klíče a legendy*
- *Obtížnost editace*

Podobné kategorie byly zvoleny i v následující části dotazníku, ta totiž pokrývala Offline Prohlížeč:

- *Rychlost* (Rychlost a svižnost Offline Prohlížeče.)
- *Intuitivnost*
- *Srozumitelnost znakového klíče*

V předposlední části dotazníku byly dvě hodnotící otázky (opět na stupnici 1–5) na hlavní menu a na sekci O Aplikaci.

Poslední část dotazníku se věnovala subjektivním hodnocením. Nacházelo se zde textové pole a respondent byl požádán o sepsání pár vět, jimiž aplikaci subjektivně ohodnotil. Následovalo poté ještě celkové číselné hodnocení aplikace (opět na stupnici 1–5), podobně jako na portálu obchodu Google Play (nebo Apple App Store).

Poté už následoval jen závěr, kde měl respondent možnost podepsat se a vyjádřit se k dotazníku nebo k bakalářské práci.

7.2.2 Výsledky uživatelského testování

Poté, co autor práce dostal odpovědi od všech deseti zúčastněných respondentů, bylo provedeno vyhodnocení výsledků dotazníkového šetření a uživatelského testování aplikace. Z výsledků bylo vyhotoveno několik grafů. Některé z nich jsou umístěny zde v textu, další se nachází na konci práce jako vázaná příloha 2.

Informace o respondentech a jejich zařízeních

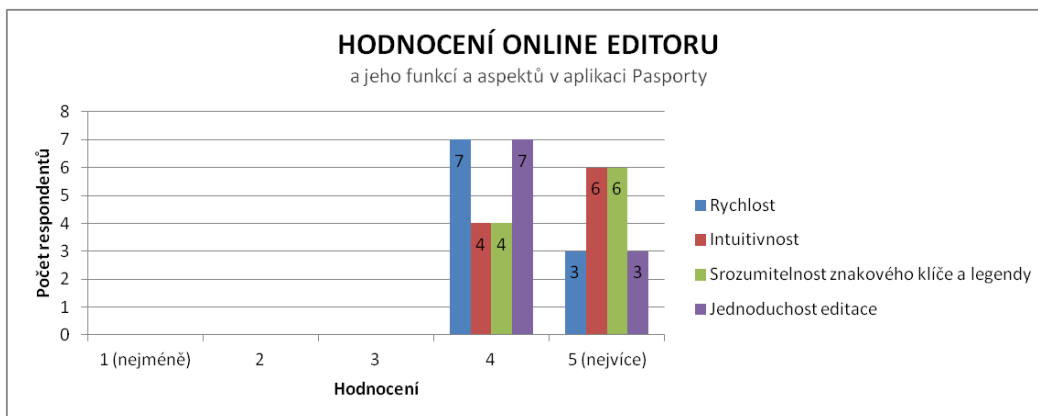
Jak bylo zmíněno výše, tak z deseti respondentů mělo pět respondentů geoinformatické vzdělání a zbylých pět ne. Co se pohlaví týče, tak se uživatelského testování zúčastnilo sedm mužů a tři ženy. Devět z deseti respondentů bylo ve věku 18 až 25 let, jeden respondent označil možnost 26 až 35 let. Z oslovených testerů měli dva zkušenost s tvorbou mobilních aplikací. Termíny pasport a pasportizace znalo sedm z deseti oslovených respondentů.

Co se týče informací o zařízeních respondentů, tak jeden respondent použil tablet, ostatní smartphone. V uživatelském testování převládala zařízení firmy Xiaomi, konkrétně modely Mi Note 10 Pro, Mi Mix 2, Redmi 4X, Redmi 5, Redmi 6 a Redmi Note 7. Dále byly použity dvě zařízení firmy Huawei, modely P20 Lite a Mate 20 Pro. Po jednom zařízení pak bylo od výrobců Motorola (Moto G5S Plus) a Sony (Xperia Z5).

V zařízeních převládá druhý nejnovější OS Android, verze 9. Nejstarší byla verze 7.1. Grafy k této kategorii se nachází na konci práce jako vázaná příloha 2.

Hodnocení Online Editoru

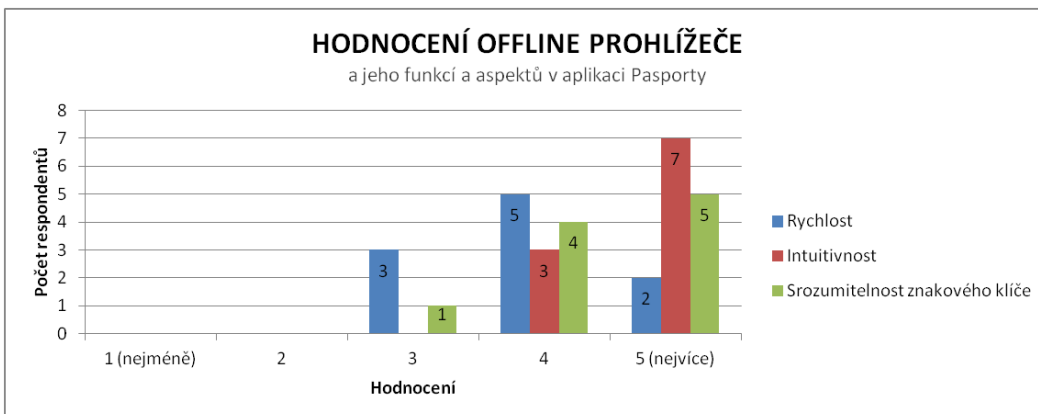
Online Editor byl ve všech zkoumaných oblastech ohodnocen nadprůměrně až skvěle. Nejvíce byla oceněna jeho Intuitivnost a Srozumitelnost znakového klíče a legendy. Hodnocena stupněm 4 byla převážně Rychlost a Jednoduchost editace.



Obr. 31 Graf hodnocení Online Editoru

Hodnocení Offline Prohlížeče

Offline Prohlížeč byl hodnocen hůře než Online Editor. Třem respondentům se zdál pouze průměrně rychlý. Nejvíce testerů ocenili jeho Intuitivnost.



Obr. 32 Graf hodnocení Offline Prohlížeče

Hodnocení hlavního menu a sekce O Aplikaci

Jednomu respondentovi se zdálo hlavní menu průměrné, sedm respondentů jej pak hodnotilo jako skvělé. Nadprůměrně byla hodnocena také sekce O Aplikaci. Grafy k této kategorii se nachází na konci práce jako vázaná příloha 2.

8 VÝSLEDKY

Pro splnění hlavního cíle této bakalářské práce, tedy sestavit univerzální pasportizační klient pro mobilní platformu Android bylo nutné splnit další dílčí cíle.

V teoretické části práce se jednalo o:

- Rešerši odborné literatury
- Termíny pasport a pasportizace
- Rešerši stávajících mobilních mapových klientů
- Specifikaci vývojových postupů a metod vývoje softwaru pro mobilní zařízení
- Definici aspektů tvorby mobilních mapových aplikací
- Specifikace zásad UX/UI mobilních řešení

V praktické části práce byly stanoveny tyto cíle:

- Navrhnout a vyvinout mobilní pasportizační aplikaci
- Provést výkonnostní a uživatelské testování aplikace

Velká část rešerše odborné literatury proběhla na samotném počátku tvorby této bakalářské práce. Nejprve bylo nutné nastudovat literaturu týkající se platformy Android a programovacího jazyka Java. Tyto znalosti musely být pro potřebu tvorby mobilní mapové aplikace procvičovány, a aby se tvorba aplikace podařila, muselo být dosaženo jejich určité úrovně. S objevováním dalších užitečných zdrojů v průběhu psaní práce probíhalo další doplňování znalostí.

Dalším dílčím cílem, který byl stanoven, bylo popsat čtenářům termíny pasport a pasportizace. Pasportizaci si autor této práce vyzkoušel na povinné praxi ve třetím ročníku studia a následně i v zaměstnání. Proto byla podkapitola 3.1 velmi obsáhlá, plná informací z autorových zkušeností a byla také doplněna o vhodné fotografie z terénních měření. Byla také provedena rešerše pasportizace v zákonech České republiky.

Následně byl proveden průzkum trhu – rešerše stávajících mobilních mapových klientů. Během řešení podkapitoly 3.4 byly tyto aplikace vyhledávány na Internetu – na internetových fórech a v obchodě Google Play. Po pečlivém výběru osmi aplikací byly tyto nainstalovány do autorova zařízení, byly podrobeny testování a poté byly autorem této práce popsány a ohodnoceny. V této podkapitole bylo také popsáno několik druhů a možností webových řešení.

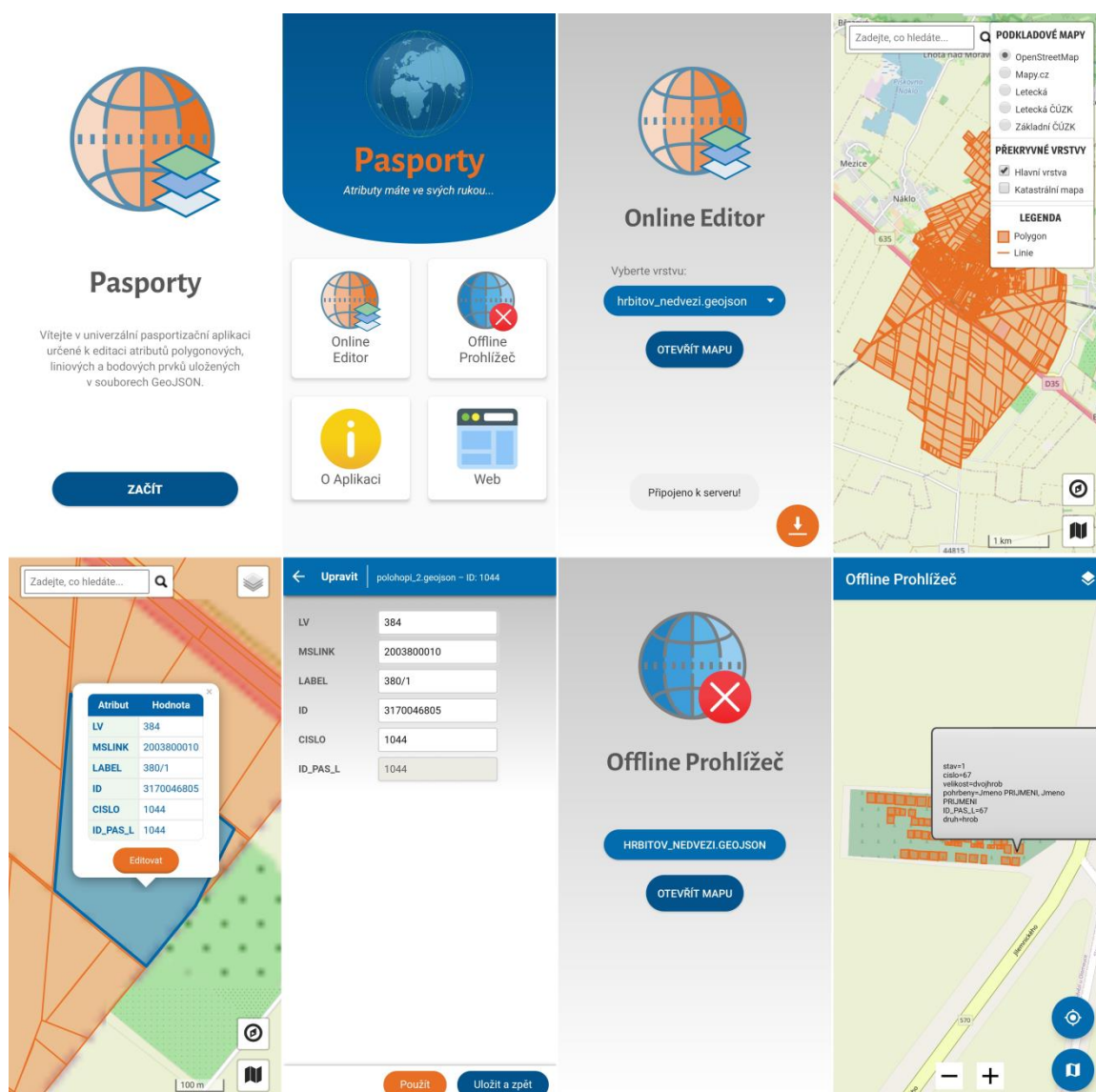
Poté byla autorem na základě nastudované literatury a dalších zdrojů v podkapitole 3.6 provedena specifikace vývojových postupů a metod vývoje softwaru pro mobilní zařízení. Byly charakterizovány tři současné vývojové postupy tvorby mobilních aplikací: Nativní, Hybridní a PWA. Co se týče metod vývoje softwaru, tak zde byly popsány dva tradiční teoretické přístupy: Vodopádový model a Agilní model vývoje softwaru. V neposlední řadě zde byly vysvětleny důležité pojmy týkající se vývoje softwaru jako např. životní cyklus softwaru, softwarové prototypování, iterace, či verzování softwaru.

V následující kapitole 4 byla představena autorova motivace k tvorbě mobilní mapové pasportizační aplikace. Motivace byla podepřena autorem dosud zjištěnými skutečnostmi v předcházejících kapitolách. Zde bylo také zjištěno, že je nezbytné

před samotným návrhem a tvorbou takové mapové aplikace definovat aspekty tvorby mobilních mapových aplikací.

Toto bylo provedeno v poslední kapitole teoretické části bakalářské práce, v kapitole 5. Zde byly aspekty na základě nastudované literatury vybrány od autorů Vondrákové (2014) a Nětka (2015) a byly uzpůsobeny tvorbě mobilní mapové aplikace pro systém Android a byl tak splněn další cíl této práce. Aspekty byly zanalyzovány, popsány a definovány mnohdy i s příklady tak, aby byly pro čtenáře co nejpochoptelnější. V podkapitole 5.2 bylo také nutné popsat zásady UX (User Experience) a UI (User Interface) mobilních řešení. Oba tyto termíny byly charakterizovány a bylo provedeno jejich porovnání.

Poté bylo přistoupeno k praktické části této bakalářské práce, k samotné pilotní studii – návrhu a tvorbě mobilního pasportizačního klienta pro mobilní platformu Android. Na základě autorových poznatků, průzkumu trhu, motivace a definovaných aspektů a zásad tvorby mobilních aplikací byl navržen a ve vývojovém prostředí Android Studio naprogramován univerzální mobilní pasportizační klient s názvem *Pasporty*.



Obr. 35 Koláž ze snímků obrazovky z aplikace Pasporty

Tato aplikace je aplikací hybridní, což znamená, že sestává z webové části – Online Editoru a nativní části – Offline Prohlížeče. Aplikace byla naprogramována v jazycích Java, XML, HTML5, PHP a JavaScript. Aplikace nabízí možnost prohlížet pasporty ve formátu GeoJSON a editovat jejich atributy v uživatelsky přívětivém prostředí založeném na open-source řešeních jako jsou například knihovny Leaflet nebo OSMDroid. Aplikace umožňuje uživateli online a offline přístup k datům.

Aplikace byla v průběhu vývoje podrobována výkonnostnímu testování (viz podkapitola 7.1), během kterého byly laděny chyby, což korespondovalo se zásadami agilního vývoje softwaru. Na jeho základě byla vytvořena finální verze aplikace *Pasporty*. Celkem bylo postupně vytvořeno 43 testovacích verzí aplikace, které byly testovány na pěti různých zařízeních. Nakonec bylo provedeno uživatelské testování (viz podkapitola 7.2) a to ve formě dotazníkového šetření, kde byl osloven vzorek jak odborníků (geoinformatiků a kartografů), tak laiků. Tito měli možnost aplikaci otestovat a v dotazníku zodpověděli na otázky, v nichž bodově hodnotili jednotlivé části aplikace. Následně byli respondenti požádáni o napsání subjektivní recenze. Některé z těchto recenzí byly uvedeny do textu práce. Samozřejmostí také je, že byly autorem popsány a charakterizovány samotné pojmy uživatelské a výkonnostní testování a byly vypsány jejich zásady.

Co se týče výsledků uživatelského testování, tak aplikace zde byla hodnocena velmi dobře a byla všemi zúčastněnými doporučována. Respondenty byl vyzdvižen jak už samotný nápad, tak i design aplikace, její funkce a celkově téma této bakalářské práce. Grafy s výsledky uživatelského testování a další podrobnosti se nachází v podkapitole 7.2.

9 DISKUSE

Pasporty jsou důležitými dokumenty úzce spjatými s informačními systémy o území a jsou využívány všemi obcemi v České republice. Některé pasporty jsou totiž ze zákona povinné. Papírové pasporty jsou naštěstí postupně vytlačeny těmi, které jsou vytvořeny digitální formou. Z tohoto důvodu existuje mnoho společností, které se jejich tvorbou zabývají. Zhotovení pasportu je finančně a technologicky náročné a jeho aktualizace v případě změny taktéž.

Existuje mnoho mobilních mapových aplikací, v nichž by se pasport dal vytvořit či upravit, některé jsou ovšem příliš podrobné a složité na ovládání a některé zase nenabízejí dostatečné množství funkcí pro jeho správu. Z tohoto důvodu byl naprogramován univerzální pasportizační klient pro platformu Android s názvem Pasporty, který je na tuto problematiku specializován a je určen pro typickou cílovou skupinu – starosty malých obcí, které nemají dostatečné finanční náklady na aktualizaci pasportu. Aplikace umožňuje v reálném čase pasport prohlížet a editovat jeho atributy.

I přes to, že autor neměl s programováním aplikací pro mobilní zařízení žádnou zkušenost je vytvořená aplikace více než použitelná a byla ohodnocena testery nadprůměrně až skvěle. Je ale nutno poznamenat, že během tvorby se objevilo několik problémů. Prvním z nich je skutečnost, že z velkého množství zařízení, na nichž byla aplikace zkoušena nebo testována, v případě dvou zařízení aplikace nefungovala. Bylo vytvořeno 43 testovacích verzí aplikace za účelem optimalizace, ale i tak na tomto malém zlomku ze všech zařízení, která by měla splňovat hardwarové a softwarové požadavky, aplikace nefungovala. Tento fakt jen podtrhuje důležitost celého vývojového procesu nejenom tvorby mobilních aplikací. Tento proces je obrovsky komplexní, zdoluhavý a finančně náročný.

Dalším problémem, který v průběhu tvorby vyvstal, se týká editace pasportu více uživateli zároveň. Tuto funkci bohužel aplikace nenabízí. Probíhá zde totiž editace celé mapové vrstvy, a nikoliv jen jednoho z prvků. Hlavním faktorem, který tento nedostatek způsobil, je fakt, že autor neměl a doteď nemá zkušenosti s tvorbou databází v mobilních zařízeních, které by víceuživatelskou editaci umožňovaly. Řešení, které autor v této práci zvolil, bylo pro nezkušeného tvůrce, jednodušší.

Posledním zmíněným problémem je nemožnost editace geometrie jednotlivých mapových prvků. Tento nedostatek byl vysvětlen a obhájen v návrhu aplikace v podkapitole 6.1.

Z těchto tří důvodů a kvůli dalším, menším problémům nebyla aplikace publikována na obchodu Google Play, ale je dostupná pouze jako instalační balíček APK. Celkový přínos této bakalářské práce a samotného návrhu tvorby univerzální pasportizační aplikace je podle autora značný a tato práce může dále sloužit jako inspirace a návod pro potenciální tvůrce podobných mobilních mapových aplikací.

Hybridní řešení tvorby aplikace je moderním přístupem a s nástupem technologií jako například Apache Cordova se blížíme do doby, kdy bude tvorba aplikací více a více jednodušší a dostupnější. Díky těmto technologiím je už dnes možné vytvářet kvalitní multiplatformní aplikace, ovšem podle autora této bakalářské práce se stále zatím jedná o hudbu budoucnosti.

10 ZÁVĚR

Hlavním cílem této bakalářské práce bylo sestavit univerzální pasportizační klient pro mobilní platformu Android. Veškeré dílčí cíle práce byly stejně jako cíl hlavní splněny a uvedeny v textu práce.

Co se dílčích cílů práce týče, nejprve byla provedena obsáhlá rešerše a byly definovány termíny pasport a pasportizace. Pro načerpání inspirace byl také proveden průzkum a analýza současného trhu mobilních mapových aplikací.

V teoretické části této práce byly také specifikovány vývojové postupy a metody vývoje softwaru pro mobilní zařízení, stejně tak jako zde byly uvedeny zásady UX/UI mobilních řešení. Posledním teoretickým cílem bylo definovat aspekty tvorby mobilních mapových aplikací. Tento cíl práce byl taktéž splněn.

Hlavní cíl práce – sestavit univerzální pasportizační klient pro mobilní platformu Android – byl, na základě rešerše, průzkumu trhu a dalších vyzkoumaných a specifikovaných zásad a aspektů, splněn a byla vytvořena mobilní mapová aplikace *Pasporty*. Tato aplikace byla vytvořena hybridním přístupem ve vývojových prostředích Android Studio a Visual Studio Code a naprogramována je v jazycích Java, XML, HTML, PHP a JavaScript. Aplikace umožňuje uživatelům zobrazit, prohlížet a jednoduše editovat atributy polygonových, liniových a bodových vrstev rozličných pasportů v reálném čase online i offline.

Posledním cílem práce bylo aplikaci podrobit výkonnostnímu a uživatelskému testování, které byly, s použitím testovacích dat dodaných vedoucím práce, úspěšně provedeny a byly tak naplněny všechny cíle této bakalářské práce. Aplikace *Pasporty* je dostupná ke stažení na webu k této bakalářské práci a nachází se i jako příloha na CD.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

ALLEN, Grant. *Android 4: průvodce programováním mobilních aplikací*. Brno: Computer Press, 2013. ISBN 978-80-251-3782-6.

AVERSEANTE, Fabio. *Native, Hybrid and Progressive Web Applications — Building a Mobile App Today* [online]. 2019 [cit. 2020-08-02]. Medium. Dostupné z WWW: <<https://medium.com/twodigits/native-hybrid-and-progressive-web-applications-building-a-mobile-app-today-db076642eb40>>.

ČESKO. Zákon č. 563 ze dne 31. prosince 1991 o účetnictví. In: *Sbírka zákonů České republiky*. 1991, částka 107, Dostupný také z: <<https://www.zakonyprolidi.cz/cs/1991-563#p29>>.

ČESKO. Vyhláška č. 104 ze dne 7. května 1997, kterou se provádí zákon o pozemních komunikacích. In: *Sbírka zákonů České republiky*. 1997, částka 36, Dostupný také z: <<https://www.zakonyprolidi.cz/cs/1997-104#p5>>.

ČESKO. Zákon č. 114 ze dne 25. března 1992 o ochraně přírody a krajiny. In: *Sbírka zákonů České republiky*. 1992, částka 28, Dostupný také z: <<https://www.zakonyprolidi.cz/cs/1992-114#p63>>.

ČESKO. Zákon č. 183 ze dne 11. května 2006 o územním plánování a stavebním řádu. In: *Sbírka zákonů České republiky*. 2006, částka 63, Dostupný také z: <<https://www.zakonyprolidi.cz/cs/2006-183#p161>>.

ČORIČ, Oskar. *Rozdíl mezi UI a UX* [online]. 2017 [cit. 2020-08-02]. Oskar Čorič. Dostupné z WWW: <<https://oskarcoric.cz/rozdil-mezi-ui-a-ux/>>.

HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK. *Big Data a NoSQL databáze*. Praha: Grada, 2015. Profesionál. ISBN 978-80-247-5466-6.

CHYTIL, Jiří. *Teorie programování – Verze softwaru* [online]. 2005 [cit. 2020-08-02]. Programujte.com. Dostupné z WWW: <<http://programujte.com/clanek/2005070702-teorie-programovani-verze-softwaru/>>.

MARTINÁK, Lukáš. *Výkonnostní testování webových aplikací* [online]. Brno, 2012 [cit. 2020-08-02]. Dostupné z WWW: <<http://hdl.handle.net/11012/53719>>. Diplomová práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav informačních systémů. Vedoucí práce Radek Burget.

NÉTEK, Rostislav. *Rich Internet Application pro podporu rozhodovacích procesů Integrovaného záchranného systému*. Olomouc: Univerzita Palackého v Olomouci pro Katedru geoinformatiky, 2015. Terra notitia. ISBN 978-80-244-4805-3.

NĚTEK, Rostislav a Tomáš BURIAN. *Free and open source v geoinformatice*. Olomouc: Univerzita Palackého, 2018. ISBN 978-80-244-5291-3.

NÝVLT, David. *Prototypování při vývoji softwaru* [online]. Praha, 2011 [cit. 2020-08-02]. Dostupné z: <<https://theses.cz/id/pkpf55/>>. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Alena Buchalcevoová.

PATTON, Ron. *Testování softwaru*. Praha: Computer Press, 2002. Programování. ISBN 80-722-6636-5.

RUSSELL, Alex. *Progressive Web Apps: Escaping Tabs Without Losing Our Soul* [online]. 2015 [cit. 2020-08-02]. Infrequently Noted. Dostupné z WWW: <<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>>.

ŠIMŮNEK, David. *Jaký je rozdíl mezi Waterfall a Agile přístupem* [online]. 2019 [cit. 2020-08-02]. David Šimůnek. Dostupné z WWW: <<https://www.davidsimunek.com/post/jaky-je-rozdil-mezi-waterfall-a-agile>>.

VÁCLAVÍK, Jan. *JAK VYVÍJET MOBILNÍ APLIKACE* [online]. 2015 [cit. 2020-08-02]. Jan Václavík. Dostupné z WWW: <<http://janvaclavik.cz/jak-vyvijet-mobilni-aplikace/>>.

VONDRÁKOVÁ, Alena. *Netechnologické aspekty mapové tvorby*. Olomouc: Univerzita Palackého v Olomouci pro katedru geoinformatiky, 2014. Terra notitia. ISBN 978-80-244-3970-9

VOŽENÍLEK, Vít a Jaromír KAŇOK. *Metody tematické kartografie: vizualizace prostorových jevů*. Olomouc: Univerzita Palackého v Olomouci pro katedru geoinformatiky, 2011. ISBN 978-80-244-2790-4.

Android Studio [online]. [cit. 2020-08-02]. Android Developers. Dostupné z WWW: <<https://developer.android.com/studio>>.

Building web apps in WebView [online]. [cit. 2020-08-02]. Android Developers. Dostupné z WWW: <<https://developer.android.com/guide/webapps/webview>>.

Co je UX a UI design [online]. [cit. 2020-08-02]. bpromotion. Dostupné z WWW: <<http://cojeuxui.cz/>>.

HTML Living Standard [online]. 2020 [cit. 2020-08-02]. Web Hypertext Application Technology Working Group. Dostupné z WWW: <<https://html.spec.whatwg.org/>>.

Mobile software development lifecycle [online]. 2016 [cit. 2020-08-02]. Microsoft Docs. Dostupné z WWW: <<https://docs.microsoft.com/cs-cz/xamarin/cross-platform/get-started/introduction-to-mobile-sdlc>>.

RFC 7946 – The GeoJSON Format [online]. 2016 [cit. 2020-08-02]. Internet Engineering Task Force (IETF). Dostupné z WWW: <<https://tools.ietf.org/html/rfc7946>>.

Školení Open Source GIS [online]. 2017 [cit. 2020-08-02]. GISMentors. Dostupné z WWW: <<https://training.gismentors.eu/open-source-gis/formaty/vektor.html>>.

Usage statistics of server-side programming languages for websites [online]. 2020 [cit. 2020-08-02]. W3Techs. Dostupné z WWW: <https://w3techs.com/technologies/overview/programming_language>.

Uživatelské testování s instančním návrhem zlepšení [online]. [cit. 2020-08-02]. Sherpas. Dostupné z WWW: <<https://sluzby.sherpas.cz/testovani>>.

PŘÍLOHY

SEZNAM PŘÍLOH

Vázané přílohy

- Příloha 1 Test existujících mobilních mapových aplikací
Příloha 2 Grafy k podkapitole 7.2.2
Příloha 3 Subjektivní recenze mobilní mapové aplikace Pasporty

Volné přílohy

- Příloha 4 Poster
Příloha 5 CD

Elektronické přílohy

- Příloha 6 Aplikace Pasporty *(na CD a na webu)*
Příloha 7 Zdrojové kódy *(na CD a na webu)* a záloha aplikace *(na CD)*

Popis struktury CD

Adresář	Podadresář	Soubor
Aplikace	-	Pasporty.apk
Data	-	TestovaciData1.zip
	-	TestovaciData2.zip
Prilohy	-	Poster.pdf
TextPrace	-	Roller_BP.docx
	-	Roller_BP.pdf
Web	-	Roller20.zip
Zaloha_A_ZdrojoveKody	-	ZdrojoveKody.pdf
	NativniCast	Pasporty.zip
	WebovaCast	pasporty.zip

Příloha 1

Test existujících mobilních mapových aplikací

Locus Map

Základní parametry

Aktuální verze: 3.44.3

Autor: Asamm Software, s. r. o.

Licence: Proprietární software; zdarma a Pro



1

Podporované formáty

Vektor: KML, KMZ, GPX, GeoJSON, LOC a další

Rastr: TAR, MBTiles, rastry v SQLite databázi, IMG, GEMF a další

Databáze: SQLite

Webové služby: WMS, MBTiles

Obecné informace

Autor uvádí, že se jedná o multifunkční mobilní outdoorovou navigační aplikaci pro turistiku, cyklistiku, geocaching, plavbu, létání a další sporty a cestování. Aplikace umožňuje import a export (např. cyklistických) tras v různých formátech; jejich sdílení, import a export POI bodů, import geocachek atd. Aplikace nabízí i navigaci.

Klady

Velké množství funkcí. Další funkce lze doinstalovat. Schopnost běžet offline – včetně offline map a české hlasové a zvukové navigace. Pro „zkušeného“ cyklistu či turistu skvělá volba.

Zápory

Aplikace je celkem složitá na ovládání. Poměrně špatná práce s některými formáty dat (např. GeoJSON – pouze import; někdy dokonce soubor vůbec nelze otevřít).

¹ Ikona aplikace Locus Map (zdroj: <https://play.google.com/>)

Locus GIS

Základní parametry

Aktuální verze: 1.4.1
Autor: Asamm Software, s. r. o.
Licence: Proprietární software; zdarma a placená



2

Podporované formáty

Vektor: SHP, GPX, XML
Rastr: –
Databáze: umožňuje import a export dat na/z Dropbox a Google Drive
Webové služby: WMS

Obecné informace

Profesionální GIS aplikace umožňující sbírat a zpracovávat geografická data přímo v terénu. Aplikace disponuje velkým množstvím funkcí. Dobře spolupracuje s externím profesionálním GPS přijímačem. K jednotlivým entitám lze připojit fotografie. Je možné používat offline mapy v různých formátech. Aplikace spolupracuje s Dropboxem a Google Drive. Má jednoduché a intuitivní ovládání.

Klady

Aplikace umožňuje přímý export do aplikace QGIS. Nabízí velké množství funkcí. Má jednoduché ovládání.

Zápory

Pracuje pouze se soubory Shapefile a CSV. Verze zdarma je omezená počtem vrstev.

Gisella

Základní parametry

Aktuální verze: 1.1.6
Autor: ENVIPARTNER, s.r.o.
Licence: Proprietární software; zdarma a Pro



3

Podporované formáty

Vektor: SHP, KML, GeoJSON
Rastr: –
Databáze: umožňuje import a export dat na/z Dropbox a Google Drive
Webové služby: –

² Ikona aplikace Locus GIS (zdroj: <https://play.google.com/>)

³ Ikona aplikace Gisella (zdroj: <https://play.google.com/>)

Obecné informace

Gisella je mobilní GIS aplikace umožňující vytvářet a spravovat všechny geografické objekty přímo v mobilním zařízení. Nabízí vše od správy mapových objektů přes vrstvy až po celé mapové projekty.

Klady

Jednoduché a intuitivní ovládání. Kvalitní zákaznická podpora včetně oficiálních videonávodů. Aplikace spolupracuje s Google Drive a Dropboxem. Možnost připojit fotografii, zvuk či video k objektům na mapě.

Zápory

Free verze je omezena počtem prvků na vrstvu (50 prvků/vrstva). Export vrstev je ve verzi zdarma také omezen, a to na export do KML. Aplikace nenabízí offline mapy.

MapTiler Mobile

Základní parametry

<i>Aktuální verze:</i>	2.0.1
<i>Autor:</i>	Klokant Technologies GmbH
<i>Licence:</i>	Proprietární software; zdarma a placená



4

Podporované formáty

<i>Vektor:</i>	GeoJSON, SHP, DXF, DWG, GPX, CSV, KML
<i>Rastr:</i>	GeoTIFF, TIFF, JPEG, DEM, WebF, JP2, JPEG2000, Erdas, Grass, Sentinel2, SRTM a další
<i>Databáze:</i>	používá vlastní cloud; možnost exportu na GoogleDrive
<i>Webové služby:</i>	WMTS, TileJSON, XYZ tiles, OpenMapTiles

Obecné informace

Mobilní mapová aplikace, která mimo sběru dat nabízí mimo jiné i možnost nahrání vlastní rastrové vrstvy v různých formátech.

Klady

Aplikace je rychlá. Obsahuje velkou škálu podkladových map. Možnost nahrání vlastního georeferencovaného obrázku buď ve formě rastrového souboru, nebo ve formě odkazu na webové služby. Export nasbíraných dat a prvků ve formátu GeoJSON na Google Drive.

Zápory

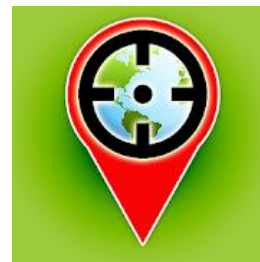
U verze zdarma je omezen import GeoJSONu na počet prvků. Pro spoustu úkonů je potřeba se zaregistrovat a zaplatit si placenou verzi.

⁴ Ikona aplikace MapTiler Mobile (zdroj: <https://play.google.com/>)

Mapit GIS

Základní parametry

Aktuální verze: 7.2.5Core
Autor: Mapit GIS LTD
Licence: Proprietární software; zdarma a placená



5

Podporované formáty

Vektor: SHP, CSV, KML, GeoJSON, DXF (export), TomTom OV2 (export)
Rastr: MBTiles
Databáze: –
Webové služby: WMS, MBTiles, Arc GIS Server Tiled Services

Obecné informace

Mapit GIS je aplikace navržená pro sběr dat v terénu pomocí GPS. Aplikace může být použita pro měření výpočty plochy nebo vzdálenosti. Data mohou být organizována do vrstev a exportována do běžných formátů.

Klady

Jednoduchá, ale výkonná aplikace. Spolupracuje s externími GPS přijímači. Verze zdarma je velmi obsáhlá.

Zápory

Občas dojde ke ztrátě dat během ukládání a občas aplikace běží pomaleji, než by měla.

Collector for ArcGIS

Základní parametry

Aktuální verze: 19.1.0
Autor: Esri
Licence: Proprietární software;



6

Podporované formáty

Vektor: SHP, GPX, CSV; mapové služby z ArcGIS Online
Rastr: mapové služby z ArcGIS Online
Databáze: GDB (Geodatabase)
Webové služby: mapové služby z ArcGIS Online

⁵ Ikona aplikace Mapit GIS (zdroj: <https://play.google.com/>)

⁶ Ikona aplikace Collector for ArcGIS (zdroj: <https://play.google.com/>)

Obecné informace

Collector for ArcGIS je aplikace pro sběr dat v terénu od společnosti Esri. Aplikace umožňuje stahovat mapy a pracovat s nimi offline nebo připojit k prvkům fotografie. Lze použít externí GPS přijímače.

Klady

Propojení se službami Esri a integrace s dalšími aplikacemi a službami od této společnosti.

Zápory

Vyžaduje přihlášení k účtu ArcGIS Online, to ovšem málokdy funguje. Aplikace se seká a pomalu se načítá, občas dokonce zamrzne úplně. Aplikace obsahuje hodně chyb.

SW Maps – GIS & Data Collector

Základní parametry

Aktuální verze: 2.4.8
Autor: Softwel
Licence: Freeware; Proprietární software



7

Podporované formáty

Vektor: SHP, KMZ, GeoJSON, GeoPackage
Rastr: MBTiles
Databáze: –
Webové služby: WMS, WMTS, TMS, XYZ, MBTiles

Obecné informace

SW Maps je freewarová geoinformační mobilní aplikace pro sběr, prezentaci a sdílení geografických dat. Podporuje externí GPS přijímače. Zajímavostí je, že je aplikace vyvinuta Nepálskými vývojáři a je publikována jako freeware. V aplikaci se neobjevují ani reklamy.

Klady

Jednoduchá aplikace disponující nezbytnými funkcemi pro sběr dat v terénu.

Zápory

Poměrně komplikované nahrávání a import externích souborů. Editace atributů je také složitá. Nelze připojit fotografii k nasbíraným datům.

⁷ Ikona aplikace SW Maps – GIS & Data Collector (zdroj: <https://play.google.com/>)

QField pro QGIS

Základní parametry

Aktuální verze: 1.2.0 - Matterhorn

Autor: OPENGIS.ch

Licence: Obecná veřejná licence GNU v. 2 (GNU GPL v2)



8

Podporované formáty

Vektor: SHP, GeoPackage

Rastr: GeoPackage, MBTiles, (TIFF)

Databáze: Spatialite, Postgis

Webové služby: WMS, WFS, WFS-T, MBTiles

Obecné informace

QField se zaměřuje na efektivní práci v terénu s GIS daty. Kombinuje minimalistický design se sofistikovanou technologií, díky níž lze pohodlně a jednoduše dostat data z terénu až do kanceláře. Jedná se o aplikaci s otevřeným zdrojovým kódem dostupným pod licencí GNU GPL v2.

Klady

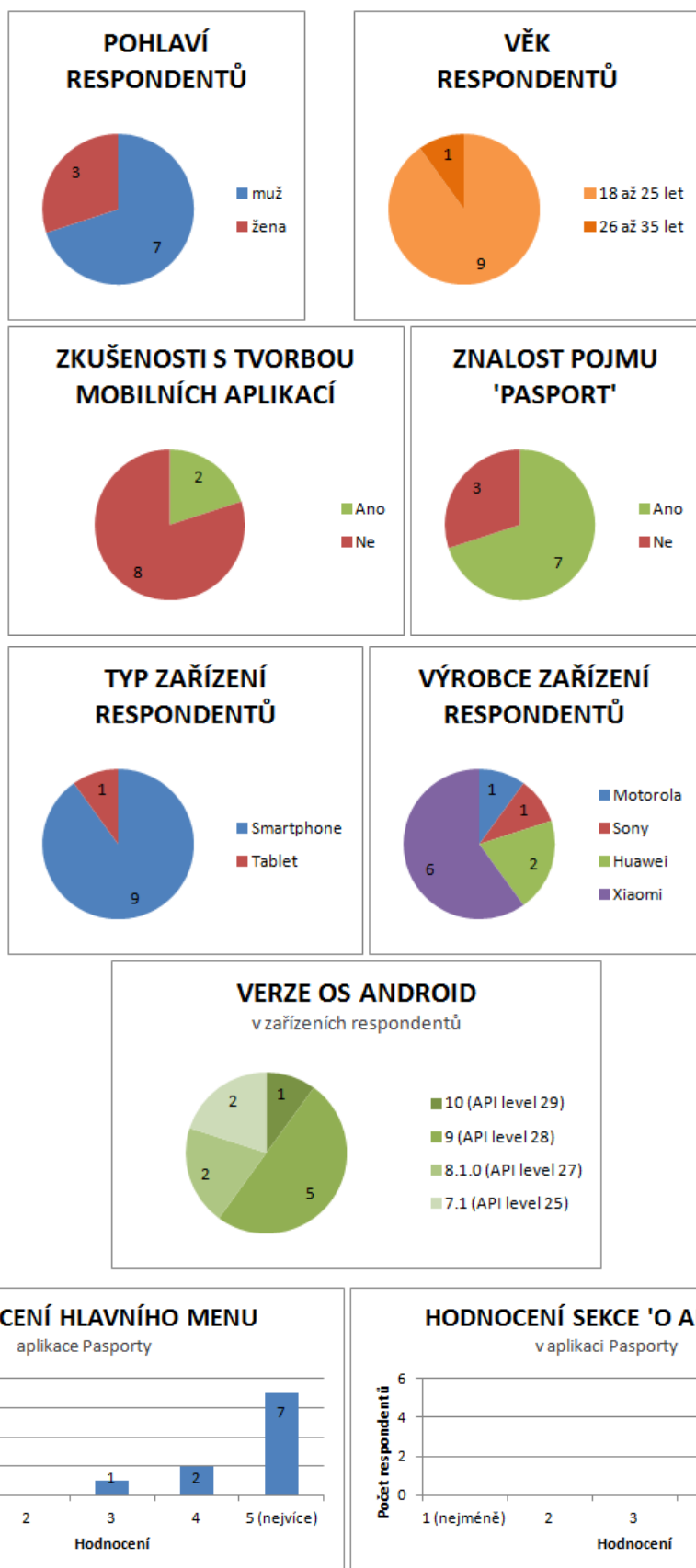
Aplikace je velmi podrobná. Obsahuje velké množství funkcí. Běží velmi rychle a neseká se.

Zápory

Aplikace je celkem složitá na ovládání. Složitá je i editace atributů a geometrie.

⁸ Ikona aplikace QField pro QGIS (zdroj: <https://play.google.com/>)

Grafy k podkapitole 7.2.2



Příloha 3

Subjektivní recenze mobilní mapové aplikace Pasporty

„Co se UI a UX týče, prostor pro zlepšení je velký, to ale není účel práce. Z geoinformatického a geografického hlediska toho moc posoudit nedokážu a o pasportech slyším prvně. Rovněž nedokážu posoudit bezpečnost aplikace. Nicméně, aplikace vypadá přehledně a fungovala poměrně rychle. Na to, že autor nemá zkušenosti s UI, UX a programováním hodnotím velmi kladně.“

„Aplikace byla intuitivní a přehledná, což usnadňuje její použití bez jakékoli znalosti oboru.“

„Aplikace je intuitivní a vhodná i pro nezkušeného uživatele. Oceňuji širokou nabídku podkladových map a vzhled hlavního menu. Při použití na zařízení se slabším internetovým připojením bylo posouvání mapového pole v Online Editoru bohužel pomalé.“

„Aplikace je hezká, přehledná a poměrně rychlá. Nabízí všechny funkce, které jsou pro rychlou editaci atributů u pasportů důležité. Možná někomu bude chybět editace geometrie, ale jistě to má své důvody. Offline Prohlížečka je hezky zpracovaná, jen je načítání vrstvy trochu zdlouhavější při otevírání větší vrstvy, ale na to autor upozornil.“

„Kvalitní provedení, jednoduchost a účelnost a povedený design dělá tuto aplikaci více než nadprůměrnou.“

„Aplikace se mi zdá velmi vydařená. Vše potřebné je intuitivní a na vhodných místech, kde by je uživatel přesně čekal. Trochu mi chybí nějaké menu (například pokud jsem v Online Editoru pro navrácení zpět na hlavní stránku). Ovšem věřím, že v praxi by stejně člověk pracoval většinu času pouze v jedné z možností (online/offline). Pro tvorbu pasportů by měla být aplikace jako je tato základním nástrojem a svou rychlostí a jednoduchostí se mi zdá vhodnější než aplikace, které jsem měla šanci vyzkoušet. Škoda jen, že nelze editovat i geometrie.“

„Aj keď patrím k ľuďom mimo odbor geoinformatiky, aplikácia sa mi veľmi páčila.“