

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta



Interaktivní aplikace pro orientaci a navigaci v kampusu Jihočeské univerzity

Bakalářská práce

Pavel Beran

Školitel: Ing. Rudolf Vohnout, Ph.D.

České Budějovice 2020

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE





Student: <i>(jméno, příjmení, tituly)</i>	Pavel Beran
Obor – zaměření studia:	Aplikovaná informatika – Bezpečnost a sítě
Katedra/ústav PŘF JU:	Ústav aplikované informatiky
Školitel: <i>(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)</i>	Ing. Rudolf Vohnout, Ph.D.
Garant z PŘF JU: <i>(jméno, příjmení, tituly, katedra – jen v případě externího školitele)</i>	
Školitel – specialista, konzultant: <i>(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)</i>	Mgr. Věra Filipová; Mgr. Jakub Geyer
Téma bakalářské práce:	Interaktivní aplikace pro orientaci a navigaci v kampusu Jihočeské Univerzity

Cíle práce:

Cílem práce je návrh a realizace aplikace pro mobilní zařízení, která usnadní orientaci v kampusu Jihočeské Univerzity a umožní základní navigaci mezi budovami. V aplikaci budou dále dostupné informace o tom, co se v jednotlivých budovách nachází (místnosti, učebny, učitelé a základní informace o nich – email, číslo kanceláře, apod.). Tyto informace a poloha budou navíc dostupné i pro budovy mimo hlavní kampus. Jako zdroj dat by aplikace měla využívat vhodné dostupné univerzitní systémy, zejména informační systém STAG a OrgStructure. Součástí informací o jednotlivých budovách by mělo být rovněž několik fotografií objektu. Aplikace musí umožňovat jazykovou lokalizaci (v základu EN/CZ). Součástí práce budou bezpodmínečně následující dokumenty:

- *Zadávací dokumentace*
- *Analýza funkčních a nefunkčních požadavků*
- *Analýza existujících řešení*
- *Dokumentace zdrojových dat*
- *Dokumentace kódu*
- *Testovací dokumentace*
- *Uživatelská dokumentace*

Základní doporučená literatura:

Financování práce
Školitel práce podpis: 
U externích vedoucích fakultní garant práce podpis: 
Garant oboru bak. studia (nepožaduje se u oboru biologie) podpis: 
Vedoucí katedry/ústavu PřF JU, kde proběhne obhajoba podpis: 
Případný souhlas vedoucího ústavu AV podpis:

V Českých Budějovicích dne

Podpis studenta: 

Beran, P., 2020: Interaktivní aplikace pro orientaci a navigaci v kampusu Jihočeské univerzity. [Interactive application for orientation and navigation on the campus of the University of South Bohemia. Bc. Thesis, in Czech.] – 83 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Cílem práce je vytvořit mobilní aplikaci pro operační systém Android, která bude poskytovat pomocnou ruku při orientaci a pohybu studentů po kampusu Jihočeské univerzity, bude umožňovat uživatelům zjistit svoji polohu a zajistí následnou navigaci na vybrané místo. Aplikace poskytne informace o budovách univerzity a intuitivní vyhledávání těchto informací. Aplikace bude bilingvní. Tato bakalářská práce obsahuje analýzu, návrh, implementaci a testování vytvořené aplikace.

Klíčová slova

GPS, navigace, mapy, Android, OrqStructure, MapBox

Abstract

The goal of this thesis is to create an application for the Android operating system, which will provide a helping hand for the orientation and movement of students at the campus of South Bohemian University, will allow users to gain their location, navigate to selected locations, obtain information about university buildings and search for this information. The application will be bilingual. This bachelor thesis contains analysis, design, implementation and testing of this application.

Keywords

GPS, navigation, maps, Android, OrqStructure, MapBox

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury. Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích, dne 11. 05. 2020

Pavel Beran.

Poděkování

Rád bych poděkoval svému školiteli, Ing. Rudolfu Vohnoutovi, Ph.D., za odborné vedení a cenné rady. Také bych chtěl poděkovat své rodině za jejich podporu při studiu.

Obsah

1 Úvod.....	1
2 Analýza	3
2.1 Zadávací dokumentace	3
2.2 Cíle práce	4
2.3 Rozložení práce.....	4
2.4 Současné nástroje a stav na trhu	4
2.4.1 Část týkající se mapy	5
2.4.2 Část týkající se informací	7
2.4.3 Nedostatky dostupných aplikací.....	8
2.4.4 Přednosti dostupných aplikací.....	9
2.5 Zjištěné problémy se stavem informačních dat Jihočeské univerzity.....	9
2.6 Případy užití.....	12
2.7 Funkční požadavky	14
2.8 Nefunkční požadavky	15
3 Teoretická východiska	16
3.1 GPS	16
3.2 Trilaterace	18
3.3 Dijkstrův algoritmus	19
4 Návrh	22
4.1 Technologie	22

4.1.1 Platforma	22
4.1.2 Programovací jazyk	23
4.1.3 Databáze	24
4.1.4 Mapa	24
4.1.5 Navigace	26
4.2 Architektura systému	27
4.3 Architektura aplikace	27
4.4 Metodika vývoje a použité nástroje	29
4.5 Wireframes.....	30
4.6 Uživatelské rozhraní	34
4.7 Návrh vzhledu mapy	35
5 Implementace.....	36
5.1 Zásady vývoje	36
5.2 Vývojové prostředí	37
5.3 Správa kódu	37
5.4 Databáze.....	37
5.5 Mobilní aplikace	39
5.5.1 Adresářová struktura a popis tříd	39
5.5.2 Spinner dialog.....	42
5.5.3 Mapa	42
5.5.4 Poloha	43
5.5.5 Navigace	43

6 Testování.....	45
7 Distribuce.....	47
8 Diskuse.....	48
9 Závěr.....	51
Seznam použité literatury	52
Seznam obrázků.....	55
Seznam tabulek.....	55
Seznam příloh	56
Příloha: Wireframes.....	57
Příloha: Uživatelská dokumentace.....	62
Příloha: Administrátorská dokumentace.....	80

1 Úvod

Bakalářská práce se zabývá tvorbou mobilní bilingvní aplikace, která usnadní orientaci a pohyb osob v kampusu Jihočeské univerzity. Cílovou skupinou pro užívání této aplikace jsou především studenti prvních ročníků a zahraniční studenti. Na začátku studia může být pro tyto osoby orientace v kampusu relativně komplikovaná. Tato aplikace by jim s tímto problémem měla pomoci. Zároveň by tím odpadla nutnost pořádání tzv. „kampus tour“¹ pro seznámení s univerzitním areálem.

V současné době je k dispozici několik volně dostupných řešení, která jsou sice obsáhlá z pohledu funkcionalit, ovšem obsahují několik překážek a nedostatků pro pohodlné používání v cílové skupině potenciálních uživatelů. Novou aplikaci lze rozdělit na dvě části.

První část obsahuje mapu a funkce s tím spojené. Mezi tyto funkce patří například zobrazení samotné mapy řešené oblasti, získání a vykreslení GPS polohy uživatele, respektive mobilního zařízení, nebo navigaci z místa A na místo B. Pokud se podíváme na řešení čistě tohoto problému, máme k dispozici především aplikace Google Maps² a lokální české Mapy.cz³. Obě tyto aplikace jsou velmi kvalitně podporovány a obsahují námi požadované funkce (mapa, získání polohy, navigace). Nicméně obsahují nedostatky, které vychází z jejich účelu. Pro naše použití jsou příliš globální. Například námi řešená oblast není dostatečně zvýrazněna, nebo neobsahuje navigaci mezi budovami.

¹Kampus tour je označení pro ukázkou a provedení studentů po areálu Jihočeské univerzity na začátku studia.

²Aplikace Google Maps je dostupná v obchodě Google Play.

³Aplikace Mapy.cz je dostupná v obchodě Google Play.

Druhá část aplikace se zabývá informacemi o objektech v řešené oblasti. Jedná se o informace, jakými jsou názvy budov, místnosti, zaměstnanci v budovách a podobně. K řešení tohoto problému není dostupná žádná mobilní aplikace.

Existují pouze webová řešení. Těmi jsou OrqStructure⁴, potažmo portál IS/STAG⁵. Bohužel tyto stránky nejsou pro mobilní použití příliš vhodné. Jejich vzhled není upraven pro mobilní telefony. Z toho vyplývá, že návštěva těchto stránek prostřednictvím mobilního zařízení není vyloženě „user-friendly“. V kontextu této práce je mobilní zařízení označení výhradně pro smartphone nebo tablet.

V tuto chvíli je tedy proces vyhledávání informací a následného lokalizování hledaného místa na mapě opravdu velice komplikovaný.

Nová aplikace míří na mobilní platformu s operačním systémem Android. Tato volba byla učiněna především ze dvou následujících důvodů. Prvním jsou výhody nativního vývoje. Tyto výhody jsou popsány v dalších kapitolách. Druhým důvodem je momentální rozložení trhu. Pokud se podíváme na situaci mobilních operačních systémů na trhu v České republice, zjistíme, že v prosinci 2019 držel operační systém Android podíl 79,2 %. Druhý iOS měl 19,99 % [1].

Co se týče mapových podkladů, lokalizace a navigace, bylo pro zpracování vybráno Maps SDK, respektive Navigation SDK od MapBox⁶. Výhodou je, že se jedná o open source obsahující velké množství možností a funkcí, které budou v aplikaci využity (například možnost upravit si vzhled mapy dle potřeb, nebo stažení určité oblasti pro použití v situaci, kdy je uživatel offline).

⁴ Organizační struktura JCU dostupná na <https://orgstr.jcu.cz>

⁵ Portál IS/STAG dostupný na <https://wstag.jcu.cz/portal>

⁶ MapBox dostupný na <https://www.mapbox.com/>

2 Analýza

Tato část práce se zaměřuje na cíle práce a na analýzu současného stavu na trhu. Zhodnotí dostupná řešení z hlediska kladných i záporných vlastností a uživatelské přívětivosti. Dále se zabývá analýzou funkčních a nefunkčních požadavků pro vyvíjenou aplikaci. Následně jsou definovány případy užití aplikace.

2.1 Zadávací dokumentace

Zadání pro vývoj aplikace vychází z diskuse s Mgr. Věrou Filipovou z útvaru pro zahraniční vztahy Jihočeské univerzity. Z této diskuse vzešly jednotlivé požadavky na hlavní funkcionality aplikace.

Požadavky:

- Mobilní aplikace bude vyvíjena pro zařízení používající OS Android.
- Aplikace bude obsahovat mapu kampusu Jihočeské univerzity.
- Vykreslení polohy uživatele.
- Navigace.
- Jazyková lokalizace EN/CZ.
- Dostupné informace o budovách (místnosti, učebny, zaměstnanci).
- Tyto informace budou dostupné i pro budovy mimo kampus.
- Výchozí zdroj dat bude organizační struktura OrqStructure a informační systém STAG.
- Součástí informací o budovách bude charakterizující fotografie objektu.

2.2 Cíle práce

Z požadavků v zadávací dokumentaci vychází následující cíle práce.

Hlavním cílem této práce je vytvoření mobilní aplikace zaměřené na operační systém Android, která bude nabízet outdoor lokalizaci a navigaci nejenom v areálu Jihočeské univerzity. Dále bude aplikace nabízet přístup k informacím o zaměstnancích Jihočeské univerzity, budovách, místnostech a vyhledávání těchto informací. Díky tomu, na jaké uživatele je aplikace cílena, bude aplikace podporovat vícejazyčnost. Konkrétně anglický a český jazyk. Celý návrh bude koncipován tak, aby v budoucnu nebyl problém přidat další jazyk.

Práce by měla také obsahovat analýzu dostupných řešení a vytvoření zdrojových dat, nesoucích informace o budovách a zaměstnancích.

2.3 Rozložení práce

Výstupem této práce bude jedna třetina zamýšleného navigačního systému Jihočeské univerzity. Výsledný navigační systém bude složen z těchto částí:

1. Indoor lokalizace, kterou vytváří David Langer.
2. Indoor navigace vyvíjena Michalem Bartošem.
3. Outdoor lokalizace a navigace, tvorba zdrojových dat o areálu Jihočeské univerzity, respektive jejích jednotlivých objektech a osobách.

2.4 Současné nástroje a stav na trhu

Momentálně není dostupné žádné řešení, které by pokrývalo zároveň obě hlavní funkcionality nové aplikace. Pokud se podíváme na věc odděleně, zjistíme, že je k dispozici několik možností.

Při analýze trhu, byla vybrána dvě dostupná řešení pro každou funkcionalitu, která budou následně popsána. Nejprve se autor zaměří na mapovou stránku věci.

K dispozici jsou především Google Maps a Mapy.cz od české společnosti Seznam.cz. Obě služby nabízí velké množství funkcí a jsou kvalitně podporovány.

Ve druhé části, která je zaměřena na informační(databázovou) část aplikace, bude čtenář seznámen s webovými stránkami OrqStructure a IS/STAG.

2.4.1 Část týkající se mapy

Google Maps

Velice známá, nejen mobilní, aplikace od společnosti Google je masivně rozšířena. A to především díky tomu, že je tato aplikace (a mnoho dalších) předinstalovaná výrobci mobilních telefonů. Původně to bylo pravidlo společnosti Google pro výrobce, kteří chtěli využít operační systém Android spolu s obchodem Google Play. Po obdržené pokutě od Evropské komise [2], se situace změnila. Výrobci telefonů již nemusí aplikace instalovat automaticky. Momentálně je situace taková, že výrobci telefonů musí za využití Google aplikací platit.

Dalším důvodem velké obliby této aplikace jsou nulové pořizovací náklady. Aplikace je zdarma ke stažení v obchodě Google Play.

Posledním, ale neméně důležitým faktorem pro vysoké rozšíření je určitě samotná kvalita aplikace. Ta obsahuje kvalitní mapové podklady, možnost stáhnout si oblast pro použití v situaci, kdy je uživatel offline, navigaci, hlasové pokyny, geocoding⁷ a mnoho dalších funkcionalit.

⁷ Geocoding – výraz pro překlad názvu místa na mapě na souřadnice a opačně.

Nicméně pro potřeby cílové skupiny této práce obsahuje tato aplikace i několik nedostatků, vycházejících z faktu, že je aplikace zaměřena na celý svět a cílem této práce je se soustředit na jednu malou konkrétní oblast. Mezi tyto nedostatky patří například nedostatečná explicitace řešené oblasti a objektů ležících v ní.

V případě geocodingu se při bližším pohledu ukáže, že pokrytí řešené oblasti není stoprocentní. Některé budovy lze vyhledat dle názvu, některé nikoliv. Dalším problémem je částečně omezená podpora navigace uvnitř kampusu. Tímto je myšleno trasování po chodnících a cestách uvnitř kampusu.

Mapy.cz

Jedná se o lokální, široce známou a používanou aplikaci od české společnosti Seznam.cz. Tato aplikace má podobné přednosti i nedostatky jako výše zmíněné Google Maps. Taktéž obsahuje kvalitní a aktuální mapové podklady, funkcionality jako navigaci, geocoding, stažení oblasti pro offline využití a další.

Na druhou stranu ale obsahuje i podobné nedostatky. Opět se jedná především o nepříliš vizuálně vyznačenou oblast kampusu Jihočeské univerzity a objektů patřících do této oblasti.

Co se týče navigace na úrovni kampusu, je na tom tato aplikace podobně jako Google Maps a nabízí navigaci přes chodníky pro chodce. Naopak v případě geocodingu jsou na tom Mapy.cz podstatně hůře než Google Maps. Většinu budov nelze vyhledat podle jejich názvu.

Další dostupné aplikace

Mimo zmíněné aplikace je k dispozici celá řada možností často založených na volně dostupných mapových podkladech OpenStreetMap. Většina těchto aplikací ovšem kvalitativně nestačí na výše zmíněné. A to především z pohledu samotné aplikace. Mapové podklady lze ale bez problému srovnávat i s lídry v tomto segmentu.

Mezi ty kvalitnější aplikace patří například *OsmAnd*, *MAPS.ME*, *GPS*, *Maps*, *Navigate*, *Traffic & Area Calculating*, nebo *Mapy a navigace GPS*. Častým negativem u těchto aplikací jsou přítomné reklamy. Některé případy jsou v takovém stavu, že díky reklamám nelze aplikaci skoro používat ani na velmi výkonném mobilním telefonu. Všechny tyto aplikace jsou dostupné prostřednictvím Google Play obchodu. Častým jevem je také fakt, že u dané aplikace existují dvě verze. Jedna je zdarma a obsahuje zmiňované reklamy a poté placená verze, která přidává některé funkce a zároveň odstraňuje reklamu.

Z pohledu geocodingu a vyhledávání budov dle jejich názvu je stav podobný jako v případě Google Maps a Mapy.cz.

2.4.2 Část týkající se informací

Druhá část zabývající se získáním informací o objektech kampusu Jihočeské univerzity je v mnohem horším stavu. Na úrovni mobilní aplikace neexistuje žádné řešení (V době tvorby aplikace.). Jediná možnost, jak se k informacím dostat, jsou webové stránky OrqStructure, respektive STAG.

OrqStructure

Jedná se o server obsahující organizační strukturu Jihočeské univerzity. Tento systém obsahuje informace jako názvy budov, místnosti v budově, osoby sídlící v místnostech, jména, příjmení, nebo emailové adresy zaměstnanců. Tyto informace jsou uloženy ve formě objektové databáze. Data, která se týkají výhradně budov, jsou získávána ze systému pro správu majetku AMI. Tato služba je přístupná v podobně webové stránky.

Hlavním problémem je, že tato webová stránka není uzpůsobená pro přístup přes mobilní zařízení. Naopak mezi kladné vlastnosti lze zařadit vyhledávač s poměrně širokými možnostmi úpravy a parametrizace vyhledávání.

IS/STAG

Portál IS/STAG je informační systém Jihočeské univerzity. Jednou z funkcí tohoto systému je vyhledávání informací jako jsou místnosti a učitelé. Podobně jako v případě OrqStructure není uživatelská přívětivost při použití mobilního zařízení příliš vysoká.

Výhodou pro studenty v tomto systému je způsob pojmenování budov a místností. A to především v porovnání s identifikátory v systému OrqStructure. Tyto náležitosti jsou podrobněji popsány v kapitole 2.5 *Zjištěné problémy se stavem informačních dat Jihočeské univerzity*.

2.4.3 Nedostatky dostupných aplikací

Dostupná řešení obsahují některé nedostatky, které by měla nová aplikace odstranit:

- Z pohledu map a navigace:
 - Některá řešení neposkytují (nebo poskytují pouze částečně) navigaci na úrovni kampusu Jihočeské univerzity.
 - Nedostatečná čitelnost map, respektive vyznačení jednotlivých budov.
- Z pohledu informací o budovách:
 - Relativně složitá cesta k získání informací.
 - Chybí ucelený systém pro reprezentaci těchto dat.
 - Není k dispozici mobilní aplikace.
 - Nepříliš uživatelsky přívětivé při použití mobilního zařízení.

2.4.4 Přednosti dostupných aplikací

Již existující možnosti samozřejmě obsahují i jisté přednosti, které by měla nová aplikace, pokud možno převzít:

- Z pohledu map a navigace:
 - Intuitivní ovládání.
 - Určení polohy.
 - Kvalitní mapové podklady (ve smyslu přesnosti a obsáhlosti map).
 - Jednoduchý design aplikací.
- Z pohledu informací o budovách:
 - Dostupnost informací.
 - Vyhledávání.

Nová aplikace bude plnit především následující požadavky, v rozsahu dostupných služeb:

- Spojit obě funkce popsané u existujících řešení výše.
- Zobrazovat informace o objektech v kampusu Jihočeské univerzity.
- Možnost vyhledávat tyto informace.
- Vykreslit mapu oblasti.
- Získat a vykreslit polohu uživatele (fyzického zařízení).
- Vykreslit trasu od polohy uživatele k vybranému místu na mapě.

2.5 Zjištěné problémy se stavem informačních dat Jihočeské univerzity

Při analýze a následné snaze získat data o budovách a zaměstnancích Jihočeské univerzity bylo zjištěno několik problémů se stavem těchto dat. Potřebná data lze rozdělit do dvou oblastí. Budovy a ostatní nemovitosti a informace týkající se zaměstnanců.

OrqStructure a AMI

Informace o budovách primárně vychází ze systému pro správu majetku AMI. Z tohoto systému získává data o budovách informační systém OrqStructure. Systém AMI je relativně rozsáhlý, zejména z pohledu potřeb vytvářené aplikace. V tomto systému je zahrnuto vše od budov, místností, chodeb, po sklady, komory, sklepy nebo například mezipatra a pisoáry. To je samozřejmě pro potřeby aplikace naprosto zbytečné a nevyužitelné množství dat. Uživatelé budou zajímat především učitelé nebo ostatní zaměstnanci, místnosti ve kterých sídlí a učebny. Tyto informace pochopitelně v systému AMI nenajdeme. Tato data jsou dostupná prostřednictvím OrqStructure.

Na tomto místě se ukáže první vážný problém. Data zde uložená, jsou extrémně chaotická. Zejména proto, že u spousty osob chybí určité informace. Mezi tyto informace patří emailová adresa, telefon, nebo budova i místnost, kde má tato osoba pracoviště. U některých osob jsou tato data sice k dohledání, ale naopak jsou zastaralá nebo jednoduše chybná.

Dalším častým jevem, se kterým se lze setkat je, že u jednoho zaměstnance existuje několik záznamů. Důvodem může být situace, kdy tento zaměstnanec zastává více pozic. Ovšem za momentálního stavu nelze nijak efektivně zjistit, zda tato osoba skutečně vykonává všechny tyto funkce, nebo je některý záznam zastaralý a pouze neodstraněný. Problém to představuje především v situaci, kdy uživatel hledá místnost, ve které daná osoba sídlí a nalezne několik různých verzí.

Další nedostatek lze najít v označení jednotlivých budov. Jednou je budova označena identifikátorem převzatým ze systému AMI a po druhé jakousi zkratkou pro danou budovu vycházející z jejího názvu. Zaslíbený člověk sice může pochopit o kterou budovu se tedy jedná, ale student, který je zde nový, se nemá šanci v tomto zorientovat.

IS/STAG

Třetím potencionálním zdrojem informací může být portál IS/STAG. Z tohoto systému vychází rozvrhy studentů, které obsahují názvy budov a místností, ve kterých mají vyučování. Zde jsou opět názvy řešeny jinak, a to čistě pomocí zkratk vycházejících z názvů budov. Pro účely samotného rozvrhu se rozhodně jedná o fungující způsob. Student se rozhodně lépe zorientuje dle názvu typu „BB-2“ (tedy budova B, učebna 2), nežli podle identifikačního čísla v externím systému (budova - 0103, místnost – 00006). Ovšem pro potřeby aplikace je nemožné tyto odlišné identifikátory z více systémů rozumně sloučit a využít pouze ty, které jsou potřeba.

Společným nedostatkem pro všechny popsané systémy je jednojazyčnost dostupných dat. Přesněji řečeno, některé texty se dají přepnout do anglického jazyka, většina ovšem nikoliv.

Dále jsou ve STAGU k nalezení i učitelé. I zde jsou ale informace neúplné. Chybějí zde emailové adresy, telefonní čísla, pracoviště i místnosti. Pro účel této práce je tak ve výsledku momentálně nejjednodušší způsob rozumného získání potřebných dat, si „ručně vyzobat“ z uvedených zdrojů co je potřeba a posléze upravit tato data dle potřeb aplikace.

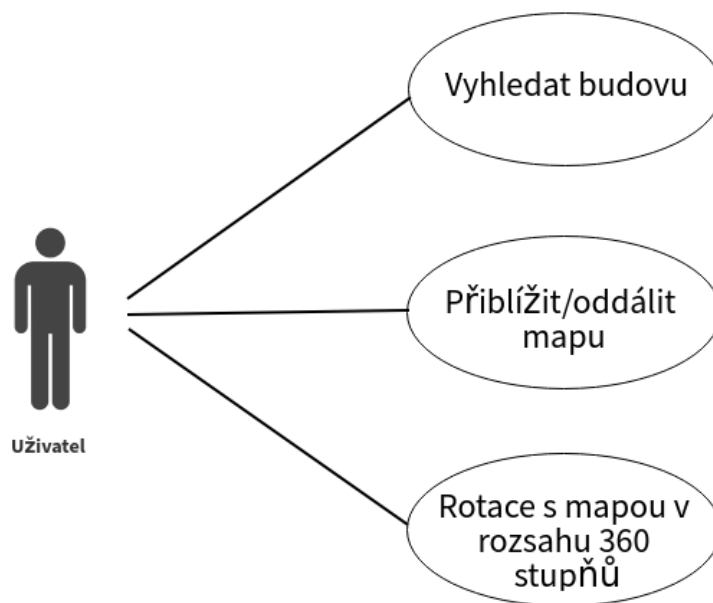
Bodové shrnutí nalezených problémů:

- Nejednotnost identifikátorů napříč systémy (OrqStructure, IS/STAG, AMI).
- Chybějící informace (OrqStructure a IS/STAG).
- Zastaralé informace (OrqStructure).
- Chybné informace (OrqStructure).
- Duplicitní informace (OrqStructure).
- Neúplná podpora jazykových mutací (OrqStructure, IS/STAG).

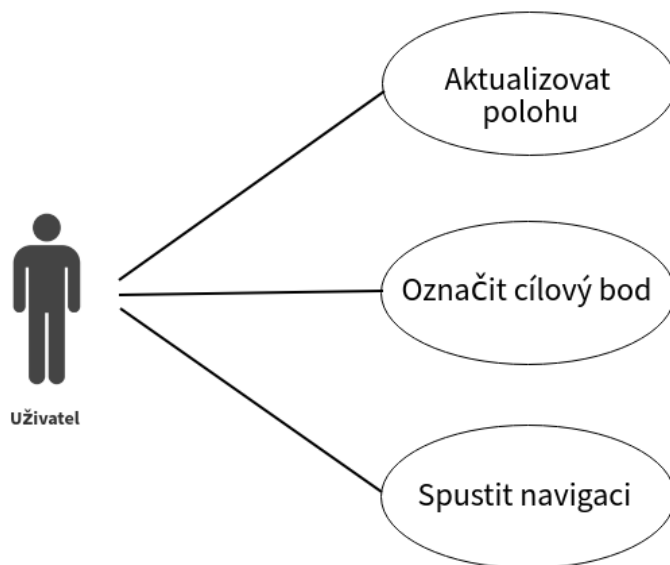
2.6 Případy užití

Na základě popsaných skutečností byly sestaveny případy užití. Pro vizualizaci a přehlednost byly související případy spojeny do několika částí.

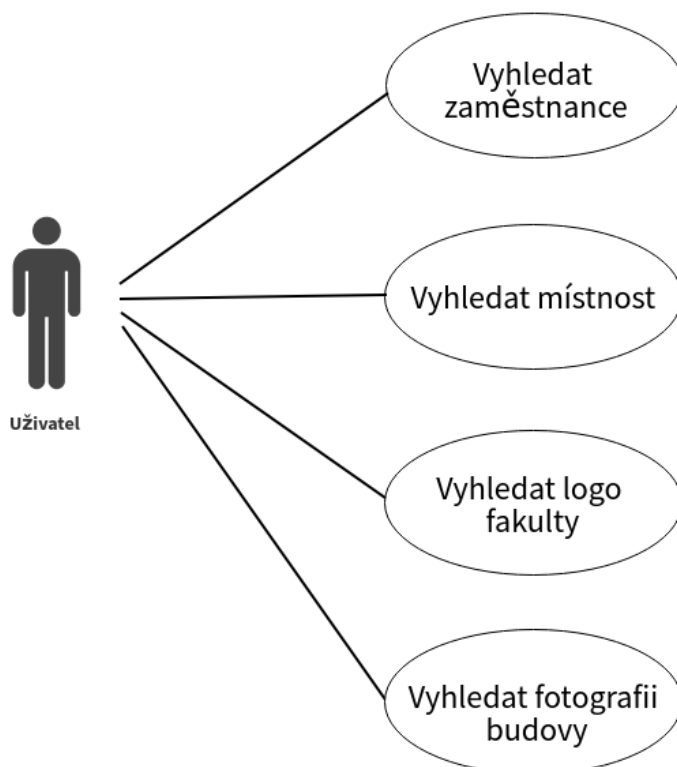
- Práce s mapou, viz Obrázek 2.1.
- Navigace, viz Obrázek 2.2.
- Vyhledávání, viz Obrázek 2.3.
- Nastavení jazyka, viz Obrázek 2.4.



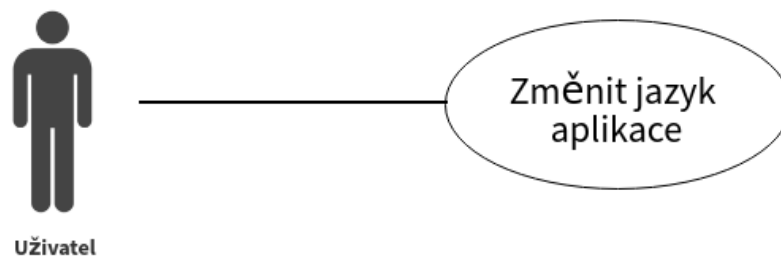
Obrázek 2. 1: Diagram případů užití – část Práce s mapou



Obrázek 2. 2: Diagram případů užití – část Navigace



Obrázek 2. 3: Diagram případů užití – část Vyhledávání



Obrázek 2. 4 : Diagram případů užití – část Nastavení jazyka

2.7 Funkční požadavky

Na základě případů užití a zadávací dokumentace vychází tyto funkční požadavky.

1. Práce s mapou:
 - a. Možnost přibližovat a oddalovat mapu.
 - b. Možnost rotace mapy v rozsahu 360 stupňů.
 - c. Možnost označit místo na mapě.
 - d. Možnost vyhledat objekt dle názvu.
2. Navigace:
 - a. Umožnit označit místo/budovu jako cílový bod.
 - b. Získání polohy zařízení.
 - c. Možnost spustit navigaci.
 - d. Nechat si vykreslit trasu.
 - e. Aktualizace polohy při pohybu zařízení.
3. Vyhledávání:
 - a. Vyhledání osob.
 - b. Vyhledání místností.
 - c. Vyhledání fotografií.
 - d. Vyhledání loga fakulty.

4. Rozhraní:
 - a. Možnost vybrat si mezi jazykem v aplikaci (CZ nebo EN).

2.8 Nefunkční požadavky

1. Aplikace bude dostupná pro operační systém Android.
2. Podpora pro verze 5.0 a vyšší.
3. Získání dat ze serveru OrqStructure a IS/STAG.
4. Data uchována na databázovém serveru.
5. Použití MVVM architektury⁸.
6. Aplikace bude obsahovat jednoduché intuitivní uživatelské rozhraní.
7. Zdrojový kód bude řádně zdokumentovaný a čitelný.

⁸ Návrhový vzor Model-View-ViewModel.

3 Teoretická východiska

Tato práce je založena na několika teoretických oblastech, které jsou popsány níže a slouží k pochopení řešené problematiky.

3.1 GPS

Global Positioning System je satelitní navigační systém. Ke své činnosti využívá více než dva tucty družic obíhající na střední oběžné dráze, které vysílají signály, díky kterým dokáže GPS přijímače určit jejich polohu, rychlost a směr pohybu.

Tento systém se používá v mnoha oblastech. Například pro tvorbu mapových podkladů, zeměměřičství nebo samozřejmě navigaci. Díky velmi přesnému určování času se GPS používá i pro vědecké zkoumání zemětřesení, nebo při synchronizaci telekomunikačních sítí. Mimo to je GPS nejspíše nejdůležitějším praktickým důkazem obecné teorie relativity od Alberta Einsteina.

GPS se dělí na tři hlavní segmenty. Kosmický segment, řídicí segment a uživatelský segment.

Kosmický segment

Kosmický segment se skládá z GPS družic. 24 družic je rovnoměrně rozloženo v šesti oběžných rovinách. Tyto roviny jsou centrické vzhledem k Zemi. Družice obíhají ve výšce zhruba 20 000 kilometrů. Oběžné dráhy jsou navrženy tak, aby bylo vždy alespoň šest družic viditelných téměř z jakéhokoli místa na planetě. Zbylé družice slouží pro upřesnění určení polohy a jako zálohy v případě výpadku ostatních družic.

Řídící segment

Řídící segment kontroluje dráhy letu družic. Hlavní řídicí stanice je pod kontrolou Leteckých Sil Spojených Států Amerických. Tato stanice pravidelně zasílá každé družici aktualizaci navigačních dat. Tyto aktualizace synchronizují družicové atomové hodiny s přesností do jedné mikrosekundy.

Uživatelský segment

Uživatelský segment tvoří GPS přijímače uživatelů. Jedná se o pasivní přijímače, což znamená, že signál pouze přijímají. Díky tomu může GPS systém obsloužit neomezený počet uživatelů [3].

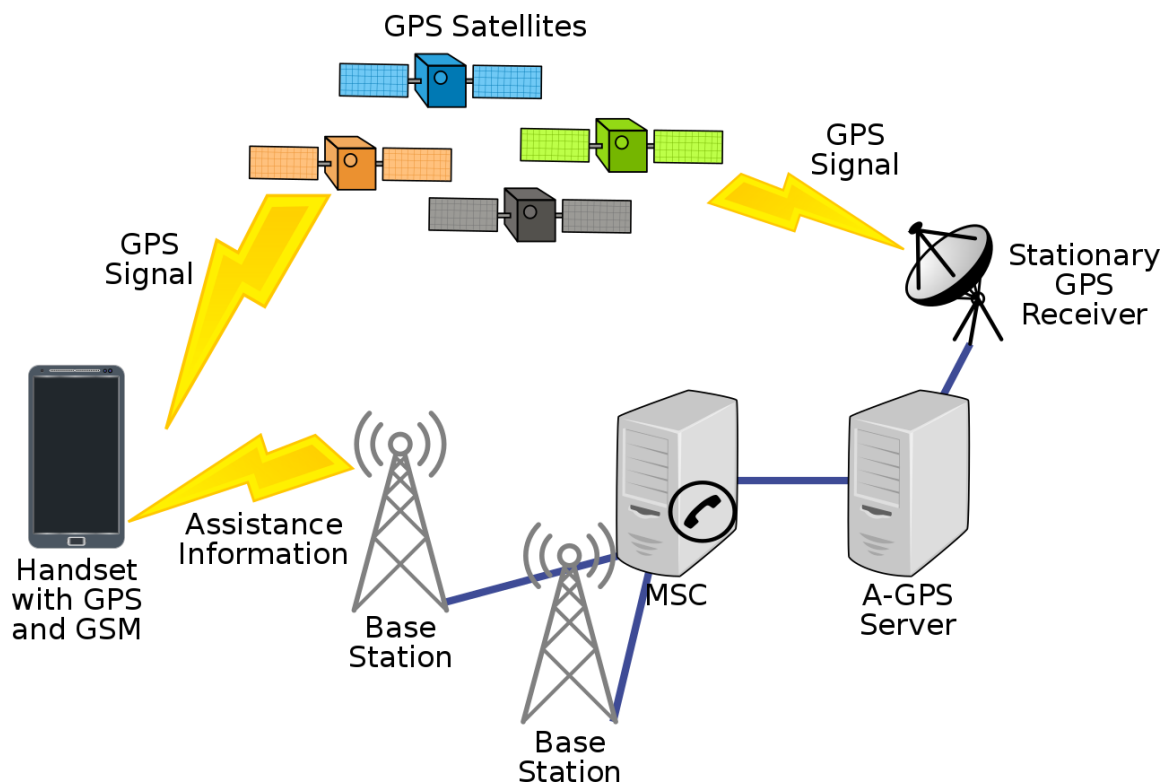
Přesnost určení polohy GPS přijímače

Pro civilní segment se pohybuje přesnost v řádech jednotkách metrů. V roce 2018 byl na trh uveden první čip, který dokáže běžně zpřesnit pozici na zhruba 30 cm. Jedná se o čip BCM47755 od společnosti Broadcom. První mobilní telefon, který tento čip nasadil byl Xiaomi Mi 8 [4].

A-GPS

Metoda pro rychlejší určení polohy GPS přijímače než pomocí samotného systému GPS. Tato metoda se používá především u mobilních telefonů. Problém při používání pouze GPS systému může být poměrně dlouhá doba než GPS přijímač, v našem případě mobilní telefon, získá informace o eferemidech⁹ GPS družic. Čekání může trvat i 10 minut. Funkce A-GPS spočívá v tom, že mobilní telefon použije datové spojení a získá informace od pozemních stanic. Tento proces výrazně ušetří čas. Pro lepší ilustraci je přiložen obrázek.

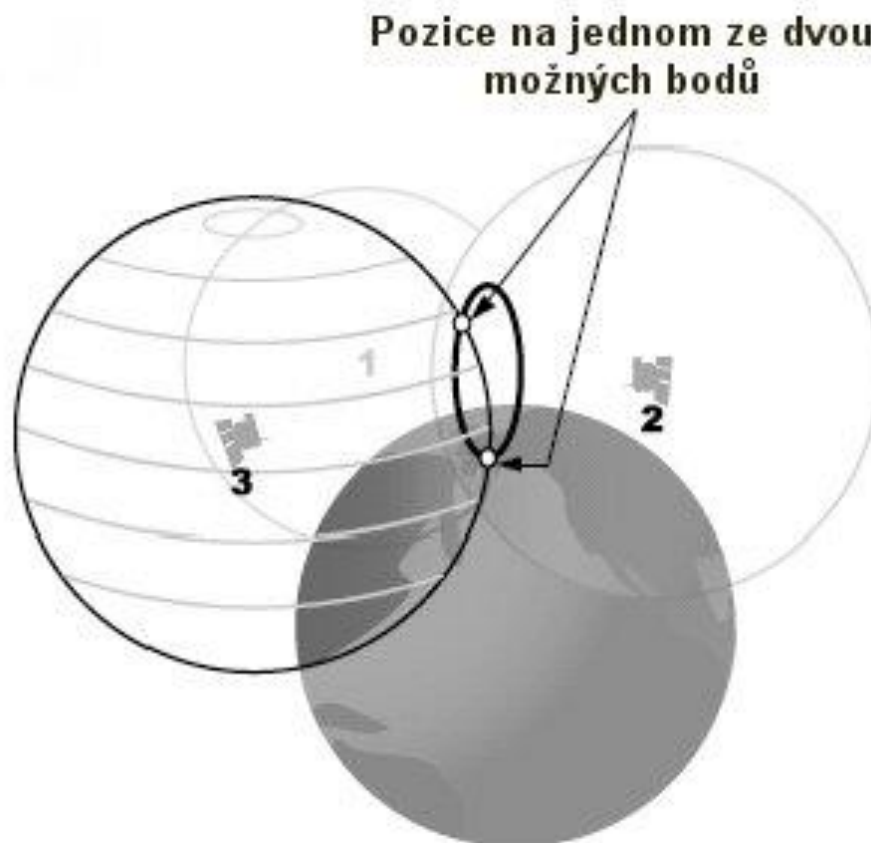
⁹ Efemeridy – Predikované polohy satelitů na oběžných drahách.



Obrázek 3. 1: A-GPS A-GPS [5]

3.2 Trilaterace

GPS přijímače mají za úkol lokalizovat čtyři nebo více satelitů, zjistit vzdálenost ke každému z nich, a díky získaným informacím spočítat svou polohu. Celý tento proces je založen na matematickém principu zvaném trilaterace. Princip je jednoduchý. V případě, že GPS přijímač zná vzdálenost od satelitu A a zároveň od satelitu B, výsledné pláště imaginárních koulí vytvoří kružnici. Je-li známá vzdálenost od třetího satelitu, výsledkem se dostane třetí kulová plocha, protínající zmíněnou kružnici ve dvou bodech. Vezme-li se Země v úvahu jako čtvrtá koule a fakt, že v jednom okamžiku může být pouze jeden bod na povrchu Země, čímž se druhý bod eliminuje, je k dispozici poloha na planetě.



Obrázek 3. 2: GPS trilaterace [6]

3.3 Dijkstrův algoritmus

Algoritmus pro nalezení nejkratší cesty v grafu. Jedná se o nejrychlejší známý algoritmus pro nalezení všech nejkratších cest ze zadaného uzlu do ostatních uzlů grafu, který nepoužívá záporné ohodnocení hran. Často používaný pro nalezení trasy v navigačních systémech.

Řekněme, že máme graf (G), množinu všech vrcholů grafu (V), specifikovaný startovní vrchol (S) a množinu všech hran grafu (E). Algoritmus funguje tak, že si pro každý vrchol v z V pamatuje délku nejkratší cesty $d[v]$.

Na začátku mají všechny vrcholy v , kromě počátečního vrcholu, který má vzdálenost nula, hodnotu $d[v]$ nekonečno, což symbolizuje, že neznáme cestu k vrcholu. Dále si algoritmus udržuje množiny P a U , kde P jsou navštívené vrcholy a U nenavštívené. Algoritmus pracuje v cyklu, dokud není množina U prázdná.

V každém cyklu se přidá jeden vrchol v_{\min} z U do P a to ten, který má nejmenší $d[v]$ ze všech v z U . Pro každý vrchol u , do kterého vede hrana $(l(v_{\min}, u))$ z v_{\min} se provede následující operace: pokud $(d[v_{\min}] + l(v_{\min}, u)) < d[u]$, pak do $d[u]$ přiřadíme hodnotu $d[v_{\min}] + l(v_{\min}, u)$, jinak nic nedělej. Po skončení je délka nejkratší cesty pro každý vrchol v z V od počátečního vrcholu uložena v $d[v]$ [7].

Pseudokód:

Inicializujte vzdálenost (d) všech vrcholů V na nekonečno

Nastav vzdálenost S na nula

Vytvoř soubor vyřešených vrcholů(P)

Vytvoř soubor nevyřešených vrcholů(U)

Přidej S do U

While(U není prázdný): Then

Vyberte jeden vrchol V z U s nejmenší vzdáleností od S

Odebrat V z U

Získejte sousední vrcholy (W) od vrcholu V

Pro každý sousední vrchol (v) z W: Udělej

Získejte váhu (w) od V do v definované podle E

If (d z v) > (d z V) + w: Then

Nastavte (d z v) s (d z V) + w

Nastavte cestu od S k v

: Konec If

Přidej v do U

Přidej V do P

: Konec For a konec While

4 Návrh

Následující část se věnuje návrhu aplikace. Čtenář se v této části seznámí s použitou architekturou aplikace, vybranými technologiemi, databází, uživatelským rozhraním a v neposlední řadě s použitými nástroji a metodikou vývoje.

4.1 Technologie

Pro vývoj mobilních aplikací existuje celá řada dostupných možností. Samozřejmě se ne všechny hodí pro každý projekt. Cílem této práce není znovu vynalézat něco, co je již dávno dostupné. S ohledem k tomu, je v práci využíváno knihoven třetích stran. Využívané knihovny, funkcionality, nebo části kódu jsou výhradně typu open source, či jsou vydávány pod licencí BSD. S ohledem na potřeby výsledné aplikace byly vybrány následující technologie.

4.1.1 Platforma

Aplikace míří na mobilní platformu. Konkrétně na zařízení fungující s operačním systémem Android. Minimální podporovanou verzí byla zvolena verze 5.0 Lollipop, neboli sdk¹⁰ verze 21.

¹⁰ SDK - software development kit.

Nativní způsob vývoje

Aplikace bude vyvíjena nativně. To přináší několik výhod oproti multiplatformnímu vývoji [8].

- Není zde žádná další vrstva nad nativním API, která přidává šanci na vytvoření více chyb, které je poté nutné řešit.
- Google ani Apple jakožto výrobci dvou hlavních zástupců OS neposkytují žádnou oficiální podporu pro multiplatformní vývoj.
- Pro nativní vývoj je stále ještě k dispozici více spolehlivých frameworků, které urychlují práci.
- Rychlost a výkon.
- Přímý přístup k funkcím zařízení. Například GPS, emailový klient, nebo telefonní hovor. Všechny tyto funkce budou v aplikaci využívány.

Zda hlavní výhoda multiplatformního vývoje, tedy možnost sdílení většiny kódu pro obě platformy, převažuje ostatní nevýhody, je složité určit. Rozhodnutí ovlivňuje řada faktorů vycházejících z konkrétního projektu, preferencí vývojového týmu nebo rozpočtu.

4.1.2 Programovací jazyk

Dle výběru platformy je potřeba zvolit programovací jazyk, ve kterém bude aplikace vyvíjena. Nativní vývoj pro operační systém Android nabízí dvě možnosti. Java nebo Kotlin. Java je zaběhnutý jazyk s velkou základnou knihoven či snadného nalezení řešení klasických problémů. Kotlin je moderní jazyk určený přímo pro vývoj aplikací na OS Android. Dokonce je možné oba tyto jazyky kombinovat v jednom projektu. Při volbě tedy záleží především na preferencích vývojáře. V tomto případě byl zvolen jazyk Java.

4.1.3 Databáze

Pro uložení dat byl zvolen momentálně populární typ databázi – NoSQL. Konkrétně Cloud Firestore. Jedná se o cloudovou databázi. Toto moderní řešení má několik podstatných výhod. Mezi ně patří zaručený a teoreticky neohrožený přístup k datům, nebo offline podpora. Další výhodou je samotné získávání dat. Není potřeba vytvářet vlastní REST API, které se poté musí provozovat na nějakém serveru [9]. Aplikace přistupuje k datům pomocí nativních SDK sad. V podstatě tím necháváme veškerý backend na starost cloudu.

Další nemalou výhodou tohoto řešení je celá řada dalších funkcí, které Google Cloud platforma nabízí. Díky tomu je zde velký potenciál pro vylepšování aplikace do budoucna. Mezi tyto užitečné funkce patří například distribuce aplikace, analytické centrum, Test Lab pro testování aplikace, Functions a mnoho dalšího.

4.1.4 Mapa

Pro zobrazení mapy existují různá řešení. Tato řešení lze rozdělit do dvou skupin. Čistě vlastní zpracování mapy, či využití dostupných nástrojů, obsahujících potřebné funkcionality a následné úpravy pro potřeby dané aplikace.

Vlastní zpracování

Prvním řešením může být vytvoření vlastní mapy pomocí obrázku, na kterém se určí vybrané souřadnicové body. Typicky rohy obrázku. Toto řešení se již v zárodku ukázalo jako nepříliš reálné. Hlavním důvodem nepoužití tohoto způsobu je požadavek na to, aby mapa obsahovala i školní budovy, které se nachází mimo hlavní kampus.

V rámci této práce je vytvoření takto rozsáhlé oblasti, v podstatě celého města České Budějovice, v rozumné kvalitě prakticky nemožné. Tento způsob by se dal označit za vhodný pouze pro opravdu malé oblasti, které jsou navíc velmi špatně zmapované, či nejsou zmapované vůbec.

Google Maps SDK

Mezi další možnosti lze samozřejmě zařadit nástroje Google Maps API. Toto řešení obsahuje veškeré potřebné funkcionality a logicky se nabízí jako první možnost. Nicméně v tomto případě existují jistá právní omezení při využití jejich API v situaci, kdy je zařízení offline a také změna politiky pro volné využití nástrojů zahrnuté do jejich API. Google sice stále nabízí možnost volného využití, tím že na začátku vývojář dostane rozpočet dvě stě dolarů, ale stále je potřeba si zaregistrovat platební metodu a při překročení uvedené částky, si začne strhávat poplatky [10].

MapBox Maps SDK for Android

MapBox Maps SDK for Android je nástrojový set pro zobrazování map v aplikacích pro operační systém Android. Jedná se o open source projekt, poskytující mapové služby v poměrně velkém rozsahu. Konkrétně mapové podklady, které Mapbox využívá, pocházejí z dalšího open source projektu – OpenStreetMap. Mapy jsou k dispozici ve vektorové formě, což umožňuje jejich neomezené a plynulé zvětšování a zmenšování.

Mimo to dává MapBox k dispozici velmi užitečný webový nástroj MapBox Studio pro vlastní úpravu stylu použité mapy. Zde je například možnost si vytvořit vlastní dataset ve formě GeoJSON, který lze poté přidat do vlastního stylu.

I zde lze samozřejmě najít položku o placení, ovšem v poněkud jiné míře. A to především pro větší společnosti a projekty s velkým komerčním potenciálem. Pro účely této aplikace zde není nutnost registrace platební metody, a MapBox v základu nabízí dostatečné měsíční množství požadavků na volání jejich API.

Shrnutí

Na základě funkčních požadavků na výslednou aplikaci bylo vybráno řešení od MapBox, které poskytuje veškeré potřebné funkcionality ve velmi kvalitním podání.

4.1.5 Navigace

Tato kapitola úzce souvisí s předchozí kapitolou 5.1.4 *Mapa*. Z volby řešení pro samotnou mapu již vychází volba také pro funkce navigace. Proto zde nebudou popsány další dvě možnosti jako v předchozím případě, a to především z důvodu opakování téměř shodných vlastností daných řešení i zde.

MapBox Navigation SDK for Android

MapBox Navigation SDK for Android je set nástrojů se zaměřením na poskytnutí funkcí důležitých pro navigaci v aplikacích orientovaných na operační systém Android. SDK obsahuje velké množství funkcionalit včetně:

- určení pozice
- vykreslení trasy
- zjišťování směru
- poskytování hlasových pokynů
- zobrazení pokroku uživatele v cestě k cíli v reálném čase
- určení, která strana cesty vede do cíle
- off-route detekce

Velice důležitá funkce pro tuto práci je možnost výběru profilu pro vypočítávání a následné vykreslení trasy k cílovému bodu. Zde bude využit profil *walking*, který pro výpočet trasy zahrnuje chodníky pro chodce. Tato možnost výrazně zlepšuje použitelnost aplikace uvnitř kampusu Jihočeské univerzity.

4.2 Architektura systému

Na následujícím obrázku můžete vidět architekturu systému. Díky použití Firestore odpadá mezistupeň v podobě REST API, místo kterého se využívají nativní SDK nástroje pro získání dat z databáze.

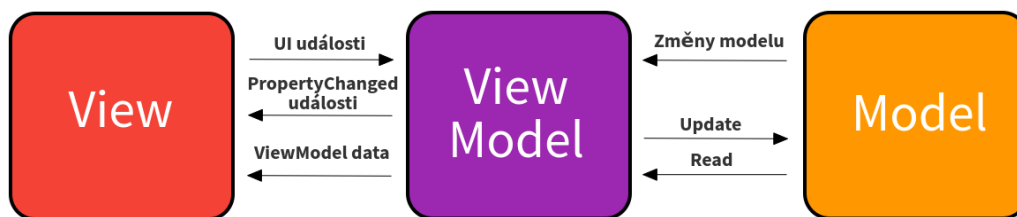


Obrázek 4. 1: Architektura systému

4.3 Architektura aplikace

Návrhový vzor MVVM

Architektura Model-View-ViewModel. Někdy také označováno jako Presentation model. Jedná se o návrhový vzor pro vývoj softwaru. Nabízí řešení, jak oddělit logiku aplikace od uživatelského rozhraní. Výsledkem je méně kódu, přehlednost a ulehčení provádění případných změn v implementaci v budoucnu. MVVM odděluje data, stav aplikace a grafické rozhraní. Skládá se ze tří následujících částí.



Obrázek 4. 2: MVVM

View

Podobně jako ve vzorech model-view-controler (MVC) a model-view-presenter (MVP) je View struktura uspořádání a vzhled toho, co vidí uživatel na obrazovce. Zobrazuje reprezentaci modelu a přijímá interakci uživatele (kliknutí, gesta, klávesnice atd.). Následně předá tuto interakci ViewModelu. View reprezentuje uživatelské rozhraní v jazyce XML.

ViewModel

Třída ViewModel spojuje Model a View. ViewModel si drží stav aplikace. Té se dotazuje uživatelské rozhraní, které podle ní vykresluje ovládací prvky. A naopak zadá-li uživatel něco do rozhraní (View), zpropaguje se tento stav do ViewModelu. Alfou a omegou této třídy jsou dvě rozhraní. Prvním je *INotifyCollectionChanged*, které uvádí, pokud je nějaký prvek dané kolekce odebrán či přidán. Jinak řečeno toto rozhraní aktualizuje View. Druhé rozhraní je *INotifyPropertyChanged*. Toto rozhraní popisuje událost, která nastane, pokud dojde ke změně vlastností z ViewModelu.

Model

Třída Model popisuje data, se kterými aplikace pracuje. Dále popisuje business logiku aplikace. Tato vrstva je naprosto oddělena od stavu ovládacích prvků. V podstatě lze modelem označit vše, co je pod úrovní grafického rozhraní [11].

Single Activity Multiple Fragments

Princip tohoto návrhu spočívá v použití jedné hlavní aktivity, která obsahuje navigační prvky aplikace a kontejner pro zobrazení jednotlivých fragmentů. Fragmenty poté představují jednotlivé obrazovky. Výhodou je velmi rychlá navigace v aplikaci, nebo naprostá kontrola nad stavem aplikace.

4.4 Metodika vývoje a použité nástroje

Metodika

Vývoj téměř jakéhokoliv softwaru je dynamický proces. Během vývoje dochází často k požadavkům na změny funkcionalit. Z tohoto důvodu je potřeba zvolit některou z agilních metodik vývoje. Jednou z nich je metodika zvaná Scrum.

Scrum pomáhá k efektivní týmové práci na komplexních produktech. Tato metodika obsahuje několik rolí. Product owner, Scrum Master a tým pro vývoj. Tato metodika v základu rozděluje práci na product backlog a sprint backlog.

Product backlog je uspořádaný seznam úkolů, které musí být splněny. Lze říci, že není nikdy kompletní, a vyvíjí se spolu s vývojem samotného produktu. Za tento backlog zodpovídá product owner.

Sprint backlog je seznam rozpracovaných úkolů v daném sprintu. Sprint je časový horizont, za který se má dokončit daný sprint backlog. Časový horizont jednoho sprintu je maximálně jeden měsíc. Určení úkolů, které mají být provedeny během dalšího sprintu se nazývá sprint planning.

Další důležitou částí je daily scrum. Jedná se o patnácti minutovou událost pro vývojový tým, kde si členové sdělí, co vypracovali od poslední schůze a zároveň se naplánuje dalších 24 hodin. Sprint retrospective je příležitost pro tým ke zhodnocení předchozího sprintu, či navrnutí zlepšení pro budoucí sprint [12].

Vzhledem k tomu, že na tomto projektu pracovala jedna osoba, která zastupovala jak vývojový tým, tak scrum mastera, musel být Scrum lehce zjednodušen.

Organizace vypadala následovně. Vývoj byl rozdělen do jednotlivých částí. Složitější a časově náročnější úkoly byly rozděleny na menší části. Těmto částem se přiřadila priorita a datum ukončení.

U částí, u kterých jejich název plně nevystihl jejich účel, byl přidán popis. Pro vizualizaci všech částí scrumu byl použit nástroj Trello¹¹. Zde byly vytvořeny tyto sloupce:

- Product backlog – veškeré nevyřešené úkoly.
- Sprint backlog – úkoly určené k dokončení v aktuálním sprintu.
- Aktuálně vyvíjeno – aktuálně řešené úkoly.
- Testováno – dokončené úkoly určené k otestování.
- Dokončeno – veškeré dokončené úkoly.

Sloupce *Product backlog* a *Sprint backlog* byly seřazeny dle priority jednotlivých úkolů. Úkol s nejvyšší prioritou byl vybrán do sloupce *Aktuálně vyvíjeno*. Po stádiu vývoje byl úkol přesunut do sloupce *Testováno*. Po úspěšném otestování byl úkol dokončen. Po celém tomto procesu byl úkol uložen verzovacím nástrojem Git¹² do repozitáře.

4.5 Wireframes

Drátěné modely slouží jako náhled nějakého řešení. Nejedná se o grafický návrh, nejsou zde obrázky, barevná úprava atd. Pomocí drátěných modelů se definují funkce a obsah jednotlivých obrazovek aplikace. Aplikace bude obsahovat 6 hlavních obrazovek, pět vyskakovacích oken závislých i nezávislých na interakci uživatele a hlavní menu.

¹¹ Trello dostupné na <https://trello.com>

¹² GIT dostupný na <https://git-scm.com>

V této kapitole budou popsány jednotlivé hlavní obrazovky. Některé wireframy si můžete prohlédnout na obrázku 4.3: Wireframe – Home, DatabasePersons.

Home

Tato obrazovka je nastavena jako defaultní. Při každém spuštění aplikace je právě tato obrazovka otevřena. Celou plochu obrazovky zabírá mapa. Primární bod mapy, na kterou se zaměří kamera po spuštění, je nastaven na aktuální GPS pozici zařízení. V případě, že má telefon vypnutou funkci pro určení polohy, je kamera nastavena na střed areálu Jihočeské univerzity. Dále se na této obrazovce nachází tři ovládací prvky. Dvě tlačítka a jeden přepínač. Funkce všech tří prvků je pevně spjata se samotnou mapou.

Prvním je tlačítko pro vyhledávání. Po jeho stisku se zobrazí vyskakovací okno s nabídkou všech školních budov v areálu školy, respektive ve městě České Budějovice. V tomto seznamu lze buďto listovat, nebo využít fulltext vyhledávání. Po vyhledání budovy v seznamu lze stiskem na položku najít budovu na mapě.

Druhé tlačítko slouží k určení aktuální polohy uživatele. Toto tlačítko se využívá především v situaci, kdy uživatel zapne funkci pro určení polohy až poté, co spustí aplikaci a aplikace ho upozorní, že není tato funkce povolena.

DatabasePersons

Obrazovka, na které se pracuje s databází. Konkrétně s tou částí, která se týká zaměstnanců Jihočeské univerzity. Při spuštění této obrazovky jsou k dispozici dva ovládací prvky. Spinner a searchView. Pomocí spinneru si uživatel zvolí, podle jakého parametru chce vyhledávat. Na výběr bude příjmení, jméno, telefon, email a označení budovy, ve které sídlí.

SearchView je klasické textové pole, do kterého uživatel píše co/koho chce hledat. V případě, že hledaná data existují, zobrazí se každý výsledek ve formě vlastní cardView. CardView obsahuje osm řádků s různými informacemi. Tři z nich jsou zároveň odkazy.

Prvním je email, po jehož stisku se otevře mailový klient a do položky adresát se vloží adresa dané osoby. Druhým je telefonní číslo, po jehož stisku se otevře aplikace pro hovory. Číslo osoby se přeneso do této aplikace a uživatel může provést hovor.

Posledním prvkem je budova, ve které osoba sídlí. Po stisku se otevře *Home* obrazovka se zaměřenou kamerou na budovu odpovídající hledané osobě. Na následujícím obrázku jsou vidět obrazovky *Home* a *DatabasePersons*. Zbylé obrazovky jsou v příloze na konci práce.



Obrázek 4. 3: Wireframe – Home, DatabasePersons

DatabaseRooms

Principiálně totožná obrazovka jako výše popsaná *DatabasePersons*. Rozdílné jsou pouze informace, které obsahuje.

Spinner nabízí hledání dle označení budovy, názvu budovy, označení místnosti a typu místnosti. Výsledkem je opět *cardView*, které obsahuje informace o dané místnosti. Ovládacím prvkem je budova, ve které se místnost nachází. Po stisku se otevře obrazovka *Home* a kamera najede na určenou budovu.

FacultyImages

Tato obrazovka je v podstatě barevná legenda k mapě kampusu. Obsahuje obrázky s logy jednotlivých fakult. Tato loga barevně korespondují s barevným vyznačením budov na mapě. Například logo přírodovědecké fakulty má tmavě zelenou barvu. Veškeré budovy na mapě, které patří k přírodovědecké fakultě, jsou tedy označeny tmavě zeleně.

Jediným ovládacím prvkem na této obrazovce je searchView. Uživatelé díky tomu mohou vyhledávat loga dle názvu fakulty.

Photogallery

Obrazovka obsahující fotografie jednotlivých budov, které slouží k lepší orientaci studentů. V těchto fotografiích je opět možnost buďto listovat, nebo použít vyhledávání.

AboutApp

Čistě informativní obrazovka. Uživatel zde najde základní informace o aplikaci. Konkrétně název, účel aplikace, nebo jméno autora.

ImportantLinks

Doplňková obrazovka obsahující jednoduché, ale mnohdy velice nápomocné informace. Nacházejí se zde například odkazy na různé školní weby nebo odkaz na stránky pro hledání spojů městské hromadné dopravy. Pro zahraniční studenty mohou být nápomocná i telefonní čísla integrovaného záchranného systému.

MainMenu

Hlavní menu vysouvající se pohybem od levé strany obrazovky. Do tohoto menu se lze dostat z kterékoliv obrazovky, a naopak z menu se lze dostat na jakoukoliv obrazovku.

Vyskakovací obrazovky

Aplikace obsahuje několik vyskakovacích oken. Od nejjednodušších a čistě informativních, až po okna vyžadující interakci uživatele, která má poté nějaký výsledek. První případ slouží pouze k předání nějaké informace. Typicky se jedná o informování uživatele o tom, že není k dispozici připojení k internetu, přístup k poloze zařízení, nebo například že hledaná data v databázi neexistují.

Druhým případem je například vyskakovací okno určené k nastavení jazyka aplikace – *Language*.

Language

Vyskakovací okno, ve kterém uživatel nastavuje jazyk aplikace. Ve výchozím stavu po instalaci aplikace je nastaven anglický jazyk. Uživatel má momentálně na výběr dvě možnosti. Zvolená možnost se uloží a při každém dalším spuštění aplikace bude použit ten jazyk, jaký si uživatel zvolil.

4.6 Uživatelské rozhraní

Uživatelské rozhraní je extrémně důležitá část každé aplikace pro její finální úspěch. Koncového uživatele typicky nezajímá nic o tom, jak aplikace funguje na pozadí, co ke své činnosti používá a podobně. Pokud není prostředí uživatelsky přívětivé, naprosto se tím zkazí celkový dojem z aplikace. V případě softwaru tohoto typu tak rozhodně neplatí známé přísloví „nesuď knihu podle obalu“. Proto je nutné navrhnout vhodné uživatelské rozhraní.

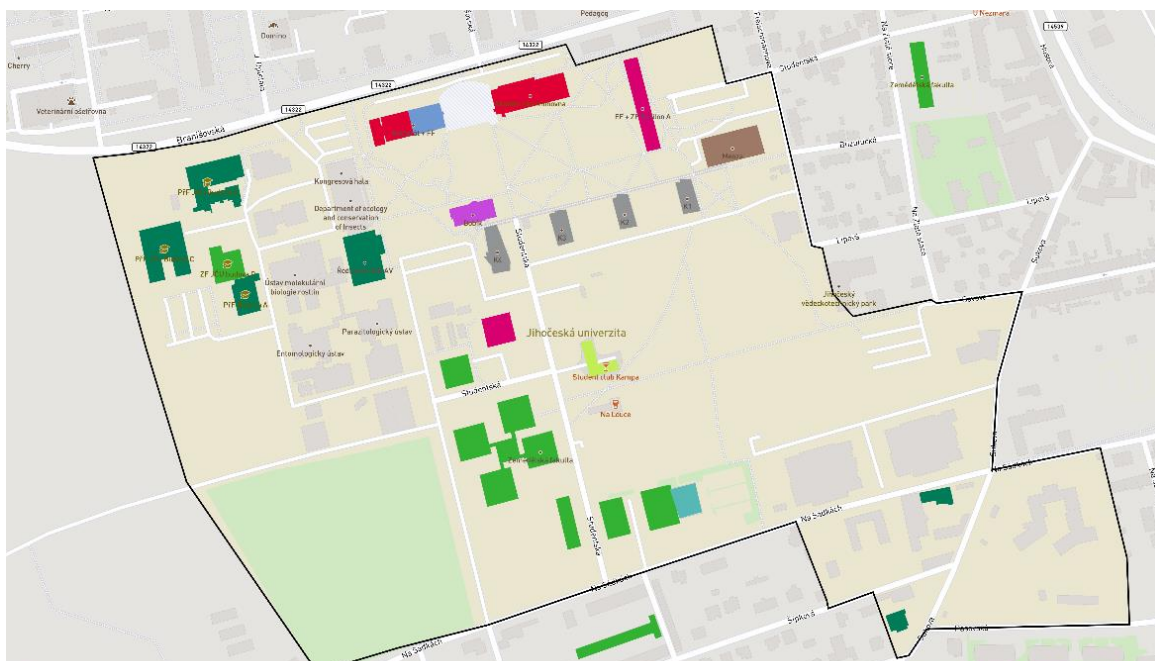
Při návrhu grafického rozhraní v této aplikaci byl kladen důraz na minimalismus, jednoduchost a přehlednost. Celé grafické prostředí bylo barevně směřováno do červené barvy s hex kódem #E00034, která je typická pro Jihočeskou univerzitu. Použity byly standardní a tím pádem ověřené ovládací prvky pro operační systém Android.

4.7 Návrh vzhledu mapy

Úprava základních dostupných mapových podkladů byla provedena pomocí Mapbox studia [13]. Mapbox studio obsahuje širokou škálu možností pro modifikaci mapy. Současně existuje mnoho volně dostupných stylů od komunity, které lze využít, nebo si lze vybrat základní mapu a tu si kompletně upravit podle vlastních potřeb.

V této práci byla zvolena druhá možnost. Tedy vlastní vytvoření stylu mapy. Mezi tyto možnosti úprav patří například přidání vlastních popisků, pomocí vlastních data-setů lze vytvářet nové vrstvy nad původní mapou, nebo měnit barvy či přiblížení v již originálních použitých vrstvách. Díky těmto nástrojům lze velmi efektivně docílit požadovaného výsledku. V tomto případě se jedná o explicitaci řešené oblasti a budov ležících uvnitř kampusu.

Jednou z klíčových úprav je vyznačení budov. Budovy byly rozděleny dle jednotlivých fakult a následně označeny barvou odpovídající dané fakultě. Tyto barvy opět vychází z oficiálního manuálu pro vizuální styl univerzity [14].



Obrázek 4. 4: Upravená mapa kampusu

5 Implementace

V této kapitole autor popisuje implementaci konkrétní aplikace s využitím výše popsaných technologií, knihoven a nástrojů. Obsahem této části není kompletní zdrojový kód aplikace. Budou vybrány a následně popsány pouze nejpodstatnější části zdrojového kódu. Toto rozhodnutí je učiněno zejména kvůli čitelnosti a přehlednosti. Dalším důvodem je zbytečnost popisovat některé triviální části zdrojového kódu.

5.1 Zásady vývoje

Při samotné implementaci je nutné dodržovat základní zásady vývoje. Důvodem je samozřejmě srozumitelnost kódu i pro osoby, které se nepodíleli na implementaci. Mezi tyto zásady patří:

- Celý obsah psaný v anglickém jazyce.
- Pojmenování proměnných podle jejich významu.
- Pojmenování metod podle jejich významu.
- Pojmenování tříd podle jejich významu.
- Styl pojmenovávání obecně držet v zažitých konvencích použitého programovacího jazyka.
- Vytvářet dokumentaci metod a dalších netriviálních částí kódu.
- Dodržení formátování kódu.
- Verzování projektu.

5.2 Vývojové prostředí

Pro vývoj aplikace bylo použito Android Studio od firmy Google. Jedná se o vývojové prostředí založené na IntelliJ IDEA. Celé prostředí je zdarma dostupné pro platformy Windows, Mac OS X a Linux. Tento program obsahuje i emulátory mobilních telefonů pro testování aplikace a všechny běžné funkce známé z ostatních vývojových prostředí. Součástí je i grafický editor pro tvorbu uživatelského rozhraní aplikace.

5.3 Správa kódu

Pro ukládání a verzování projektu byl použit nástroj Git se soukromým repozitářem, zdarma hostovaným webovou službou GitHub¹³. Tento nástroj je při vývoji velice užitečný. Díky němu je možné se kdykoliv vrátit k jakékoliv předchozí uložené verzi projektu.

Zároveň se tímto řeší zálohování projektu. Při použití gitu lze také ke svému projektu přistupovat nezávisle na použitém hardwaru a můžete tak pohodlně vyvíjet z více různých míst.

5.4 Databáze

Pro uložení zdrojových dat byla zvolena cloudová databáze Firestore. Přesněji řečeno se jedná o dokumentově orientovanou NoSQL databázi. To znamená, že data se ukládají do dokumentů, které jsou organizovány do kolekcí. Každý dokument obsahuje pár *klíč – hodnota*. Dokumenty poté mohou obsahovat subkolekce a vnořené objekty, které mohou obsahovat primitivní nebo složité objekty. S několika málo rozdíly lze dokument označit jako JSON [15].

¹³ GitHub dostupné na <https://github.com>

V rámci této práce byla vytvořena databáze obsahující dvě kolekce. *Employees* a *rooms*. Jeden dokument v kolekci *employees* představuje jednu specifickou osobu. Stejně tak jeden dokument v kolekci *rooms* reprezentuje jednu místnost.

Nezanedbatelnou výhodou použití této databáze je také podpora pro ukládání výsledků vyhledávání do cache paměti a následný přístup k datům v situaci, kdy je zařízení offline.

Import dat do Firestore

Potřebná data pro databázi byla získána z informačních systémů OrqStructure a IS/STAG. Výsledná data byla vytvořena polo-automatizovaně. Veškerá použitá data jsou veřejně dostupná na výše zmíněných portálech.

Originální hrubá data o osobách byla stažena z OrqStructure pomocí skriptu *getRoomWithPeople?roomfilter=_ _ _* (Volná místa slouží k určení budovy, popřípadě místnosti, u které je žádoucí získání informací. Například **0236** označuje Akademickou knihovnu.). Tento skript vrací data v podobě .csv souboru. Následně byla všechna tato data upravena tak, aby neobsahovala duplikáty, chybné údaje či chybějící položky.

Data o místnostech byla získána z portálu IS/STAG. V sekci prohlížení lze vyhledat všechny budovy a místnosti. Následně je možnost stáhnout .csv soubor s informacemi o jednotlivých místnostech. Tato data byla opět upravena pro potřeby této práce.

Po potřebné úpravě dat byly oba .csv soubory konvertovány do JSON podoby pomocí online nástroje Convert CSV to JSON¹⁴.

Samotný import do databáze proběhl pomocí projektu [16] psaném v JavaScriptu, respektive Node.js¹⁵. Tento jednoduchý projekt byl vytvořen ve vývojovém prostředí IDE Visual Studio 2019 od společnosti Microsoft.

¹⁴ Convert CSV to JSON dostupný na <http://convertcsv.com/csv-to-json.htm>

¹⁵ NodeJS dostupný na <https://nodejs.org/en>

5.5 Mobilní aplikace

V následující kapitole je popsána struktura projektu a implementace samotné aplikace. Dále se zde nachází základní popis adresářů a popis tříd.

5.5.1 Adresářová struktura a popis tříd

Jednotlivé třídy jsou seskupené do adresářů na základě funkcionalit těchto tříd. Zbylé samostatné třídy nejsou v žádném adresáři. Projekt obsahuje tyto čtyři hlavní adresáře:

- adapter,
- model,
- ui,
- viewHolder.

Adapter

Adresář *adapter* obsahuje čtyři třídy. Každá třída slouží k vykreslování dat do příslušného `recyclerView`, k čemuž využívá určený `viewHolder`.

- `FacultyImageAdapter` – adaptér vykreslující loga fakult.
- `PhotoAdapter` – adaptér vykreslující fotografie budov.
- `PersonsDbAdapter` – adaptér vykreslující informace o zaměstnanci.
- `RoomsDbAdapter` – adaptér vykreslující informace o místnosti.

Model

Adresář *model* obsahuje pět tříd. Třídy představující datové modely.

- `ImageModel` – datový model pro logo fakulty.
- `PhotoModel` – datový model pro fotografii budovy.
- `PersonDbModel` – datový model pro osobu.

- RoomModel – datový model pro místnost.
- Places – model obsahující list všech školních budov a jejich zeměpisných souřadnic.

ViewHolder

Adresář *viewHolder* obsahuje čtyři třídy. Pomocí těchto tříd inicializujeme pro jednotlivé obrazovky prvky jako *textView*, *imageView* a podobně. Díky této konstrukci se lze vyhnout opakovanému používání potenciálně náročné metody *findViewById(int)*.

- ViewHolderImage,
- ViewHolderPhoto,
- ViewHolderPersonsDb,
- ViewHolderRoomDb.

Ui

Nejrozsáhlejší adresář projektu. Tento adresář představuje uživatelské prostředí a obsahuje osm dalších adresářů. Každý adresář patří k jedné obrazovce, které jsou detailněji popsány v kapitole 5.5 *Wireframes*.

- AboutApp – obrazovka zobrazující základní informace o aplikaci.
- DatabasePersons – obrazovka zobrazující informace o osobách.
- DatabaseRooms – obrazovka zobrazující informace o místnostech.
- DialogLanguages – obrazovka zobrazující nastavení jazyka aplikace.
- FacultyImage – obrazovka zobrazující loga fakult.
- Home – obrazovka zobrazující mapu.
- ImportantLinks – obrazovka zobrazující důležité odkazy a telefonní čísla.
- Photogallery – obrazovka zobrazující fotografie budov.

Každý adresář, s výjimkou *AboutApp*, *DialogLanguages* a *ImportantLinks*, obsahuje dvě až tři třídy. Adresáře *DatabasePersons* a *DatabaseRooms* mají následující strukturu:

- Fragment – V použité architektuře představuje View.
- ViewModel – Slouží ke komunikaci mezi fragmentem a zbytkem aplikace.
- Repository – Zde se získávají data z databáze.

Adresáře *FacultyImage*, *Photogallery* a *Home* obsahují fragment a viewModel.

AboutApp a *ImportantLinks* nevyužívají viewModel, protože se jedná pouze o obrazovky se statickým textem.

DialogLanguages obsahuje jedno statické textové pole a dvě tlačítka ve formě radioButtonů. Pro práci s jejich stavem je tak využít *SharedPreferences*, protože chceme, aby zůstal stav uložen i po vypnutí aplikace.

Zbývající třídy, které nejsou zařazeny do žádného adresáře:

AnimateCamera

Tato třída obsahuje metodu, která pracuje s pozicí kamery nad MapBox mapou.

GpsConnectionCheck

Třída kontrolující, zda je v telefonu zapnuto zjišťování polohy.

InternetConnectionCheck

Třída kontroluje, zda je k dispozici připojení k internetu. Zde se kontroluje jak wifi připojení, tak přístup přes mobilní data.

Languages

V této třídě se řeší jazykové nastavení aplikace.

MainActivity

Třída pracující s hlavním menu aplikace, respektive navigací mezi fragmenty.

SimpleAlertDialog

V této třídě se nachází metoda pro univerzální vyskakovací informační dialog.

5.5.2 Spinner dialog

Pro vyhledávání budov na obrazovce *Home* byl použit spinner dialog dostupný z repozitáře GitHubu [17]. V licenčním souboru u tohoto kódu je výslovně napsáno, že jej lze využít pro soukromé, a dokonce i pro komerční využití.

5.5.3 Mapa

Téměř veškeré náležitosti týkající se samotné mapy jsou zpracovány metodou *onMapReady(@NonNull final MapboxMap mapboxMap)* [18], která je implementována třídou *OnMapReadyCallback*. V této metodě se nastavuje styl mapy, veškeré přidané ukazatele na mapě jako je například pozice GPS přijímače, nebo přidávání vlastních vrstev. Také se zde pracuje s pozicí kamery.

Důležitým prvkem pro funkčnost aplikace je MapBox access token. Tento token lze vytvořit v účtu MapBox a pomocí něho získává aplikace přístup k mapě, navigaci a dalším funkcím.

Podobný token slouží i určení stylu mapy. Po vytvoření vlastního stylu se vytvoří tento identifikátor, který je poté nutné zanést do kódu.

Pro interakci s mapou je využita metoda *onMapClick(NonNull LatLng point)*. V této metodě je řešen cílový bod pro navigaci, který uživatel určí kliknutím na libovolné místo

na mapě. Poté se zde volá metoda *getRoute(Point origin, Point destination)*, která je popsána v kapitole 6.5.5 *Navigace*.

5.5.4 Poloha

Lokalizace je s použitím MapBox řešena pomocí metody *enableLocationComponent* [19], třídy *LocationEngineListener* a třídy *PermissionsListener*, která se stará o povolení získané od uživatele k použití GPS přijímače v telefonu. Tato metoda se poté volá v metodě popsané výše.

```
public void enableLocationComponent(@NonNull Style loadedMapStyle,
                                     @NonNull MapboxMap mapboxMap) {

    LocationComponent locationComponent =
        mapboxMap.getLocationComponent();

    locationComponent.activateLocationComponent(
        LocationComponentActivationOptions.builder(getApplicationContext(),
            loadedMapStyle).build());

    locationComponent.setLocationComponentEnabled(true);

    locationComponent.setCameraMode(CameraMode.TRACKING);

    locationComponent.setRenderMode(RenderMode.NORMAL);
}
```

Parametrem *loadedMapStyle* se určuje styl, který má být použit. Parametrem *mapBoxMap* se určuje mapa, která má být použita. Také je zde potřeba zvolit nastavení pro kameru a render mód.

5.5.5 Navigace

O funkci samotné navigace, respektive vykreslení trasy, se stará metoda *getRoute(Point origin, Point destination)* [20]. Trasa se vykreslí mezi bodem *origin*, který představuje aktuální pozici, a bodem *destination*, který si uživatel zvolí kliknutím na libovolné místo na mapě, respektive vytvořením ukazatele na mapě.

V této metodě se například volí, jaký navigační profil bude použit. Vzhledem k zamýšlenému využití aplikace byl zvolen profil WALKING, který k vykreslení trasy využívá chodníky pro chodce, což je hlavní cílová skupina této aplikace.

```
private void getRoute(Point origin, Point destination) {
    NavigationRoute.builder(getContext())
        .accessToken("Token")
        .origin(origin)
        .profile(DirectionsCriteria.PROFILE_WALKING)
        .destination(destination)
        .build()
        .getRoute(new Callback<DirectionsResponse>() {
            @Override
            public void onResponse(@NotNull Call<DirectionsResponse> call,
                                   @NotNull Response<DirectionsResponse> response) {
                currentRoute = response.body().routes().get(0);

                if (navigationMapRoute != null) {
                    navigationMapRoute.removeRoute();
                } else {
                    navigationMapRoute = new NavigationMapRoute(null, mapView,
                                                                    mapboxMap, R.style.NavigationMapRoute);
                }
                navigationMapRoute.addRoute(currentRoute);
            }
        });
}
```

Také je důležité zadat access token pro přístup k MapBox directions API, vybrat profil pro výpočet a vykreslení trasy, a nakonec samozřejmě přidat trasu na mapu.

6 Testování

Testování probíhalo již při vývoji aplikace primárně jejím autorem a následně po dokončení projektu byly do testování zapojeny osoby třetích stran. Celý proces testování probíhal především manuálně. Cílem bylo ověřit splnění všech funkčních požadavků a zjistit názory na grafické prostředí od osob, kterých se netýkal vývoj aplikace. Posledním motivem pro testování bylo zjistit, zda je aplikace stabilní a nedochází k pádům či jiným neočekávaným chybám.

K testování bylo využito jak fyzických, tak virtuálních mobilních telefonů od široké škály výrobců, tak aby se otestovala funkčnost aplikace v různých prostředích. Těmito prostředími jsou myšleny především proprietární grafické nadstavby samotných výrobců. Testované přístroje můžete vidět v následující tabulce:

Virtuální	Fyzické
Galaxy A5 2017	Xiaomi Mi 9T Pro
Asus ZenFone 4	Samsung Galaxy S9
Xiaomi MI 8	Samsung Galaxy A40
Huawei Mate 9	Samsung Galaxy A20
Pixel XL	LG G7 ThinQ
Pixel C	
Pixel 2	
Lenovo S5	
OnePlus 5T	
Galaxy S6	
Huawei Honor Play	
Sony Xperia XZ2	

Tabulka 6. 1: Testovaná zařízení

Testování virtuálních strojů Pixel bylo prováděno během vývoje přímo ve vývojovém prostředí Android Studio. Zbylá virtuální zařízení byla testována v rámci prostředí Test Lab [21], které nabízí Cloud Firestore. Zjištěné nedostatky a připomínky byly nahlášeny autorovi a následně odstraněny, případně minimalizovány. Na tomto místě je potřeba upozornit, že autor tímto rozhodně netvrdí, že lze očekávat bezchybný software¹⁶.

¹⁶ "Software is written by humans and therefore has bugs." – JOHN JOCOBS

7 Distribuce

Aplikace je dostupná prostřednictvím distribučního nástroje Firebase App Distribution [22]. Tento způsob byl zvolen ze dvou důvodů. Prvním je nulová finanční náročnost pro autora. Na rozdíl od klasického Google Play, kde musí vývojář platit za možnost distribuce své aplikace. Druhým důvodem jsou zajímavé a užitečné možnosti tohoto nástroje. Mezi tyto možnosti patří například velice propracovaný systém statistik o stahování a používání aplikace jednotlivými osobami.

Celý postup stažení a instalace je intuitivní a stačí následovat pokyny. Celý proces je poté popsán v příloze *Uživatelská dokumentace*. Na tomto místě najdete i odkaz ke stažení aplikace.

8 Diskuse

Další vývoj aplikace

Výsledná aplikace samozřejmě obsahuje prostor pro budoucí rozvoj a vylepšování. V této kapitole autor stručně shrne tyto možnosti.

První možností je doplnění fotografií veškerých školních budov. Tento bod bohužel nebyl splněn. Důvodem je pandemie Covid-19, která výrazně narušila plány na realizaci profesionálních fotografií. V době odevzdání práce tak aplikace obsahovala fotografie jedenácti budov.

Hlavní směr budoucího vývoje však spočívá v propojení s indoor navigací, případně aplikací pro IS/STAG. Tyto aplikace jsou vyvíjeny dalšími studenty.

Dále připadá v úvahu rozšíření vyhledávání v databázi o filtrování. Uživatel by mohl například omezit vyhledávání osoby pouze na určitou fakultu, nebo budovu. Toto opatření by mohlo pomoci uživatelům, kteří z nějakého důvodu neznají jméno (či další vyhledávací kritéria dané osoby) a osobu mohou například hledat pouze tak, že vědí, na jaké fakultě nebo v jaké budově tato osoba pracuje.

Další možnosti jsou založeny na rozšíření obsahu:

- Přidání dalších jazyků.
- Přidání dalších školních budov, které se nachází mimo České Budějovice.

Na všechny tyto kroky je aplikace připravena.

Porovnání zjištěných výsledků

Ke srovnání dosažených výsledků byl vybrán článek zabývající se aplikací založenou na Google Maps Api, která byla tvořena pro kampus Jadavpur University [23]. Popis jejich aplikace je velmi podobný aplikaci popsané v této práci. Hlavním cílem bylo nabídnout návštěvníkům univerzitního areálu lepší přehled o daném místě. Hlavní funkce jsou následující:

- Získání GPS polohy a vykreslení na mapu v reálném čase.
- Kvalitnější vyobrazení budov v kampusu.
- Vykreslení nejkratší trasy z bodu A do bodu B.

Autoři se zde shodují s autorem této práce v tvrzení, že aplikace Google Mapy a další dostupné aplikace dostatečně nezobrazují objekty uvnitř konkrétních kampusů či v dalších podobných areálech. Cesty, budovy nebo parkoviště pomocí této aplikace nelze jednoduše vyhledat. Proto jsou podobné aplikace, zaměřené typicky na nějaký uzavřený areál, velmi přínosné.

Jako řešení byla použita vlastní vrstva nad základní mapou poskytovanou Google Maps Api. Tudíž zde bylo použito podobné řešení, které obsahuje tato práce. Autoři však nezmiňují, zda se rozhodovali mezi více řešeními a rovnou popisují použití Google Maps Api pro zobrazení mapy a výpočet trasy. Toto řešení již není tak dostupné jako v době tvorby jejich aplikace. Může za to změna platební politiky, respektive cenová náročnost za využití Google Maps Api, a to především za vypočítávání jednotlivých tras či použití v situaci, kdy je zařízení offline.

Výhodou použití služeb MapBox v této práci tak momentálně spočívá především v nulovém finančním zatížení pro autora. Tento fakt má samozřejmě svá omezení. Pokud by bylo cílem vytvořit aplikaci globálního charakteru s potenciálním zpoplatněním, kterou by využívaly desetitisíce uživatelů denně, pak by ani řešení vycházející z MapBox nebylo zdarma.

Pokud se zaměříme na nabízené možnosti a kvalitu od obou služeb, tedy Google Maps Api a MapBox, zjistíme, že řešení MapBox oproti Googlu rozhodně v ničem nezaostává, a naopak přidává některé velice užitečné možnosti a funkce.

9 Závěr

Výsledkem této bakalářské práce je funkční mobilní aplikace pro jasně nejrozšířenější operační systém na trhu (v době tvorby práce), která výrazně ulehčuje orientaci v kampusu Jihočeské univerzity a vyhledávání informací o objektech v tomto areálu.

Součástí práce je rešerše aktuálního stavu na trhu s dostupnými mobilními aplikacemi zaměřující se na probíranou problematiku. Dále byla provedena analýza požadavků na aplikaci, podle které byly vybrány použité technologie pro její vývoj.

Aplikace obsahuje intuitivní grafické prostředí, které vychází z navržených wireframů, jednoduché barevné palety a z prostředí založeném na minimalistickém přístupu. Vývoj vycházel z požadavků zákazníka a probíhal dle určené metodiky Scrum, ve které byly jednotlivé úkoly přidány do backlogu a následně řazeny dle priority. Po dokončení byly úkoly verzovány pomocí nástroje Git.

Funkcionalitami této aplikace jsou zobrazení upravené mapy, získání GPS polohy, navigace z polohy uživatele na vybrané místo, vyhledávání budov na mapě, vyhledávání v databázi, která obsahuje informace o zaměstnancích a budovách, změna jazyka uživatelského prostředí nebo zobrazení a vyhledávání fotografií jednotlivých budov ve fotogalerii.

Aplikace je připravena na přidání či odebrání dalších budov i osob.

Celkově se jedná o část zamýšleného systému pro navigaci v kampusu Jihočeské univerzity. Tato práce vyřešila mapu s outdoor navigací a zdroj dat. Další části jsou zaměřeny na indoor navigaci v budovách. Ovšem i jako samostatná část je vytvořena aplikace naprosto funkční a z hlediska potřeb cílové skupiny plně použitelný software.

Seznam použité literatury

- [1] Mobile Operating System Market Share in Czech Republic - January 2020. Statcounter Global Stats [online]. [cit. 2020-02-16]. Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/czech-republic/2019>
- [2] Antitrust: Commission fines Google €4.34 billion for illegal practices regarding Android mobile devices to strengthen dominance of Google's search engine [online]. 18 July 2018 [cit. 2020-02-23]. Dostupné z: https://ec.europa.eu/commission/presscorner/detail/en/IP_18_4581
- [3] Americký družicový navigační systém NAVSTAR GPS. Český kosmický portál [online]. [cit. 2020-02-16]. Dostupné z: <http://www.czechspaceportal.cz/3-sekce/gnss-systemy/gnss-mimo-evropu/americky-navstar-gps>
- [4] World's first dual-frequency GNSS smartphone hits the market. European Global Navigation Satellite Systems Agency [online]. 04 June 2018 [cit. 2020-02-16]. Dostupné z: <https://www.gsa.europa.eu/newsroom/news/world-s-first-dual-frequency-gnss-smartphone-hits-market>
- [5] A-GPS. In: Wikimedia Commons [online]. 28 October 2017 [cit. 2020-02-16]. Dostupné z: <https://commons.wikimedia.org/wiki/File:A-GPS.svg>
- [6] BERGMANN. Jak funguje GPS? In: Svět hardware [online]. 21.6.2006 [cit. 2020-02-16]. Dostupné z: <https://www.svethardware.cz/jak-funguje-gps/21826/img/body-31.4746.jpg>
- [7] PINANDITO, Aryo, Agi Putra KHARISMA a Rizal Setya PERDANA. Framework Design for Map-Based Navigation in Google Android Platform [online]. [cit. 2019-02-10]. Dostupné z: https://www.researchgate.net/publication/324246479_Framework_Design_for_Map-Based_Navigation_in_Google_Android_Platform
- [8] NANOOQ IT. Mobilní vývoj — nativně nebo multiplatformně. Medium [online]. Oct 23, 2017 [cit. 2020-02-23]. Dostupné z: <https://medium.com/@nanooqit/mobiln%C3%AD-v%C3%BDvoj-nativn%C4%9B-nebo-multiplatformn%C4%9B-d113f8f3dfac>

- [9] 7 Reasons to Choose Google Cloud Firestore as Your Database Solution. BlueWhaleApps [online]. December 18, 2019 [cit. 2020-02-23]. Dostupné z: <https://bluewhaleapps.com/blog/7-reasons-to-choose-google-cloud-firestore-as-your-database-solution>
- [10] Google Maps Platform: Pricing [online]. [cit. 2020-02-16]. Dostupné z: <https://cloud.google.com/maps-platform/pricing/>
- [11] SALEH, Hazem. MVVM architecture, ViewModel and LiveData [online]. May 31, 2017 [cit. 2020-02-18]. Dostupné z: <https://proandroiddev.com/mvvm-architecture-viewmodel-and-livedata-part-1-604f50cda1>
- [12] The Scrum Guide™ [online]. 2018 [cit. 2020-02-19]. Dostupné z: <https://www.scrumguides.org/scrum-guide.html>
- [13] Mapbox Studio: Docs [online]. [cit. 2020-02-20]. Dostupné z: <https://docs.mapbox.com/studio-manual/overview/>
- [14] Jednotný vizuální styl univerzity [online]. [cit. 2020-02-20]. Dostupné z: <https://www.jcu.cz/o-univerzite/pro-zamestnance/jednotny-vizualni-styl-univerzity>
- [15] Cloud Firestore Data model. Firebase.google.com [online]. 2020-02-21 [cit. 2020-02-23]. Dostupné z: <https://firebase.google.com/docs/firestore/data-model>
- [16] PAACHU. How to upload data to Firebase Firestore Cloud Database. *Medium* [online]. Jan 23, 2019 [cit. 2020-01-05]. Dostupné z: <https://medium.com/@impaachu/how-to-upload-data-to-firebase-firestore-cloud-database-63543d7b34c5>
- [17] MD FARHAN RAJA. Fully Searchable Spinner Dialog. GitHub [online]. 2019 [cit. 2020-02-22]. Dostupné z: <https://github.com/MdFarhanRaja/SearchableSpinner>
- [18] Create a simple map view [online]. [cit. 2020-02-03]. Dostupné z: <https://docs.mapbox.com/android/maps/examples/create-a-simple-map-view/>
- [19] Show a user's location [online]. [cit. 2020-02-03]. Dostupné z: <https://docs.mapbox.com/android/maps/examples/show-a-users-location/>
- [20] Navigation Map Route [online]. [cit. 2020-02-03]. Dostupné z: <https://docs.mapbox.com/android/navigation/examples/navigation-map-route/>

- [21] Firebase Test Lab [online]. 2019 [cit. 2020-02-23]. Dostupné z:
<https://firebase.google.com/docs/test-lab>
- [22] Firebase App Distribution [online]. [cit. 2020-03-15]. Dostupné z:
<https://firebase.google.com/docs/app-distribution>
- [23] SUSOVAN, Jana a Chattopadhyay MATANGINI. An event-driven university campus navigation system on android platform [online]. 12-14 Feb. 2015, , 182-187 [cit. 2020-05-02]. DOI: 10.1109/AIMOC.2015.7083850. Dostupné z:
<https://ieeexplore.ieee.org/document/7083850>

Seznam obrázků

Obrázek 2. 1: Diagram případů užití – část Práce s mapou.....	12
Obrázek 2. 2: Diagram případů užití – část Navigace	13
Obrázek 2. 3: Diagram případů užití – část Vyhledávání.....	13
Obrázek 2. 4 : Diagram případů užití – část Nastavení jazyka.....	14
Obrázek 3. 1: A-GPS A-GPS [5]	18
Obrázek 3. 2: GPS trilaterace [6].....	19
Obrázek 4. 1: Architektura systému	27
Obrázek 4. 2: MVVM.....	28
Obrázek 4. 3: Wireframe – Home, DatabasePersons.....	32
Obrázek 4. 4: Upravená mapa kampusu	35

Seznam tabulek

Tabulka 6. 1: Testovaná zařízení	45
--	----

Seznam příloh

Přílohy, které jsou přímo součástí tohoto textu:

1. Wireframes
2. Uživatelská dokumentace
3. Administrátorská dokumentace

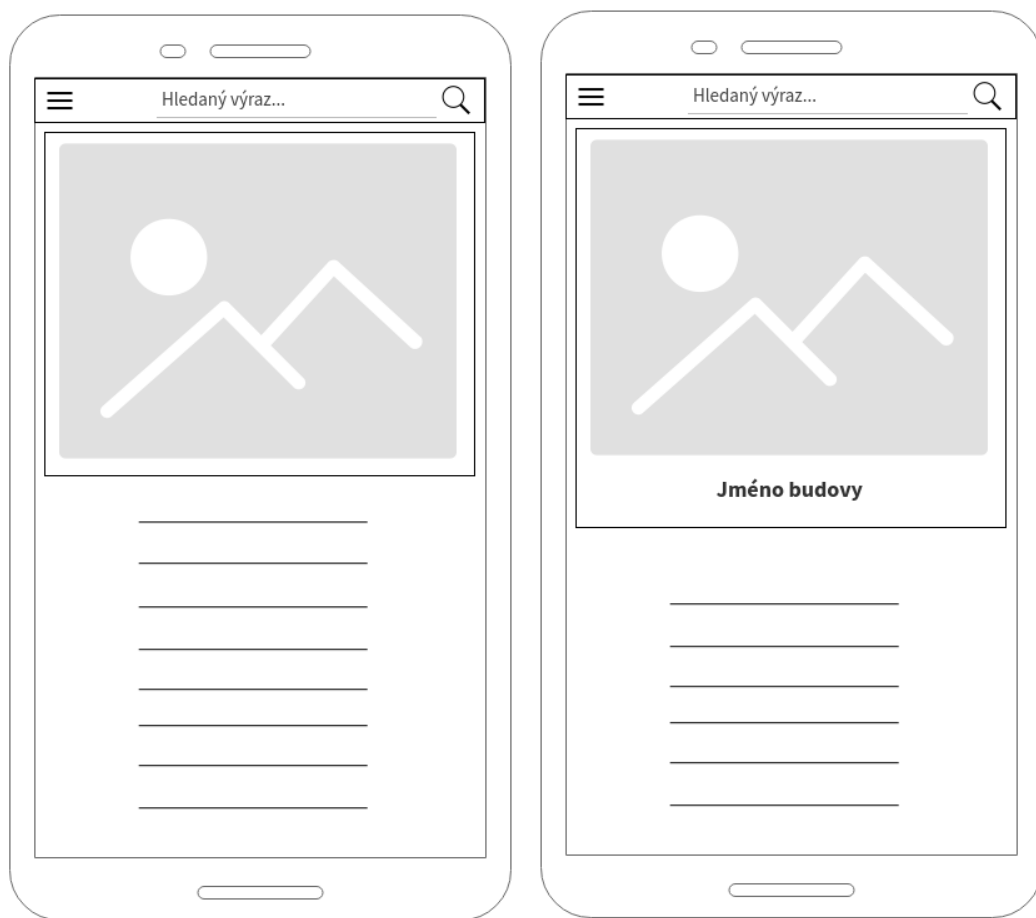
Přílohy, které jsou v přiloženém komprimovaném souboru:

1. UML diagram
2. Zdrojový kód aplikace
3. Zdrojový kód skriptu pro nahrání dat do databáze

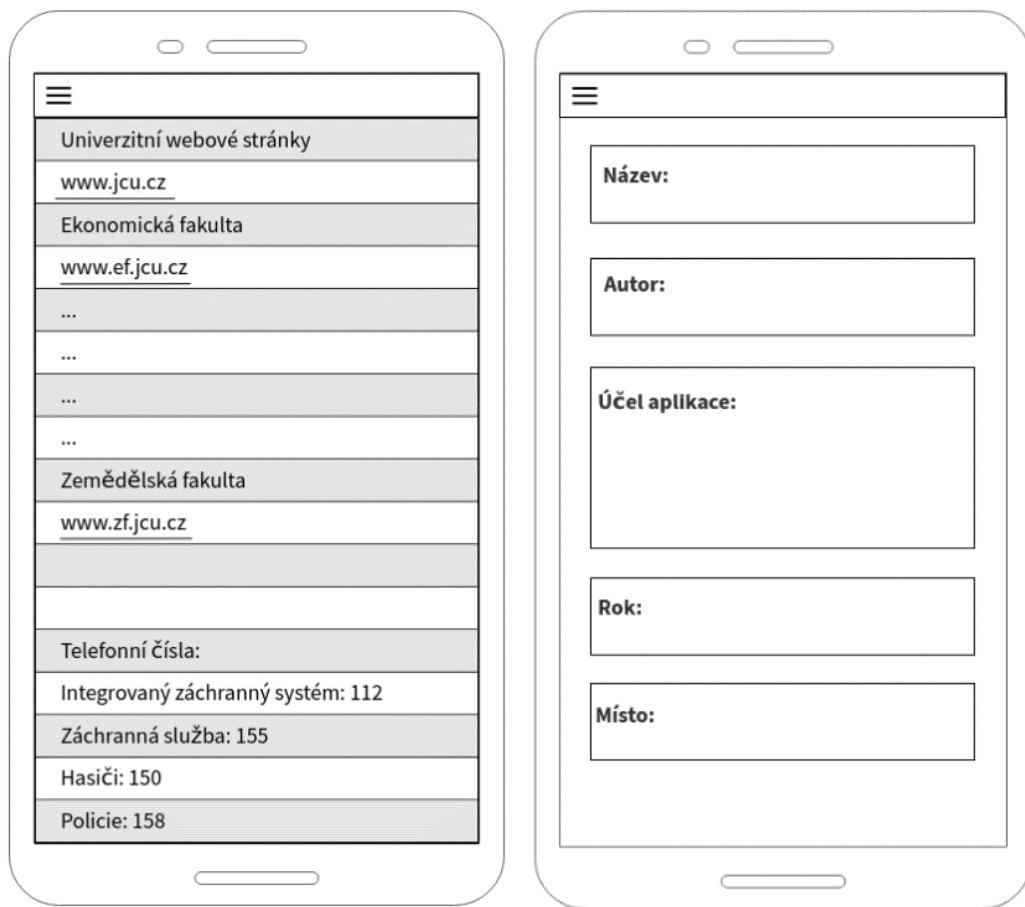
Příloha: Wireframes



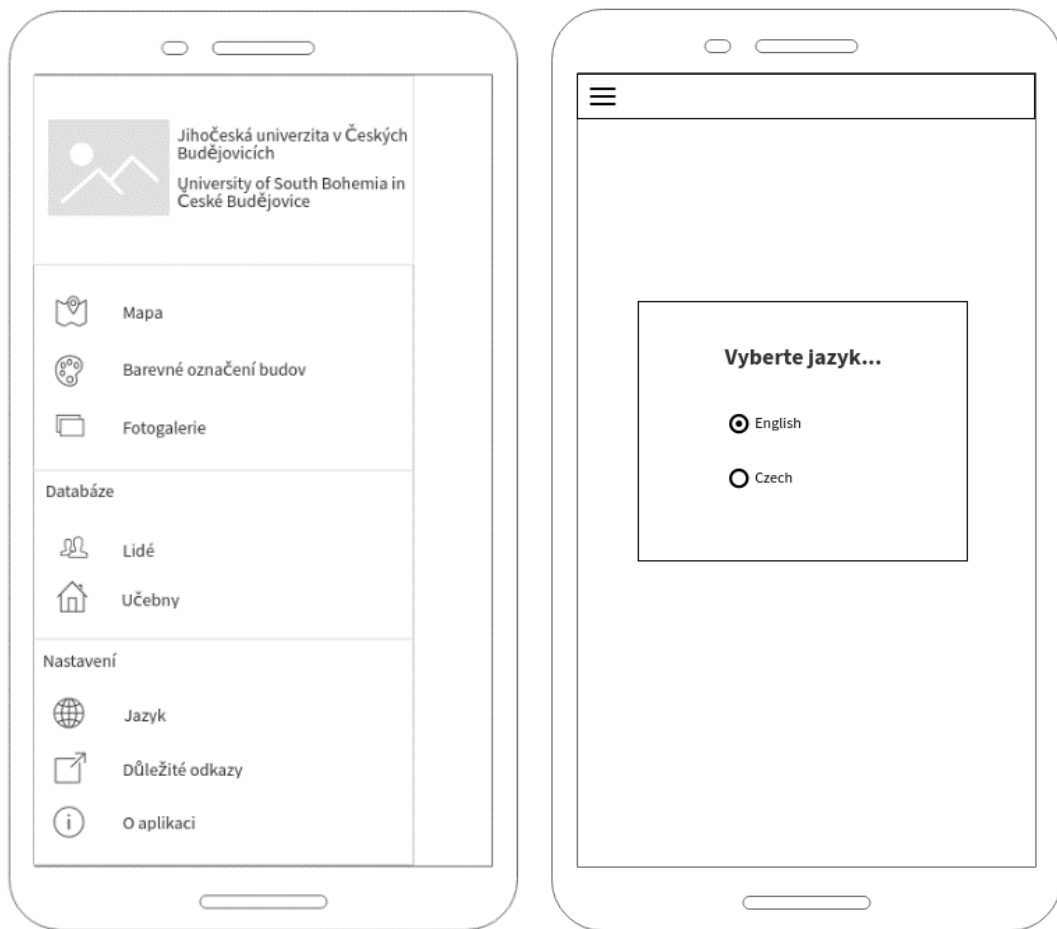
Obrázek: Wireframes - DatabaseRooms



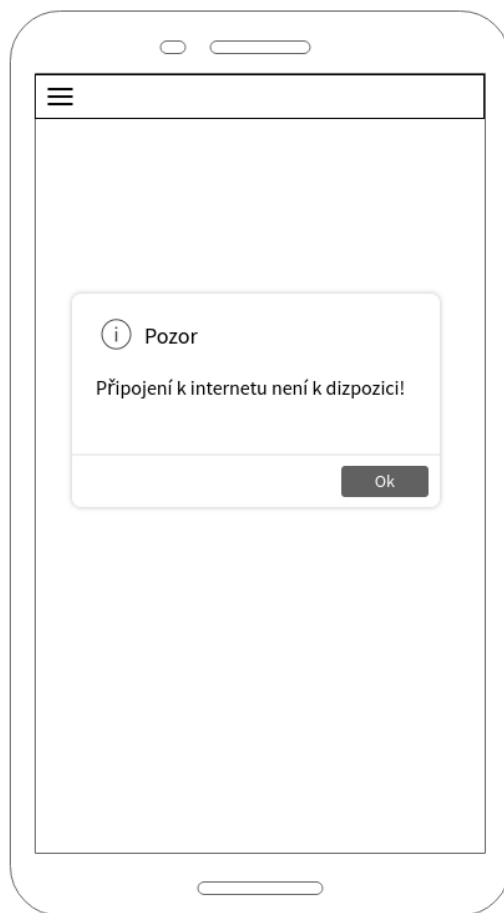
Obrázek: Wireframes - FacultyImages a Photogallery



Obrázek: Wireframes - ImportantLinks a AboutApp



Obrázek: Wireframes - MainMenu a Language



Obrázek: Wireframes - AlertDialog

Příloha: Uživatelská dokumentace

1 Podmínky pro správné fungování aplikace

1. Zařízení s operačním systémem Android 5.0 a novější.
2. Připojení k internetu.
3. Povolení instalace aplikací z jiných zdrojů, než je Google Play.

2 Stažení a instalace

V tuto chvíli je aplikace dostupná prostřednictvím distribučního systému Firestore – <https://appdistribution.firebase.dev/i/s2P5MHBa> – po otevření tohoto odkazu budete vyzváni k zadání emailové adresy.

Na zadanou adresu bude zaslána pozvánka ke stažení aplikace. Pro úspěšnou instalaci samozřejmě musíte přijatý email s pozvánkou ke stažení aplikace otevřít na mobilním zařízení. Celý postup můžete vidět na následujících obrázcích.

Upozornění: Vzhled následujících obrázků s postupem instalace se může na různých zařízeních mírně lišit.

Zde zadáte adresu, na kterou chcete dostat pozvánku ke stažení aplikace.



Map Jcu

for Android

Ready to test this app?

Enter your email address to get access

Sign up

Powered by



V dalším kroku budete požádáni o vybrání/zadání vašeho Google účtu. Důvodem je to, že celá distribuční služba Firestore je Google služba a pro její využití je potřeba Google účet. Pokud máte v telefonu přístup do Google Play, tak by to neměl být žádný problém.

 Přihlásit se k účtu Google



Vyberte účet

a pokračujte do aplikace
Firebase App Distribution

 Pavel Beran
[Redacted email address]

 Pavel Beran
[Redacted email address]

 Použít jiný účet

Budete-li pokračovat, Google bude sdílet vaše jméno, e-mailovou adresu, předvolbu jazyka a profilovou fotku s aplikací Firebase App Distribution. Před použitím aplikace Firebase App Distribution si prostudujte její [zásady ochrany soukromí](#) a [smluvní podmínky](#).

Následně stisknete tlačítko *Download*, kterým zahájíte stahování.



← Back to apps P

 **Map Jcu**
Developer contact email:
[REDACTED]

Releases

🔍 Search releases and notes

1.0 (1) ^

Latest

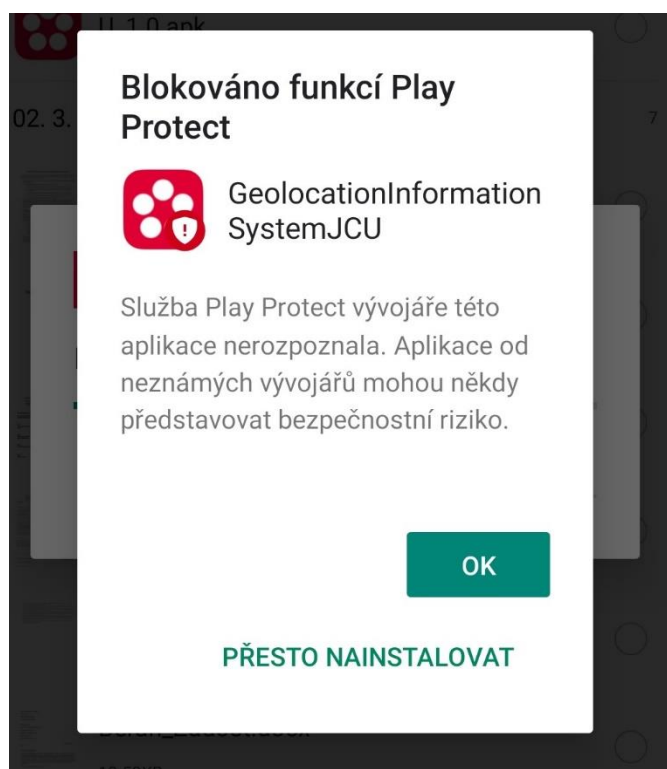
Mar 14, 2020 at 9:49 AM UTC

 Download

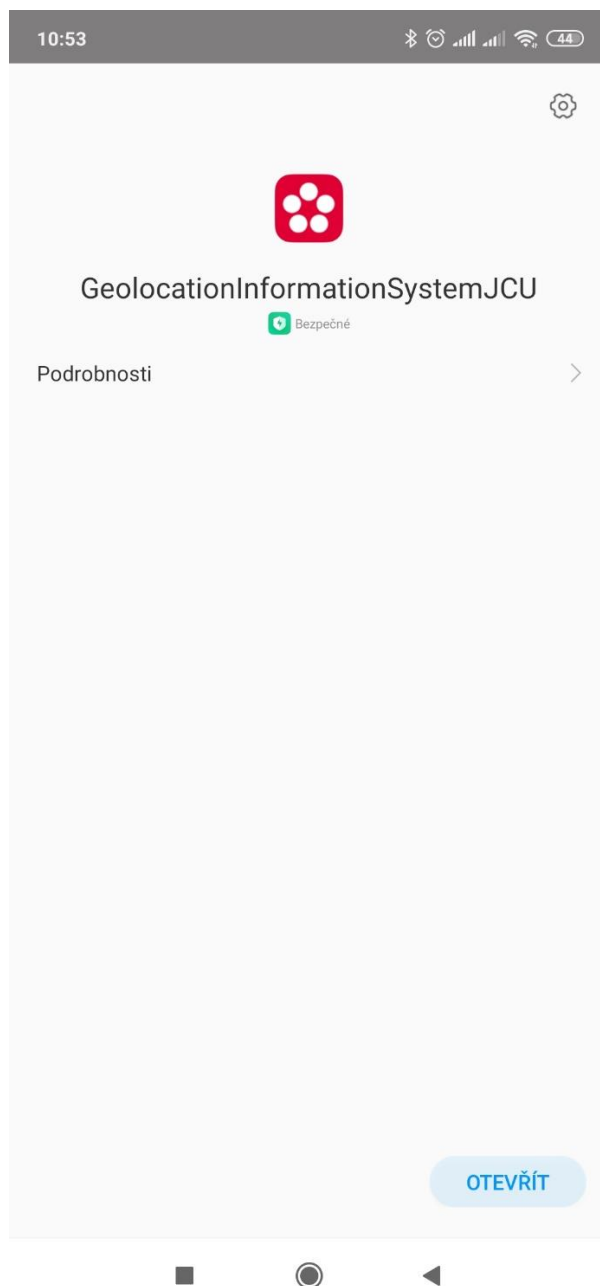
☰ Jcu Android App

Po dokončení stahování otevřete stažený soubor, čímž spustíte instalaci. V tomto bodě budete vyzváni k povolení instalování aplikací z neznámých zdrojů (Pokud jste toto již někdy v minulosti povolovali, telefon se ptát nebude.). To znamená, že budete moci nainstalovat aplikaci i mimo aplikace stažené z Google Play. Z důvodu bezpečnosti, můžete po úspěšné instalaci této aplikace, znovu zakázat instalaci z neznámých zdrojů.

Poté budete pravděpodobně upozorněni, že aplikace nepochází od registrovaného vývojáře. To znamená pouze to, že aplikace není dostupná pomocí Google Play. Klikněte na *PŘESTO NAINSTALOVAT*.

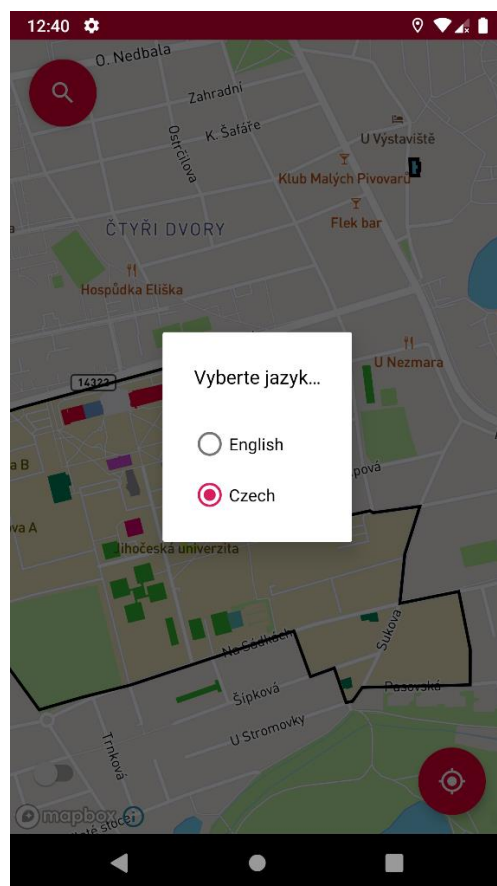
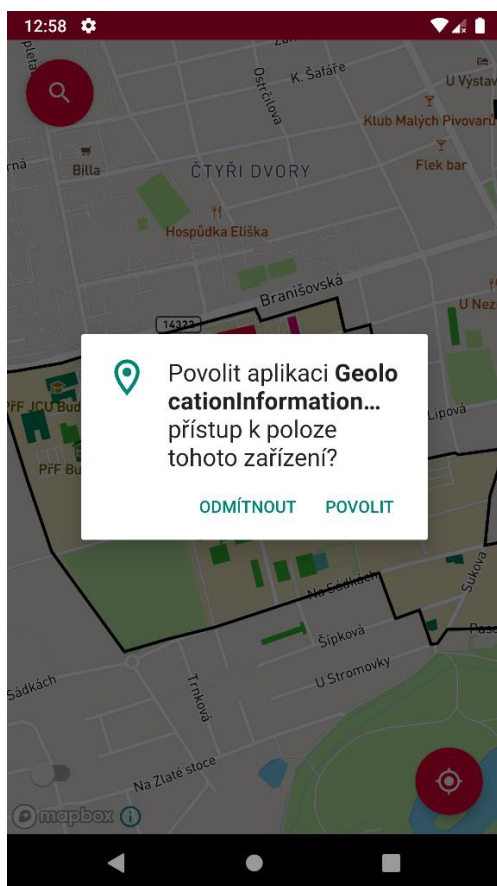


Posléze už pouze otevřením spustíte aplikaci.



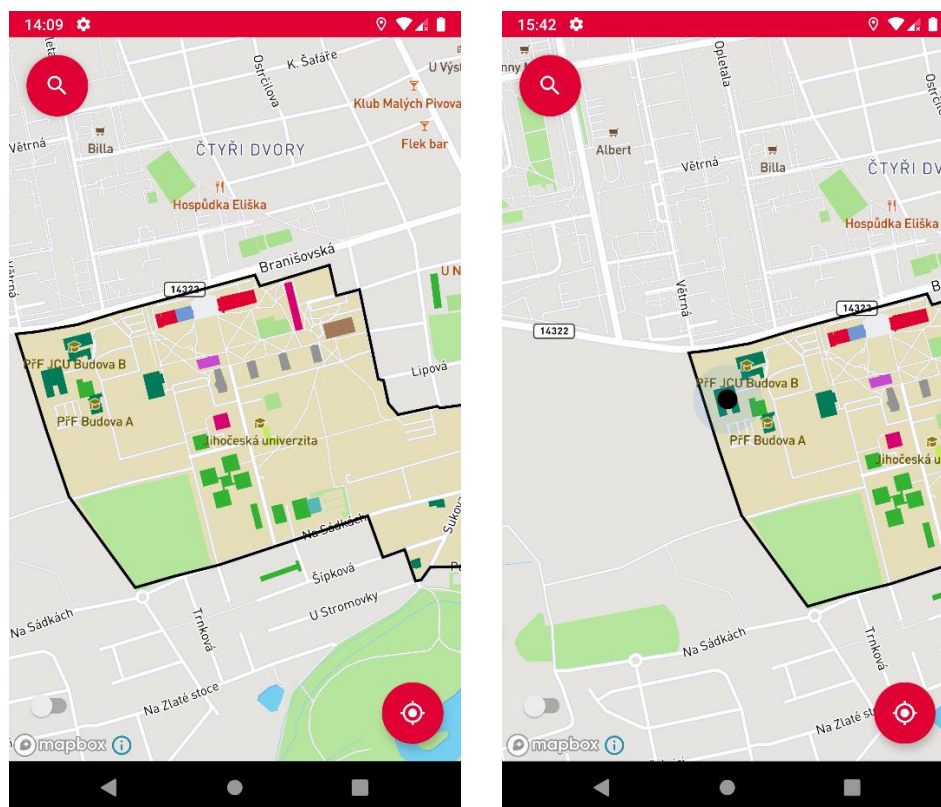
3 První spuštění aplikace

Při prvním spuštění aplikace bude uživatel požádán o povolení k získání polohy. Ve druhém kroku si uživatel vybere jazyk aplikace. Defaultně je nastaven anglický jazyk. Vybraný jazyk lze kdykoliv změnit v hlavním menu aplikace.



4 Mapa

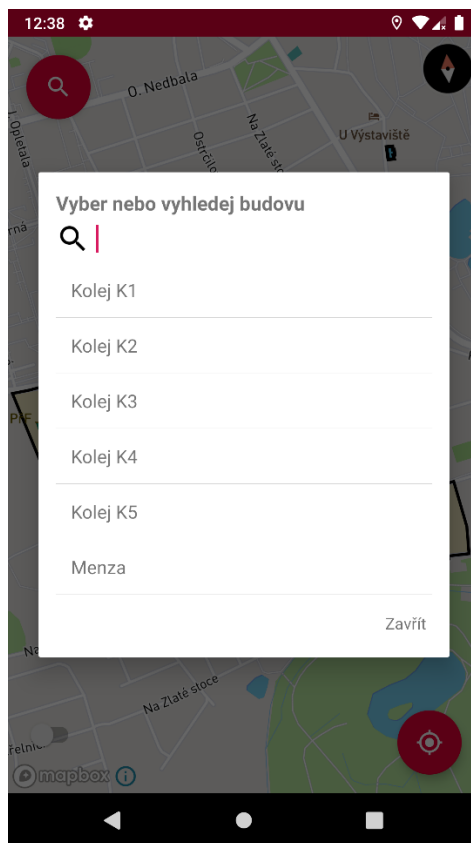
Tato obrazovka se otevře vždy po zapnutí aplikace jako první. Pokud má uživatel vypnuté získávání polohy, je kamera nastavena zhruba na střed kampusu. Pokud má polohu zapnutou, kamera se automaticky zaměří na jeho pozici.



Dále se na obrazovce nachází tři ovládací prvky.

Tlačítko pro vyhledávání budov

Tlačítko se nachází v levém horním rohu obrazovky. Po jeho stisku se zobrazí okno, které nabízí veškeré školní budovy v kampusu a v Českých Budějovicích. V tomto seznamu může uživatel listovat, nebo použít vyhledávání. Po stisku na vybranou budovu se nastaví pozice kamery na danou budovu.



Tlačítko pro aktualizaci polohy

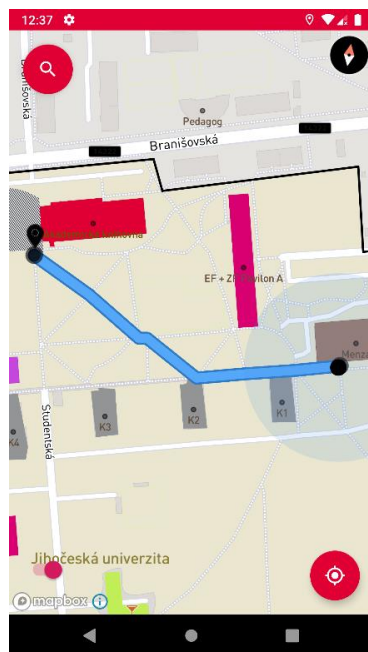
Tlačítko v pravém dolním rohu obrazovky.

Přepínač pro zapnutí/vypnutí navigace

Přepínač v levém dolním rohu obrazovky. Tímto přepínačem uživatel zapíná či vypíná funkci navigace. Postup pro vykreslení trasy je následující.

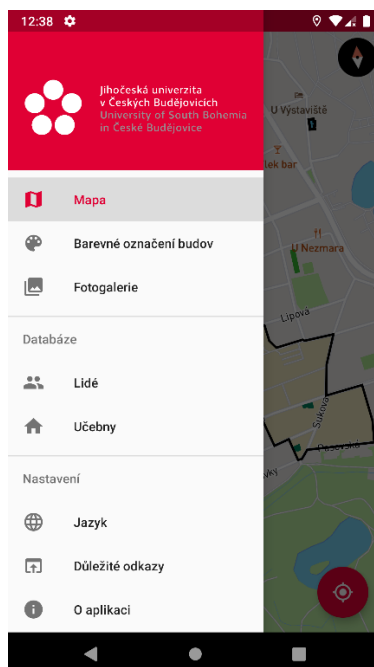
1. Povolit získávání polohy.
2. Přepínačem povolit navigaci.
3. Vybrat místo na mapě a kliknutím zvolit cílový bod. Následně se vykreslí trasa.

Na následujícím obrázku můžete vidět příklad vykreslení trasy od budovy menzy ke knihovně.



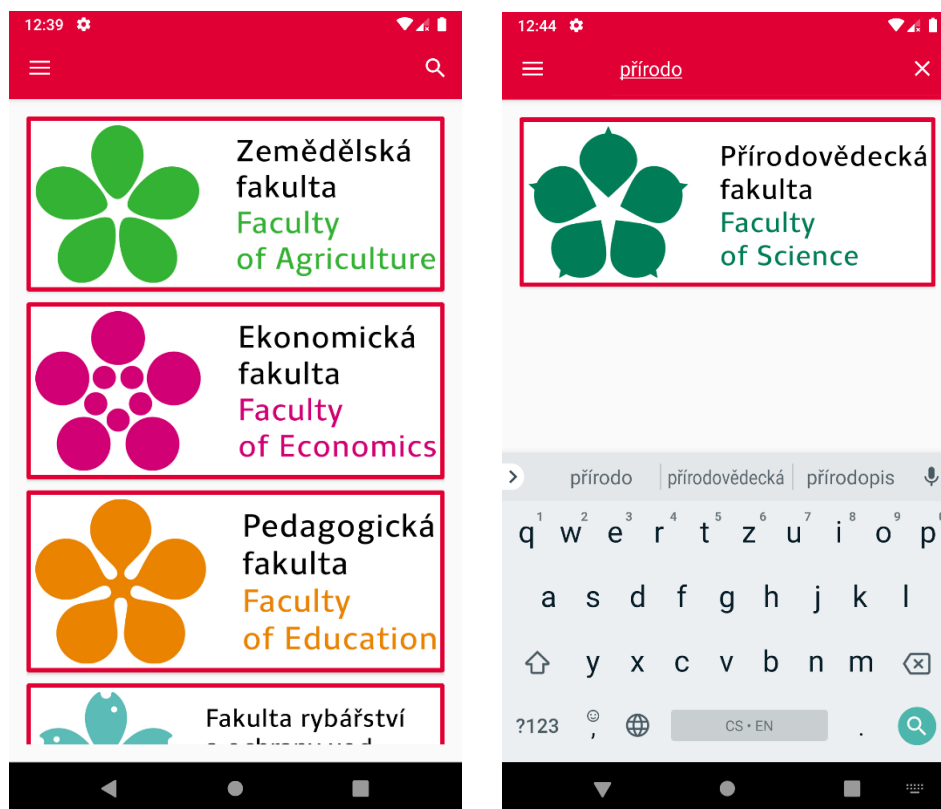
5 Hlavní menu aplikace

Do hlavního menu aplikace se lze dostat potažením od levé strany obrazovky. Toto gesto funguje ve všech obrazovkách v aplikaci.



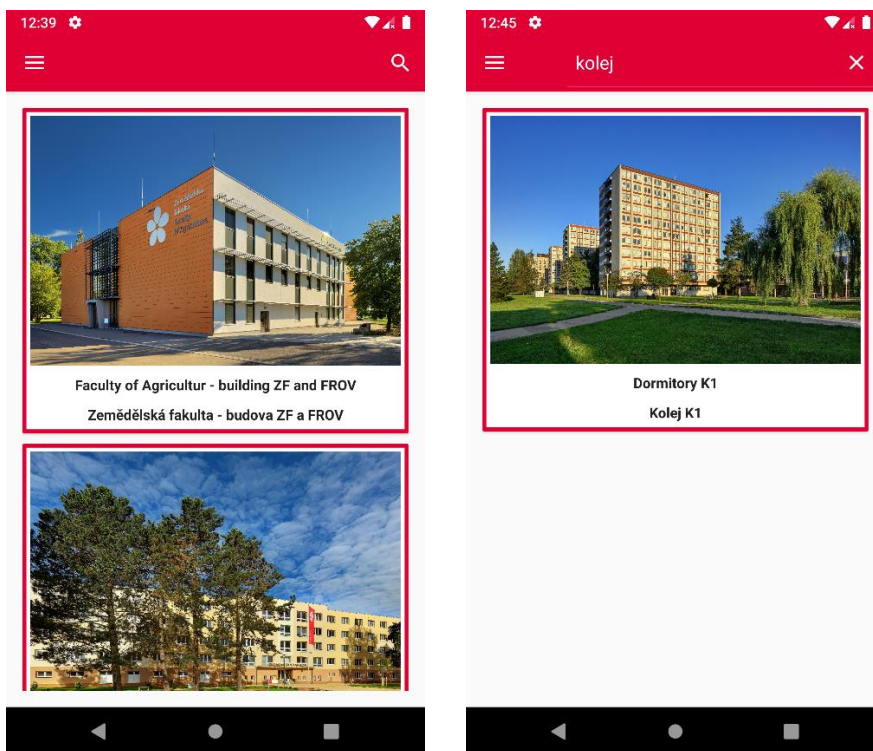
6 Barevné označení budov

Na této obrazovce si může uživatel prohlédnout loga fakult, respektive barevné označení, ze kterého vychází barva budov na mapě. V tomto seznamu obrázků může uživatel listovat nebo použít vyhledávání v horní liště.



7 Fotogalerie

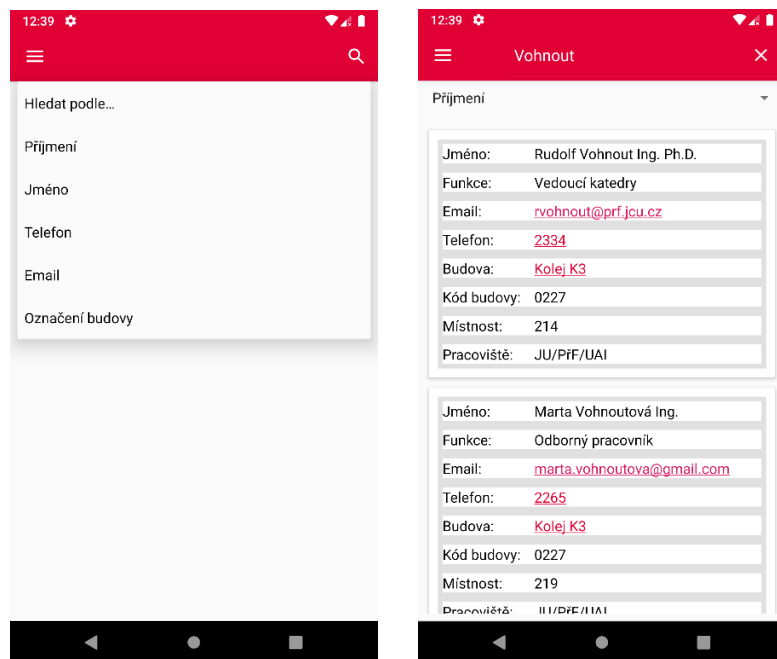
Logikou identická obrazovka jako ta předchozí. Jediný rozdíl je v tom, že zde uživatel najde fotografie budov. Listování a vyhledávání je totožné.



8 Lidé

Obrazovka zprostředkávající přístup do databáze osob. Postup pro vyhledání osoby.

1. Rozbalte možnosti vyhledávání kliknutím na „Hledat podle...“
2. Vyberte kritérium, podle kterého chce vyhledávat.
3. Zadejte textový řetězec, který chcete najít.
4. Potvrďte vyhledávání.



Upozornění: Pokud není k dispozici přístup k internetu, lze vyhledat pouze data, která již byla dříve načtena.

Po úspěšném vyhledání se zobrazí výsledky. U každé osoby jsou tři speciální textová pole, která po stisknutí vyvolají určitou akci.

Email

Po stisku emailové adresy, se otevře emailový klient v telefonu a do políčka adresát se vloží emailová adresa dané osoby.

Telefon

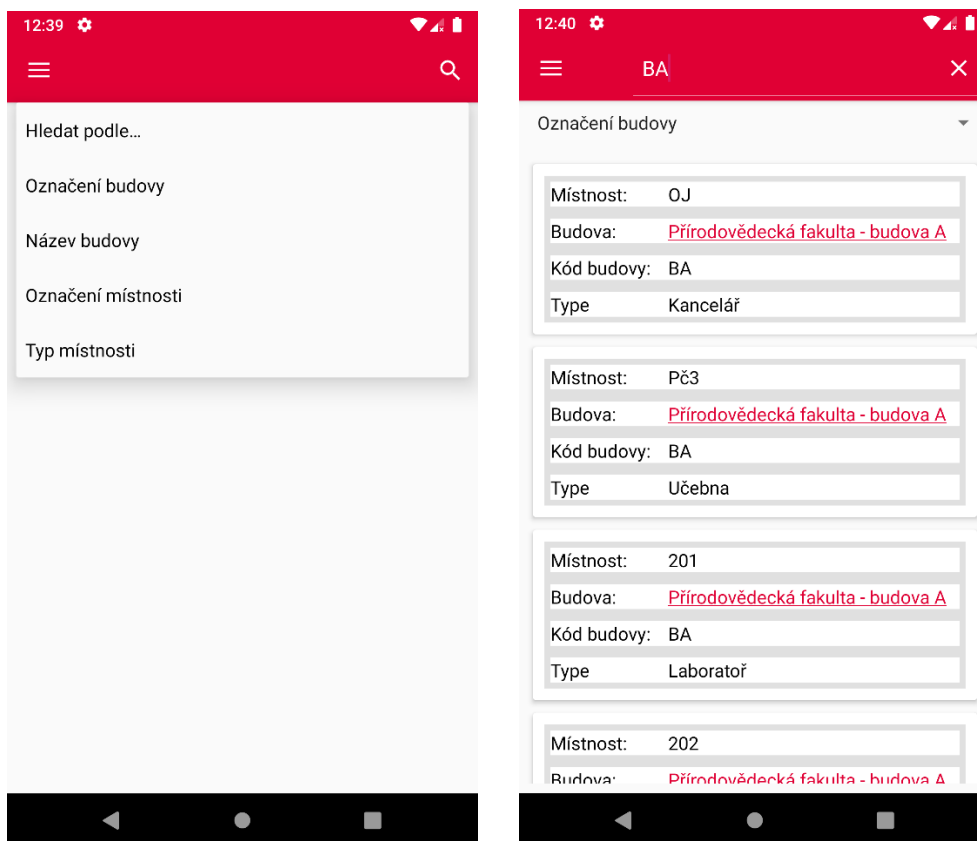
Po stisku telefonního čísla se otevře „telefon“ se zadaným číslem hledané osoby a uživatel tak může pohodlně provést hovor.

Budova

Po stisku budovy, se otevře obrazovka s mapou a kamera se zaměří na budovu, ve které hledaná osoba sídlí.

9 Učebny

Principiálně stejná obrazovka jako obrazovka *Lidé*. Rozdíl je v tom, co je zde možné vyhledat. Konkrétně učebny. Postup vyhledávání je taktéž stejný jako u výše zmíněné obrazovky.



Ve výsledcích je jeden ovládací prvek. Textové pole budova. Po stisku se otevře obrazovka s mapou a kamera se nastaví na danou budovu.

10 Jazyk

Zde se nachází nastavení jazyka aplikace. Jedná o totožné okno, které se ukáže při prvním spuštění aplikace.

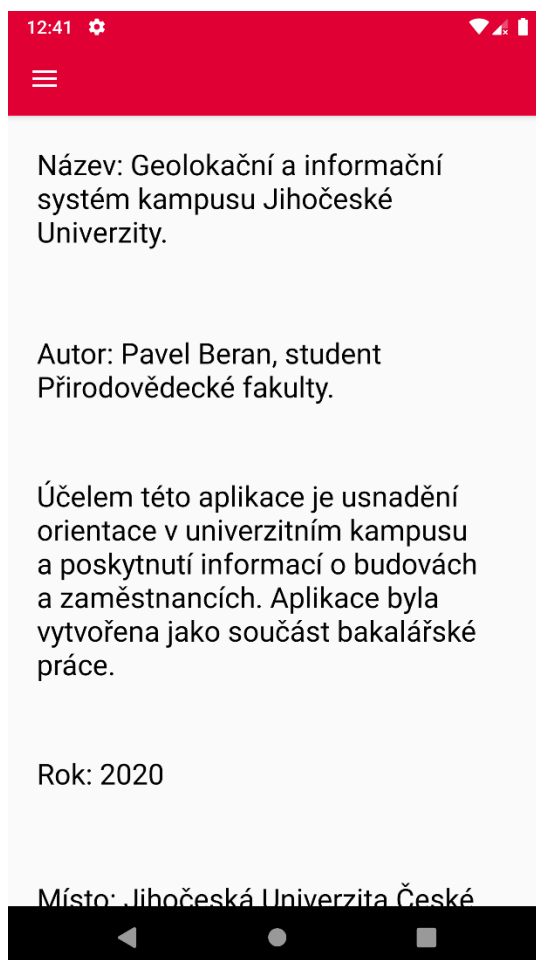
11 Důležité odkazy

Informativní obrazovka obsahující odkazy na školní weby a telefonní čísla pro integrovaný záchranný systém. Po stisku odkazu na web se v telefonu otevře prohlížeč s danou stránkou. Kliknutím na telefonní číslo se nevyvolá žádná reakce. A to z důvodu jejich důležitosti a vyhnutí se náhodnému vytočení.



12 O aplikaci

Čistě informativní obrazovka, obsahující základní informace o aplikaci.



13 FAQ

Vysvětlivky pro vyhledávací kritéria do databáze

Osoby:

- **Příjmení**
- **Jméno**
- **Telefon** – Telefonní číslo školní pevné linky. Příklad: „4100“. Univerzitní předvolba se nezadá.
- **Email** – Příklad: „rvohnout@prf.jcu.cz“
- **Označení budovy** – Příklad: „0227“ – zde se jedná o budovu Kolej K3.

Veškeré hledané řetězce nemusíte zadávat celé. Například stačí pokud při vyhledávání pomocí emailové adresy zadáte „rvoh“ a aplikace Vám vrátí všechny osoby, u kterých emailová adresa začíná na tento řetězec. V tomto případě bude výsledkem adresa rvohnout@prf.jcu.cz, respektive osoba, které adresa patří.

Toto platí u všech kritérií. Jediné omezení spočívá v minimálním počtu zadaných písmen – dvě. Na toto omezení Vás v případě zadání jednoho znaku upozorní sama aplikace.

Učebny:

- **Označení budovy** – Identická položka jako ve vyhledávání Osob.
- **Název budovy** – Příklad: „Přírodovědecká fakulta - budova b“.
- **Označení místnosti** – Příklad: „C1“. Toto označení vychází z označení místností v IS/STAG. (Stejně označení, které obsahuje rozvrh).
- **Typ místnosti** – Vyhledat můžete následující druhy místností:
 - **Kancelář**
 - **Učebna**
 - **Laboratoř**
 - **Zasedací místnost**

- **Pracovna**
- **Posluchárna**
- **Učebna s it**
- **Seminární učebna**
- **Počítačová učebna**
- **Rýsovna**
- **Jiná**

Vyhledávání funguje na stejném principu jako u vyhledávání osob s rozdílem, že minimální počet zadaných písmen pro vyhledávání je jedno.

Kontakt na autora

Vaše dotazy či návrhy pro zlepšení aplikace, může zasílat na emailovou adresu:

jcumap@gmail.com

Příloha: Administrátorská dokumentace

1 Google účet

Google účet, kterým se lze dostat do webového rozhraní databáze a celé platformy Cloud Firestore.

Odkaz na web:

<https://console.firebase.google.com/u/1/project/androidjcumap/database/firestore/data>

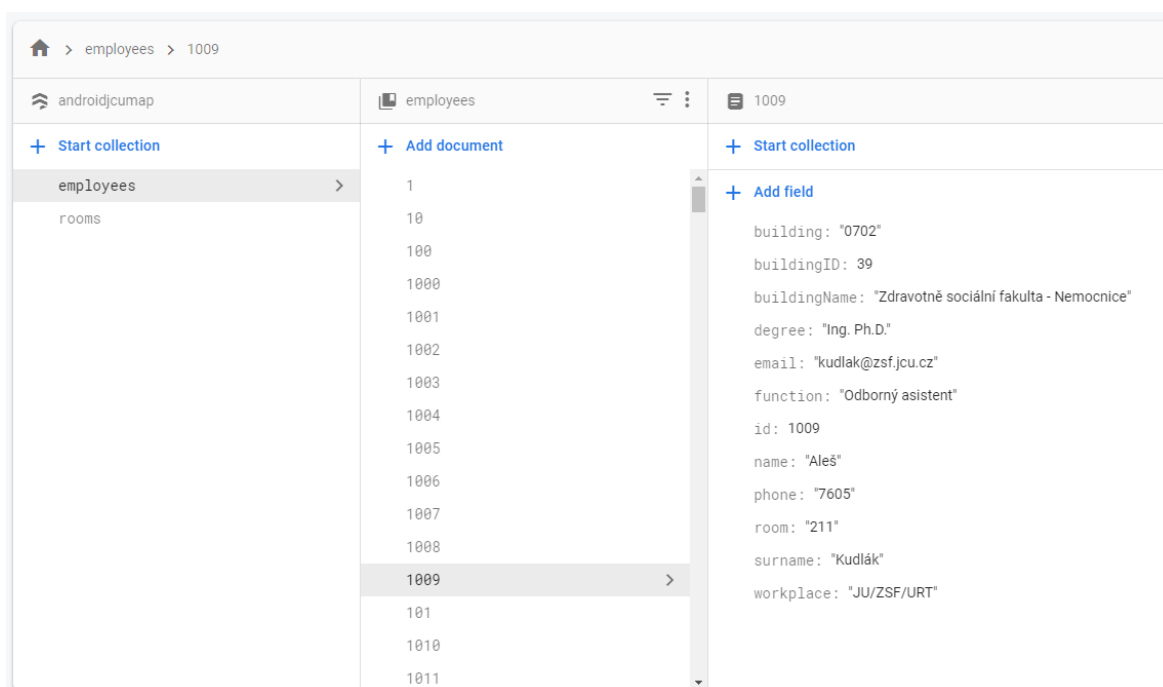
Jméno: jcumap@google.com

Heslo: *****

2 Správa dat

Firestore

Veškerou úpravu dat lze provést ve webovém rozhraní. Pro jednotlivé mazání, přidávání či úpravy záznamů jde o efektivní způsob správy.



Skript

Při větších změnách lze využít přiložený skript, který nahraje JSON soubor do databáze. Pro jeho využití je potřeba nainstalovat si Node.js. Tento skript je spíše záložní řešení pro nahrávání dat do databáze. Primárním kanálem pro úpravy dat by mělo být webové rozhraní a skript použít pouze v případě nutnosti nahrávání velkého množství nových osob nebo místností.

Celý projekt obsahuje čtyři JSON soubory a jeden JavaScript soubor.

JSON:

- serviceAccountKey.json automaticky vygenerováno.
- package-lock.json – automaticky vygenerováno.
- package.json – automaticky vygenerováno.
- convertedcsv.json – json soubor, který chceme nahrát do databáze.

Na následujícím obrázku můžete vidět příklad JSON souboru obsahující osoby, který se poté importuje do databáze.

```
{
  "1": {
    "id": 1,
    "name": "Monika",
    "surname": "Cimlová",
    "degree": "NULL",
    "function": "Vrátný",
    "workplace": "JU/KAM/028201",
    "buildingName": "Kolej K1",
    "building": "0225",
    "room": "00002",
    "phone": "4100",
    "email": "NULL",
    "buildingID": 1
  },
  "2": {
    "id": 2,
    "name": "Petra",
    "surname": "Dvořáková",
    "degree": "NULL",
    "function": "Vedoucí koleje",
    "workplace": "JU/KAM/028201",
    "buildingName": "Kolej K1",
    "building": "0225",
    "room": "101",
    "phone": "4101",
    "email": "pdvorakova02@jcu.cz",
    "buildingID": 1
  },
}
```

JavaScript:

```
const admin = require('./node_modules/firebase-admin');
// file with settings from Firestore
const serviceAccount = require("./serviceAccountKey.json");
// name of the json file
const data = require("./convertcsv.json");
// name of the collection
const collectionKey = "employees";
admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://beran-67377.firebaseio.com"
});
const firestore = admin.firestore();
const settings = { timestampsInSnapshots: true };
firestore.settings(settings);
if (data && (typeof data === "object")) {
  Object.keys(data).forEach(docKey => {
    firestore
      .collection(collectionKey)
      .doc(docKey)
      .set(data[docKey])
      .then((res) => {
        console.log("Document " + docKey + " successfully written!");
      }).catch((error) => {
        console.error("Error writing document: ", error);
      });
  });
}
}
```

Kompletní projekt je přiložen a vše je nastaveno. Tudíž se pouze stačí v konzoli přepnout do příslušného adresáře a zadat příkaz: *node index.js*

Pokud bude potřeba měnit data v kolekci *rooms*, jsou k tomu zapotřebí dva kroky.

1. Přepsat třetí řádek a místo *convertcsv.json* zadat správný soubor.
2. Přepsat čtvrtý řádek a místo *employees* zadat *rooms*.