

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Interaktivní mapa kampusu ZČU na platformě Android

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Lukas FREY**
Osobní číslo: **A18B0198P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Interaktivní mapa kampusu ZČU na platformě Android**
Zadávací katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prozkoumejte vybrané mobilní aplikace různých univerzit, jejichž cílem je usnadnit orientaci v kampusu a poskytování informačních zdrojů.
2. Vyberte vhodné funkcionality, které by měla interaktivní mapa kampusu poskytovat. Při výběru zohledněte určení aplikace pro mobilní zařízení.
3. Navrhněte interaktivní mapu kampusu ZČU pro platformu Android.
4. Navrženou aplikaci realizujte, ověřte její funkcionality a navrhněte možná rozšíření.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Ladislav Pešička**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **4. října 2021**
Termín odevzdání bakalářské práce: **5. května 2022**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 14. října 2021

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 25. června 2022

Lukas Frey

Poděkování

Tímto bych rád poděkoval vedoucímu mé bakalářské práce panu Ing. Ladislavu Pešíčkovi, za všestrannou pomoc, množství cenných rad, doporučení, připomínek a zároveň za velkou trpělivost s obdivuhodnou ochotou při konzultacích poskytnutých ke zpracování této práce. Na závěr bych chtěl poděkovat také mé rodině a přátelům, kteří mi byli podporou po celou dobu mého studia.

Abstract

Interactive map of the ZČU campus for Android. This paper contains research of universities and their approach at providing an interactive map and information sources to their students, employees, and guests. Based on findings from the research, convenient features were selected for a mobile Android app, which provides an interactive map of the campus of the University of West Bohemia in Pilsen. The app was then developed, and the features were verified based on a set of defined testing scenarios. Finally, the paper describes possible enhancements and outlines a potential future version of the app.

Abstrakt

Práce se zabývá průzkumem aplikací různých univerzit pro poskytování interaktivní mapy a informačních zdrojů studentům, zaměstnancům a hostům univerzity. Z poznatků plynoucích z průzkumu byla vybraná vhodná funkcionalita pro mobilní aplikaci na platformě Android poskytující interaktivní mapu kampusu Západočeské univerzity v Plzni na Borech. Aplikace byla následně realizována a její funkcionalita ověřena na základě definovaných testovacích scénářů. V neposlední řadě práce obsahuje popis možných rozšíření a nastiňuje tak případnou budoucí vizi aplikace.

Obsah

1	Úvod	11
1.1	Motivace	11
1.2	Cíl práce	11
2	Průzkum aplikací ostatních univerzit	12
2.1	Mazemap	12
2.1.1	Případová studie na NTNU	13
2.2	Berkeley Mobile App	14
2.3	Misericordia University	15
2.4	University of York	16
2.5	University of Manchester	17
2.6	Shrnutí	18
3	Platformy	19
3.1	Web	19
3.2	iOS	20
3.3	Android	21
3.3.1	Verze Androidu	22
3.3.2	API level	23
3.3.3	Výběr API levelu	23
3.4	Shrnutí	24
4	Možnosti vývoje aplikace	25
4.1	Přístupy k mobilnímu vývoji	25
4.1.1	Nativní vývoj	25
4.1.2	Multiplatformní vývoj	25
4.2	React Native	26
4.3	Flutter	26
4.4	Unity	26
4.4.1	C#	27
4.5	Shrnutí	27
5	Funkcionality aplikace	28
5.1	Interaktivní mapa	28
5.2	Body zájmů	28
5.3	Vyhledávání místností	28

5.4	Informace o místnostech	29
5.5	Shrnutí	29
6	Modelování kampusu	30
6.1	3D Modelování	30
6.2	Nástroje pro tvorbu 3D modelu	30
6.2.1	Sketchup	30
6.2.2	3ds Max	31
6.2.3	Blender	31
6.3	Proces tvorby	31
6.4	Shrnutí	32
7	Uchování dat	33
7.1	Ukládání do preferencí	33
7.2	Ukládání do souborů	33
7.2.1	XML	33
7.2.2	JSON	34
7.3	Databáze	34
7.3.1	MySQL	34
7.3.2	PostgreSQL	34
7.3.3	SQLite	34
7.4	SQL	35
7.5	ORM	36
7.5.1	NHibernate	37
7.5.2	Fluent NHibernate	37
7.6	Datové modely	37
7.6.1	Room	38
7.6.2	Faculty	38
7.6.3	Ostatní modely	38
7.7	Migrace	39
7.7.1	Reflexe	41
7.8	Shrnutí	41
8	Webové služby IS-STAG	42
8.1	API	42
8.2	SOAP	43
8.3	REST	44
8.4	Postman	44
8.5	Koncové body	45
8.5.1	Získání místností	45

8.5.2	Získání seznamu domén	45
8.5.3	Získání číselníků	46
8.5.4	Získání rozvrhů místností	46
8.6	Mapování na databázový model	46
8.7	Shrnutí	47
9	Struktura a řešení aplikace	48
9.1	Struktura projektu	48
9.2	Struktura kódu	48
9.2.1	Třída App	49
9.2.2	Třída CameraController	50
9.2.3	API Manager	51
9.3	Řešení problémů	53
9.3.1	Potíže s otevřenou klávesnicí	53
9.3.2	SQLite	53
9.3.3	Stahování aktualizace dat	53
9.4	Distribuce	54
9.5	Shrnutí	54
10	Rozšíření	55
10.1	Podrobnější informace	55
10.2	Oblíbené	55
10.3	Přidání vlastních bodů zájmů	55
10.4	Cachování rozvrhů	55
10.5	Půdorysy budov	56
10.6	Rozšíření mapy	56
10.7	Vnitřní navigace	56
10.8	Shrnutí	57
11	Testování	58
11.1	Hledání místnosti	58
11.2	Manipulace s body zájmů	58
11.3	Otočení obrazovky	59
11.4	Připojení k internetu	59
11.5	Různá zařízení	60
11.5.1	Huawei nova 3	60
11.5.2	Google Pixel 6 Pro	60
11.5.3	Samsung Galaxy S7	61
11.5.4	Samsung Galaxy Tab S3 9.7	62
11.6	Shrnutí	63

12 Závěr	64
Literatura	65

1 Úvod

1.1 Motivace

Touto prací bych chtěl usnadnit především nově nastupujícím studentům ZČU orientaci na univerzitním kampusu v Plzni na Borech. Areál je poměrně rozsáhlý a zvláště pro nové studenty, zahraniční studenty (např. z programu Erasmus) a nebo hosty je těžké se v areálu zorientovat. Cílem aplikace je těmto uživatelům pomoci. Sám bych podobnou aplikaci v počátcích mého studia na univerzitě uvítal a z tohoto důvodu jsem se pro svou bakalářskou práci rozhodl vývojem této aplikace.

Aplikaci budu vyvíjet pro platformu Android a kolega Stepan Mocjak bude vyvíjet podobnou aplikaci pro iOS. Takto bude pokryta většina mobilních zařízení.

Aplikace má za úkol nabídnout uživatelům 3D vizualizaci plzeňského kampusu na Borech, což usnadní zorientování v prostředí kampusu. Dále by měla aplikace umožnit uživatelům zobrazení základních informací o místnostech univerzity. Jedná se o informaci o katedře, pod kterou spadá, patro a budovu ve kterém se místnost nachází, typ a případně kapacitu učebny.

1.2 Cíl práce

Cílem této práce je vytvoření aplikace poskytující interaktivní mapu kampusu. K dosažení tohoto cíle budou nejprve prozkoumány obdobné aplikace jiných univerzit a následně zvoleny vhodné technologie pro vytvoření aplikace. Dále bude práce obsahovat popis vývoje aplikace a implementovaných funkcionalit. Další součástí práce bude ověření funkčnosti aplikace a popsání možných budoucích rozšíření.

2 Průzkum aplikací ostatních univerzit

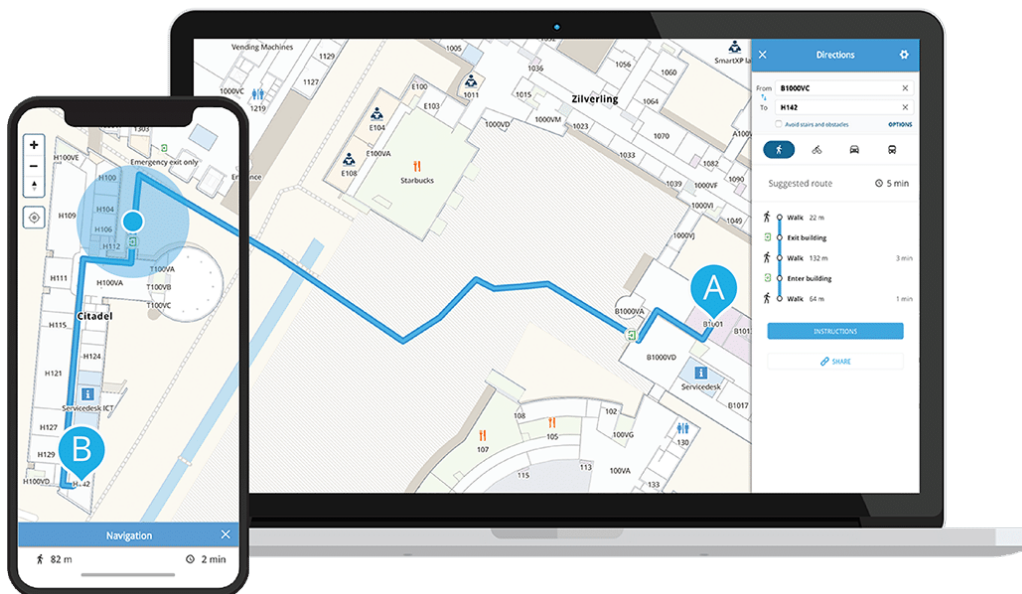
Inspiraci jsem hledal v aplikacích třetích stran, určených pro orientaci v kampusu příslušné univerzity. Některé aplikace mají i širší použití, např. aplikace Mazemap se nezaměřuje striktně na univerzity, ale také na hotely, nemocnice a další.

V následujících kapitolách prozkoumám několik vybraných aplikací, které mají za cíl poskytnout uživatelům nějakou mapu či slouží jako informační zdroj.

2.1 Mazemap

Aplikaci Mazemap [23] používá hned několik celosvětově známých univerzit. Mezi nimi jsou například University of Cambridge, Norwegian University of Science and Technology (dále NTNU) a jiné. Aplikace má mnoho funkcionalit, ale její ústřední funkcí je vizualizace kampusu a navigace v něm (viz obr. 2.1). Mimo jiné obsahuje také různé body zájmu a možnost zobrazit půdorys budovy. Budovy jsou nastavitelné a tak lze například pro soukromé budovy vypnout funkci zobrazení půdorysu veřejně. Dále podporuje jak vnitřní tak i vnější navigaci. Pro vnější navigaci využívá systém GPS a pro vnitřní podporuje mnoho různých řešení, jako například triangulace Wi-Fi či využití Bluetooth beaconů.

Mazemap tak slouží jako inspirace a představuje možnou budoucí podobu mé aplikace nad rámec bakalářské práce.



Obrázek 2.1: Ukázka aplikace Mazemap

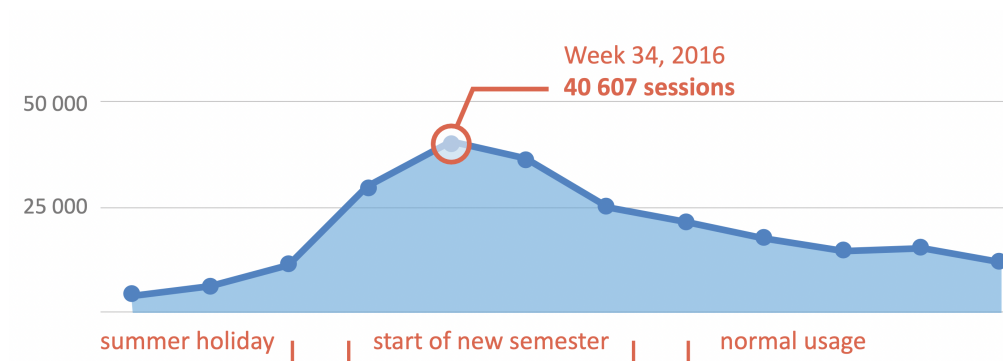
Zdroj: <https://www.mazemap.com>

2.1.1 Případová studie na NTNU

Jedna z prvních univerzit využívající aplikaci Mazemap je právě NTNU, pro kterou existuje ohledně užití aplikace případová studie z roku 2017. Mimo jiné je ve studii vidět, jak často studenti využijí mobilní mapu a největší využití má právě na začátku nového semestru.

Aplikaci používá univerzita již od roku 2011 až do současnosti. Ke dni zveřejnění studie byla univerzita NTNU, čítající 17 kampusů a 144 budov o celkové rozloze přibližně 700 000 m², zcela pokryta mobilní navigací aplikací [72].

Velmi zajímavým poznatkem jsou také výsledky z Google Analytics, které NTNU ve své případové studii zveřejnila (viz obr. 2.2). Ze statistiky vyplývá, že největší potíže mají studenti na začátku nového semestru. Studenti mají nové rozvrhy, jsou zapsaní do nových předmětů, které se pravděpodobně vyskytují v budovách a učebnách ve kterých se studenti v dřívějších semestrech nepohybovali a tak si neví rady, kde se učebny nacházejí. Právě v těchto počátcích studenti sáhli po aplikaci Mazemap, která uživatelům pomohla učebny nalézt [72].



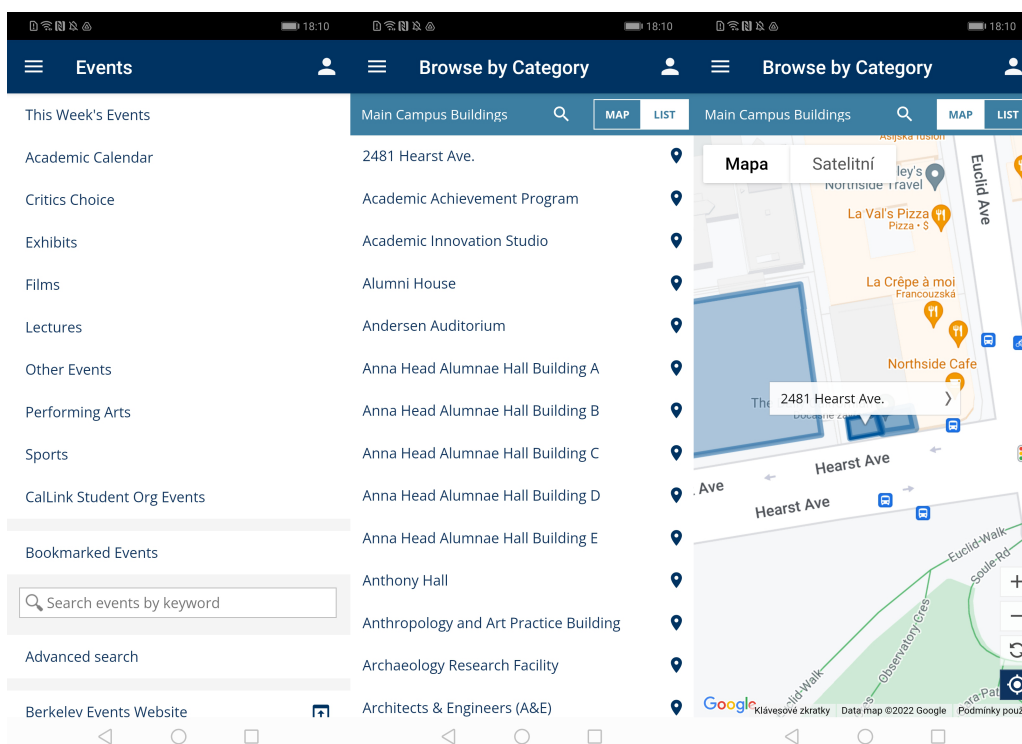
Obrázek 2.2: Statistika z případové studie na NTNU [72]

Zdroj: <https://www.jottacloud.com/p/mazemap/e9dd2117a84a4d059a0391ef4a0aac74>

2.2 Berkeley Mobile App

Dalším mobilním průvodcem studenta je studentská aplikace univerzity Berkeley. Slouží jako mapa kampusu a zároveň jako informační zdroj pro studenty [6].

Studenti mohou v aplikaci získat aktuality o univerzitě, informace o univerzitních knihovnách, sociálních akcích a veřejné dopravě ve formě textových článků uvnitř aplikace. Dále lze spustit navigaci do veřejně přístupných budov [7]. Navigace se spustí uvnitř aplikace ve vloženém zobrazení Google Maps. Jak aplikace vypadá lze vidět na obrázku 2.3.



Obrázek 2.3: Snímky obrazovky z aplikace UC Berkeley Mobile

Zdroj: https://play.google.com/store/apps/details?id=edu.ucberkeley.comhl=en_US

2.3 Misericordia University

Na této univerzitě představuje interaktivní mapa kampusu jednoduchou webovou aplikaci, ve které si mohou studenti zobrazit 2D vizualizaci kampusu (viz obr. 2.4) a zjistit informace o jednotlivých budovách a místech v kampusu. Aplikace nepodporuje hledání místností.

Webová stránka není příliš uživatelsky přívětivá pro zobrazení na mobilu, jelikož se jedná o 2D obrázek, který není nijak upravený pro mobilní zařízení, například možností přiblížení pro lepší čitelnost na malé obrazovce.



Obrázek 2.4: Interaktivní mapa Misericordské univerzity

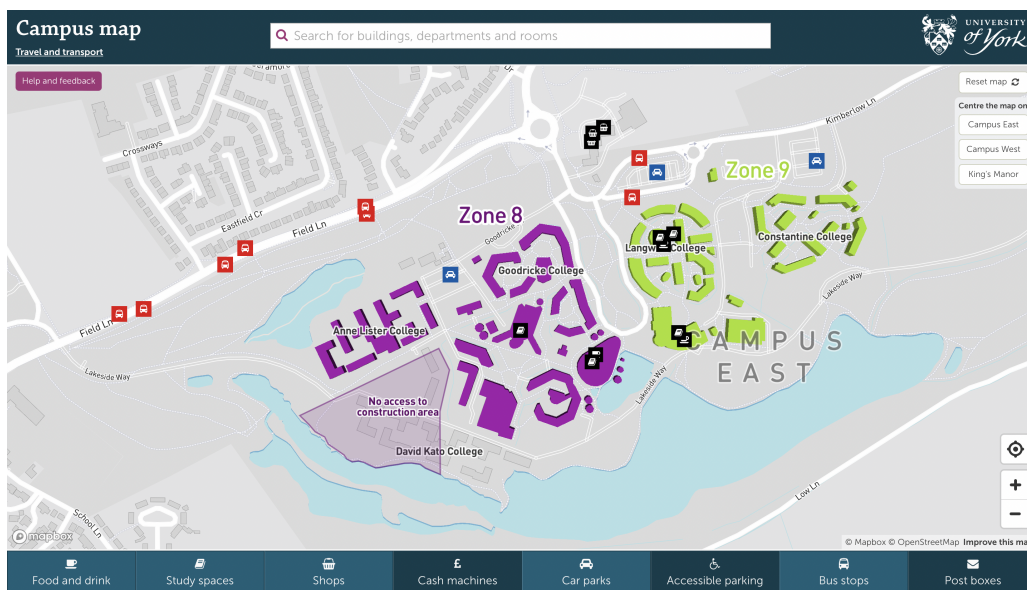
Zdroj: <https://www.misericordia.edu/admissions/admissions/interactive-campus-map>

2.4 University of York

V tomto případě se jedná také o webovou interaktivní mapu kampusu. Zde se ale nejedná o 2D obrázek kampusu, nýbrž o upravené zobrazení mapy z Open Street Maps (OSM) s přidanou vrstvou nad mapou, která zvýrazňuje univerzitní kampus a budovy v něm (viz obr. 2.5).

Na mapě lze zobrazit štítky, které poukazují na různá zajímavá místa, jako například kavárny nebo studovny. Nelze zde hledat místnosti.

Web je uživatelsky přívětivý i na mobilních zařízeních.



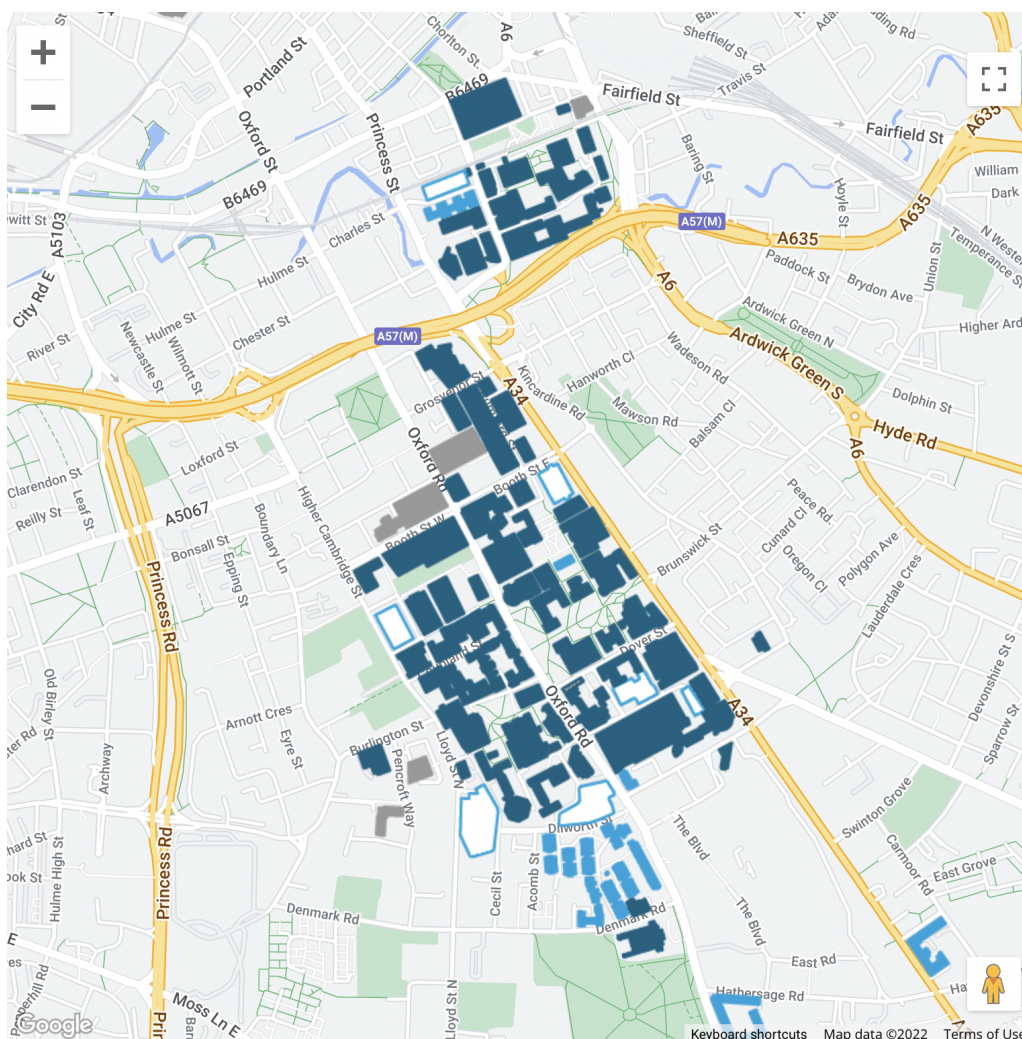
Obrázek 2.5: Interaktivní mapa Yorkské univerzity

Zdroj: <https://www.york.ac.uk/map/>

2.5 University of Manchester

Podobně funguje mapa na Manchesterské univerzitě. Jedná se taktéž o interaktivní vizualizaci kampusu ve webovém prohlížeči (viz obr. 2.6). Zde ovšem místo Open Street Maps využili Google Mapy. I zde je možné zobrazit různé body zájmů, které se zobrazí na mapě ve formě štítků. Na rozdíl od předešlých map však obsahuje i vyhledávač pro užitečné body zájmů.

I v tomto případě je web funkční i na mobilních zařízeních a s mapou lze zacházet dobře i na mobilu. Ovšem mobilní verze již neobsahuje vyhledávač pro body zájmů.



Obrázek 2.6: Interaktivní mapa Manchsterské univerzity

Zdroj: <https://www.manchester.ac.uk/discover/maps/interactive-map/>

2.6 Shrnutí

Ze zkoumaných aplikací univerzit využila každá univerzita trochu odlišnou metodu pro zobrazení kampusu. Zatímco NTNU využila mobilní aplikaci Mazemap, Kalifornská univerzita v Berkeley se rozhodla využít vlastní mobilní aplikaci.

Misericordská univerzita využívá jednoduchou webovou aplikaci s 2D interaktivní mapou. A v neposlední řadě Yorkská univerzita a Manchsterská univerzita využily 3D mapu, do které přidaly vlastní vrstvu pro vykreslení informací a barevné označení svého kampusu.

3 Platformy

Správná volba platformy je závislá na různých kritériích, které určují nejvhodnější platformu pro aplikaci.

Jedním z hlavních kritérií volby platformy je cílová skupina aplikace. Aplikace slouží jako průvodce univerzitního kampusu na Borech, primární skupinou budou tedy studenti prvních ročníků a sekundárně také všichni studenti, kteří nastupují do nového semestru. Lze totiž předpokládat, že i studenti vyšších ročníků budou muset navštívit nějakou novou učebnu v budově, kterou ještě neznají.

Dále lze předpokládat, že studenti se budou často pohybovat na veřejnosti mezi několika budovami ZČU a nemusí mít vždy přístup k internetu. Jelikož se jedná o studenty, mohou být finančně závislí na rodičích a proto nemusí mít vždy nejnovější zařízení. Určitě bude také rozdíl mezi studenty různých oborů, kde lze předpokládat, že studenti technicky zaměřených oborů budou vlastnit zařízení s novějšími technologiemi, než-li studenti humanitních oborů.

Ze statistiky za rok 2021 [55] vyplývá, že více než polovina uživatelů využívá k prohlížení internetu svá mobilní zařízení, přičemž toto číslo stále roste.

Je tedy rozumné vyvíjet aplikaci pro mobilní zařízení. A protože již paralelně vzniká obdobná aplikace na platformě iOS, bude tato aplikace určena pro Android, aby se tak pokryla co možná největší množina uživatelů.

Pro úplnost se v následujících kapitolách nejprve stručně zmíním i o ostatních platformách, než-li Android. Následně bude podrobněji popsána platforma Android a její výhody a nevýhody.

3.1 Web

Častým problémem je, že uživatelé nechtějí být nuceni do stahování aplikace a raději dávají přednost jednorázovému zobrazení například ve webovém prohlížeči. S narůstajícím počtem cenově dostupných mobilních internetových tarifů, přístupům k Wi-Fi a nepřetržitým přístupem k internetu je proto vhodné vzít v potaz i tuto platformu.

Výhodou webové platformy je nezávislost na zařízení a možnost spuštění aplikace jak na mobilních zařízeních nezávisle na operačním systému (iOS / Android), tak i na počítačích opět nezávisle na operačním systému (MacOS / Windows / Linux).

Web má ale i své nevýhody, například přestože vývojář není limitován koncovým zařízením uživatele, stále musí brát v potaz dvě věci: prohlížeč ve kterém uživatel webovou aplikaci zobrazí a velikost obrazovky uživatelského zařízení.

Prohlížeče běží na různých vykreslovacích softwarech a webová aplikace, která funguje na jednom prohlížeči nemusí fungovat na druhém prohlížeči. Hlavními hráči na trhu jsou dle statistik z roku 2021 [10] prohlížeče Google Chrome s podílem 64,45 %, Safari s podílem 18,9 %, Firefox s podílem 3,62 %, Edge s podílem 3,59 % a Opera s podílem 2,24 % (viz obr. 3.1). V České republice je to dost podobné [9] a drobné výkyvy lze zanedbat. Podporou těchto prohlížečů lze pokrýt přes 90 % webových uživatelů, což považují za dobrý kompromis mezi časovou náročností vývoje, přístupem k moderním funkcím webu a dosažitelností finální webové aplikace.



Obrázek 3.1: Podíl prohlížečů na světovém trhu za rok 2021 [10]

Podobně je tomu tak u velikosti obrazovek. Vývojář musí počítat s tím, že si uživatel může web zobrazit na počítači s velkým monitorem, na notebooku, na mobilním zařízení nebo na ostatních zařízeních jako jsou televize a hodinky.

3.2 iOS

Apple iOS je proprietární operační systém pro iPhony a ostatní mobilní zařízení vytvořené společností Apple. iOS je založený na operačním systému

MacOS, který běží na desktopových a laptopových počítačích od Applu. Apple si velmi zakládá na bezpečnosti a uživatelské přívětivosti svých systémů. Všechny softwarové a hardwarové komponenty jsou navrženy tak, aby spolupracovaly mezi sebou a s řadou dalších Apple produktů [5, 67].

První verze iOSu vyšla v červnu roku 2007. V té době se ještě operační systém jmenoval pouze iPhone OS. Ke změně názvu na iOS došlo až později od roku 2010 s příchodem iOS 4 [19, 56]. Ovšem samotný název iOS existoval již od roku 2008, kdy Apple publikovalo sadu vývojářských nástrojů (Software Development Kit, SDK), která ostatním vývojářům umožňovala vyvíjet aplikace pro mobilní zařízení firmy Apple [21].

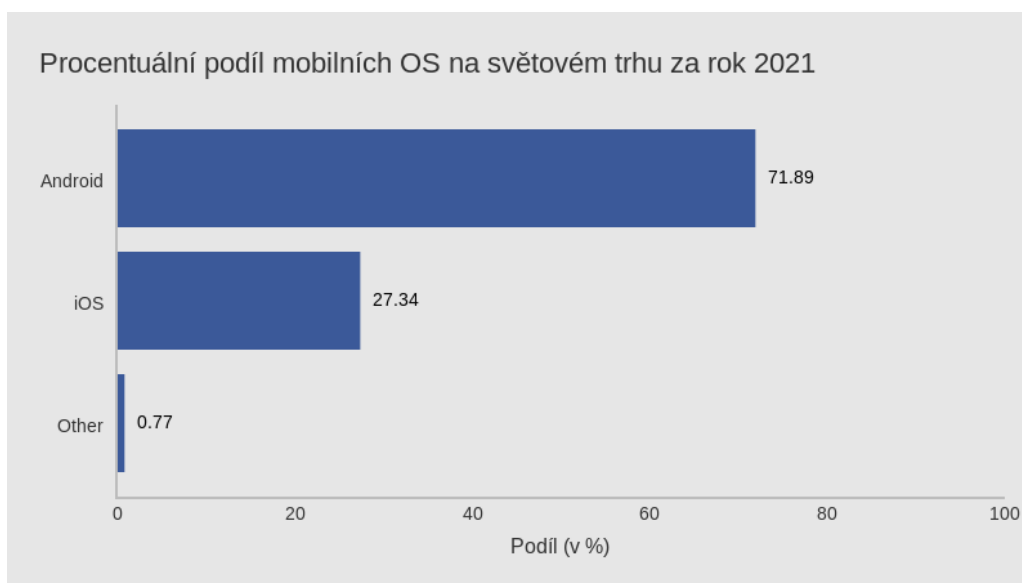
K dnešnímu dni je nejnovější verze iOSu verze iOS 15 [20] a nová verze iOS 16 vyjde ve finální verzi během podzimu [18].

Kromě prémiových produktů se Apple také pyšní svým rozsáhlým ekosystémem, který zlepšuje uživatelskou přívětivost a je automatickým vedlejším produktem toho, že uživatel vlastní několik zařízení od firmy Apple. Jádrem celého ekosystému je Apple ID. Pod tímto identifikátorem si zákazník registruje veškerá svá zařízení [65].

3.3 Android

Android je operační systém určený pro mobilní zařízení vlastněný společností Google. Google Android ovšem nevyvinul, nýbrž odkoupil. Z počátku byl Android vlastněn stejnojmennou společností Android Inc. z kalifornského Palo Alto [69].

Dnes je Android nejpobulárnějším operačním systémem pro mobilní zařízení (viz obr. 3.2). Jeho podíl na světovém trhu za rok 2021 činí 71,89 % [25]. V České republice vítězí Android dokonce s podílem 75,38 % za uplynulý rok [24]. Jedna z mála zemí, ve které Android nedrží první místo jsou Spojené státy americké, kde má podíl na trhu 41,11 % [26].



Obrázek 3.2: Podíl mobilních operačních systému na světovém trhu za rok 2021 [25]

3.3.1 Verze Androidu

První oficiální verze Androidu vyšla v roce 2008. V současné době se Android snaží vydávat novou verzi systému každý rok [82]. Počínaje verzí 1.5 začal Google pojmenovávat Android kromě klasického numerického číslování také abecedně podle nějakého dezertu. Tak vznikly například názvy jako Android 1.5 Cupcake nebo Android 4 Ice Cream Sandwich.

Od tradice pojmenování verzí systému podle dezertů Google upustil od verze 10 (nejprve Android Q). Aktuální verze Androidu je 12 a ve druhé polovině roku 2022 je plánované zveřejnění verze Android 13 [54].

Aktuální seznam verzí tohoto systému s podíly na světovém trhu je uvedený v tabulce 3.1.

Verze / Název	API Level	Podíl na trhu (za rok 2021)
5.1 Lollipop	22	2,72 %
6.0 Marshmallow	23	4,45 %
7.0 Nougat	24	3,8 %
7.1 Nougat	25	2,51 %
8.0 Oreo	26	3,96 %
8.1 Oreo	27	8,59 %
9.0 Pie	28	16,73 %
10.0 Q (později jen 10)	29	34,82 %
11.0	30	20,11%
12.0	31	Na trhu až od 2022
Ostatní		2,31 %

Tabulka 3.1: Verze Androidu a jejich podíl na trhu v roce 2021 [2]

3.3.2 API level

Kromě veřejně známých verzí Androidu existuje také verze API levelu, která určuje verzi frameworku, který Android poskytuje pro práci se systémem.

Pro aplikaci lze nastavit tři hodnoty API levelu: minimální, maximální a cílový API level [1].

Minimální API level určuje minimální verzi Androidu, na kterém je možné aplikaci nainstalovat a spustit. Nastavení této verze je povinné [1].

Maximální API level naopak definuje maximální verzi Androidu, na kterém je možné aplikaci nainstalovat a spustit. Nastavení této verze není povinné a nedoporučuje se [1].

Cílená verze API levelu udává, pro kterou verzi Androidu byla aplikace vyvinuta a testována. Není-li hodnota cílené verze nastavena, je rovna ve výchozím stavu minimálnímu API levelu [1].

3.3.3 Výběr API levelu

Aplikace nevyužívá žádné systémové notifikace, nepodporuje virtuální realitu a nevyžaduje přístup k žádným sensorům. Proto je možné aplikaci vyvíjet pro poměrně nízký minimální API level 22. Díky tomu aplikace pokryje velkou množinu uživatelů na platformě Android. Dle statistik z roku 2021 (viz tabulka 3.1) tak bude aplikace spustitelná na více než 95 % zařízeních se systémem Android [2].

Jako cílový API level aplikace jsem zvolil API level 28, jelikož odpovídá verzi systému mobilního zařízení, na kterém jsem aplikaci primárně testoval.

3.4 Shrnutí

V této kapitole jsem uvedl různé platformy, pro které lze aplikaci vyvíjet. Jelikož je aplikace určena pro platformu Android, tak jsem tuto platformu popsal podrobněji.

Aplikace nevyužívá žádné systémové notifikace, virtuální realitu a podobně a proto nevyžaduje příliš vysoký API level. Díky tomu bylo možné vyvíjet aplikaci na poměrně nízkém minimálním API levelu a pokrýt tak velké množství uživatelů systému Android.

4 Možnosti vývoje aplikace

Pro platformu Android existuje řada možností, jak vyvinout aplikaci. Následující kapitoly popisují, jakými způsoby lze aplikace vyvíjet a jaké nástroje či programovací jazyky k tomu lze využít.

4.1 Přístupy k mobilnímu vývoji

K mobilnímu vývoji lze přistoupit dvěma způsoby: nativním vývojem a multiplatformním vývojem.

4.1.1 Nativní vývoj

Nativní mobilní aplikace jsou většinou psané v konkrétním jazyce pro danou platformu, jako například Objective C nebo Swift pro iOS a Java nebo Kotlin pro Android. Obecně jsou nativní aplikace uživatelsky přívětivé, výkonné a spolehlivé. Jsou vyvíjené na konkrétní platformu a tak umožňují úzkou spolupráci s hardwarem zařízení, jako je práce s Wi-Fi, Bluetooth, GPS a nebo s různými senzory a gyroskopy [28].

4.1.2 Multiplatformní vývoj

Multiplatformní aplikace jsou aplikace, které byly přeloženy pro více platformem z jednoho sdíleného kódu. Princip multiplatformního vývoje stojí na využití největší možné části sdíleného kódu mezi platformami a předchází tak psaní nativního kódu, jelikož ten se musí vytvářet pro každou platformu zvlášť [13].

Pod multiplatformní aplikace spadají také hybridní aplikace, které jsou jakýmsi spojením mezi nativní a webovou aplikací. Jádro aplikace je psané ve webové technologii, převážně v HTML, CSS a JavaScriptu [59]. Jádro je pak obaleno nativní aplikací a zobrazeno ve webovém okně uvnitř aplikace.

Existuje řada frameworků umožňujících multiplatformní vývoj, například React Native a Flutter [53]. Tyto frameworky krátce popíšu v kapitolách 4.2 a 4.3.

4.2 React Native

React Native je open-source framework pro psaní multiplatformních aplikací [37, 53]. Přestože se aplikace v React Native vyvíjí v JavaScriptu, který je typický pro webové aplikace, nejedná se o hybridní vývoj. Pro vykreslování totiž využívá opravdové nativní komponenty, nikoliv webové okno [37].

React Native je vyvíjen společností Meta (dříve Facebook) a první verze vyšla v roce 2015 [53]. Založený je na frameworku React, který slouží pro vývoj webových aplikací [53].

4.3 Flutter

Flutter je poměrně nový, bezplatný open-source framework vyvinutý společností Google s cílem umožnit vývojářům vyvíjet multiplatformní, nativně kompilované aplikace založené na jedné kódové základně a jednom programovacím jazyce. Sestává ze dvou primárních částí: z SDK a UI knihovny [79].

SDK je sada nástrojů určená k vytváření aplikací ve Flutteru. Obsahuje nástroje pro kompilaci zdrojového kódu do nativního kódu pro konkrétní platformu (iOS a Android) [79].

UI knihovna obsahuje sadu komponent pro uživatelská rozhraní (tlačítka, vstupní políčka, atd.), které lze dle vlastní potřeby přizpůsobit. Flutter aplikace jsou psány v jazyce Dart, který vyvinul taktéž Google již v roce 2011 [16, 79].

Vývoj Flutteru byl oznámen v roce 2015 a první oficiální verze vyšla v květnu roku 2017 [15].

4.4 Unity

Unity (známé taky jako Unity3D) je jedním z nejpoblárnějších 2D/3D herních engineů. Další známý engine je například Unreal Engine.

Unity je profesionální herní engine pro tvorbu her cílených na různé druhy platform. Kromě toho je to jeden z nejpřístupnějších moderních nástrojů pro začínající herní vývojáře [64].

Herní engine je systém sloužící pro vytvoření virtuálních počítačových simulací. Obsahuje řadu nástrojů pro vykreslování 2D a 3D grafiky, simulace fyziky, umělé inteligenci, detekce kolizí, práce se zvukem a různými vstupy z klávesnice, myši, joysticků a ostatních zařízení a mnoho dalších funkcionalit [81].

Hojně se využívá pro vytváření her, ale má také své užití v jiných odvětvích, například v automobilovém a filmovém průmyslu, v medicíně a nebo v architektuře. Automobilka Volvo například využila Unity pro simulaci svých designů automobilů, ke zvýšení bezpečnosti nebo dokonce i ke strojovému učení. V seriálu The Mandalorian byl zas využit Unreal Engine pro vytvoření pozadí akčních scén [17].

V červnu 2005 vyšla první oficiální verze enginu [61]. Dnes je Unity již ve verzi 2021.3 LTS. Pro technické nadšence jsou k dispozici i předprodukční verze, aktuálně ve verzi 2022.2 [48, 74].

Pro vývoj aplikace jsem se rozhodl pro herní engine Unity primárně z těchto důvodů:

- Unity je jedním z nejznámějších herních enginů.
- Práce se díky hernímu enginu, který již obsahuje různé funkce pro práci s 3D grafikou zjednoduší.
- Oproti nativnímu vývoji, by bylo pozdější rozhodnutí aplikaci nabízet i na jiných platformách méně časově náročné, než aplikaci vyvíjet znovu nativně na dané platformy.
- S Unity mám již zkušenosti.

Dříve bylo možné aplikace v Unity vyvíjet ve třech různých programovacích jazycích: Boo, UnityScript (modifikovaná verze JavaScriptu) a C#. Počínaje verzí 5 byla podpora jazyka Boo ukončena a podpora UnityScriptu byla ukončena ve verzi 5.7. Nyní je jediným oficiálně podporovaným jazykem C# [43].

4.4.1 C#

C# je moderní, objektově orientovaný jazyk vytvořen společností Microsoft pod vedením Andersna Hejlsberga. Svou syntaxí se podobá Javě a pro programátory se zkušenostmi s jazyky jako C, C++ nebo Java není příliš obtížné se jazyk naučit [11].

4.5 Shrnutí

Mobilní aplikace lze vytvářet různými způsoby. Lze je vytvářet nativně pro konkrétní platformu, hybridně jako webovou aplikaci obalenou nativní knihovnou a nebo multiplatformně.

Já jsem se rozhodl, že budu vyvíjet aplikaci v Unity. Převážně z důvodu podpory 3D grafiky a kvůli tomu, že s Unity mám již zkušenosti.

5 Funkcionality aplikace

Aplikace by měla uživatelům nabídnout interaktivní mapu a užitečné informace ohledně místností a bodů zájmu v univerzitním kampusu ZČU na Borech.

Uživatelé by tak měli mít navíc k již existujícím rozvrhovým aplikacím také aplikaci, která jim usnadní orientaci v areálu kampusu.

5.1 Interaktivní mapa

Jednou z hlavních funkcionalit aplikace je interaktivní mapa, která slouží jako primární orientační nástroj aplikace. Uživateli umožňuje pohled na univerzitní kampus na Borských polích z ptáčích perspektivy. Uživatel má možnost rotovat kolem mapy a po přiblížení k jednotlivým budovám i kolem nich. Budovy jsou také interaktivní - po rozkliknutí budovy se otevře obrazovka se seznamem místností nacházejících se v dané budově.

Mapa slouží pro rychlou orientaci v prostředí kampusu. Každá budova je svázaná s kódem fakulty a budovy, případně s kódy, pokud se v budově nachází více označení, jako tomu je například v budově NTIS (označení UC, UN, US). Tímto systémem lze docílit, že se jednotlivé budovy na mapě barevně kódují dle barvy příslušné fakulty, do které spadají.

5.2 Body zájmů

Nepochybně uživatelé uvítají i značky rozmístěné po celé mapě, ukazující užitečné body zájmů - jako jsou například bufety a kavárny, autobusové a tramvajové zastávky nebo například místa, kde lze vypůjčit koloběžky. Uživatel aplikace má možnost si dle vlastního uvážení tyto značky aktivovat a deaktivovat podle toho, o která místa má zájem. Dále značky může i kompletně skrýt.

5.3 Vyhledávání místností

Součástí mapy je také vyhledávač, který umožňuje uživateli najít místnost dle zadaného vstupu. Vstupem může být libovolný textový řetězec. Typický kód místnosti, např. UC326. Aplikace následně vyhledá všechny místnosti, u kterých kód místnosti začíná, obsahuje či končí na hledaný výraz.

5.4 Informace o místnostech

Další důležitou součástí aplikace jsou informace, které aplikace jednoduchou a přehlednou formou podává uživateli. Tyto informace lze získat buď využitím vyhledávače nebo po rozkliknutí některé z budov na mapě a následného vybrání místnosti ze seznamu místností.

Informace obsahují budovu, ve které se místnost nachází, patro, katedru a typ místnosti. Tam, kde informační systém univerzity poskytuje potřebná data, obsahují informace i kapacitu, rozvrh a nebo zkouškové termíny.

Zatímco mapa se zaměřuje pouze na borský kampus v Plzni, získat informace lze i o ostatních plzeňských místnostech mimo borský kampus. Informace o budovách univerzity v Chebu aplikace zatím neposkytuje.

5.5 Shrnutí

Aplikace obsahuje všechny potřebné funkcionality, aby studentům pomohla nalézt místnosti a body zájmů, které studenty zajímají. Informace najde student jednoduše a přehledně buď v interaktivní mapě, která je primární obrazovkou aplikace a nebo v rámci informačních obrazovek pro budovy a místnosti.

6 Modelování kampusu

Model kampusu je grafický 3D model vytvořený v počítačovém modelovacím nástroji. Na vytvoření modelu jsem se podílel společně se Stepanem Mocjakem.

6.1 3D Modelování

3D modelování je proces tvorby 3D vizualizace předmětu, budovy, živočicha a dalších objektů tak, že manipulujeme s vrcholy, hranami a stěnami modelu v simulovaném 3D prostředí. Tento proces se využívá v široké škále oborů, jako například ve strojírenství, v architektuře, v automobilovém průmyslu nebo ve filmovém a herním průmyslu.

6.2 Nástroje pro tvorbu 3D modelu

K vytvoření 3D modelu lze využít řadu různých aplikací k tomu určených. V následujících kapitolách uvedu několik mně známých nástrojů.

6.2.1 Sketchup

Sketchup je jednoduchý a intuitivní nástroj pro tvorbu 2D a 3D modelů vytvořený v roce 1999 společností @Last Software. V roce 2006 byl odkoupen společností Google a přejmenován na Google Sketchup. Dnes, poté co byla aplikace opět odkoupena od Googlu společností Trimble Inc. (dříve Trimble Navigation), je aplikace opět známá pod původním názvem Sketchup [58].

Sketchup je nabízen ve třech verzích: Sketchup Make, Sketchup Pro a Sketchup Free [58].

Sketchup Make je freeware, jehož vývoj byl pozastaven v roce 2017. Stále je ale k dispozici ke stažení po vytvoření bezplatného účtu na webové stránce Sketchupu [58].

Sketchup Pro je placená verze, která obsahuje všechny funkcionality nástroje. Například import a export v různých formátech [58].

Sketchup Free je nástupce programu Sketchup Make. Jedná se o webovou aplikaci umožňující jednoduchou editaci 3D modelů s omezenými funkcemi. Pro přístup k aplikaci je vyžadován bezplatný účet [58].

6.2.2 3ds Max

3ds Max je profesionální 3D modelovací a vykreslovací software pro tvorbu designů, animací a 3D modelů [66]. Je to jeden z nejznámějších programů ve svém průmyslu. Často je používán herními vývojáři pro tvorbu 3D modelů do videoher. Své užití našel také v reklamním a filmovém průmyslu [78].

Vytvořen byl společností Yost Group pod tehdejším názvem 3D Studio a poprvé vydán v roce 1996 společností Autodesk [66].

6.2.3 Blender

Blender je dalším nástrojem pro tvorbu 3D modelů. Je zcela zdarma a open-source, zdrojový kód je tedy volně dostupný. Software udržuje nezisková organizace Blender Foundation a její komunita [63].

Aktuální verze Blenderu je verze 3.2 [8].

Modelování modelu kampusu v aplikaci Blender jsem zvolil převážně kvůli svým dosavadním zkušenostem s tímto nástrojem, rozsáhlé komunitě a protože je nástroj bezplatný.

6.3 Proces tvorby

Modelování kampusu probíhalo v open-source modelovacím programu Blender. Mapu plzeňského kampusu na Borech jsme vytvořili společně se Stepanem Mocjakem.

Celý kampus je sestaven jako jeden soubor složený z podobjektů reprezentujících různé části kampusu - budovy, cesty, tramvajové koleje a podobně.

Pro vytvoření kampusu jsme se rozhodli využít snímky z Open Street Maps (dále OSM), které jsme použili jako jakousi předlohu, podle které jsme mohli kampus vymodelovat nejprve na dvourozměrné ploše.

Následně jsme tyto budovy vyzdvihnuli do třetího rozměru z 3D náhledů ze satelitních snímků Google Maps. Snažili jsme se oddělit všechny budovy, fakulty a cestičky od sebe. Díky tomu lze každému objektu nastavit zvlášť materiál. Materiály v sobě mimo jiné obsahují barvu a můžeme tak jednotlivé fakulty vykreslovat v jejich barvách.

Celkem model sestává z 29 objektů - 1 objekt pro cesty a tramvajové koleje, 5 objektů pro parkoviště a 23 objektů pro části nebo celky budov.

Kolega se staral o modelování budov a já jsem se staral o modelování cest, tramvajových kolejích a parkovišť. Díky tomu, že jsou modely budov a cestiček od sebe oddělené, jsem byl schopný každému objektu přiřadit

vlastní materiál. Důvodem bylo, abych pak v Unity měl ke každému objektu přiřazen vlastní materiál a mohl tak objekty barevně kódovat například dle barvy fakulty.

6.4 Shrnutí

V této kapitole jsem vysvětlil pojem 3D modelování, vyjmenoval několik možností jak 3D modely vytvářet a popsal jak jsem společně se Stepanem Mocjakem vytvářel model kampusu na ZČU.

7 Uchování dat

Aby bylo možné aplikaci využívat i bez nutnosti internetového připojení, je zapotřebí nějakým způsobem uchovávat data uvnitř aplikace nebo na zařízení uživatele.

V následujících kapitolách se budu věnovat různým přístupům k ukládání dat na platformě Android a jejich výhodami a nevýhodami.

Následně popíšu standardní jazyk pro práci s relačními databázemi SQL a přiblížím alternativní přístup k datům, než pouze vytvářením SQL příkazů.

7.1 Ukládání do preferencí

Pro ukládání primitivních datových typů jako páry klíč-hodnota lze využít systém preferencí [62]. Jedná se o XML soubor uložený pod touto cestou [44]:

```
/data/data/<název balíčku>/shared_prefs/<název  
→ balíčku>.v2.playerprefs.xml
```

Standardně Android umožňuje ukládat více souborů a měnit jejich názvy. Ovšem to v Unity není možné.

Tento způsob se hodí především pro ukládání uživatelských preferencí a nastavení [62].

7.2 Ukládání do souborů

Další možností je ukládání dat do souborů. Může se jednat jak o textové tak i binární soubory [62]. Často se pro takové účely používají XML nebo JSON soubory. Se soubory se potom pracuje standardně přes funkce pro práci se soubory [62].

7.2.1 XML

XML je značkovací jazyk založený na SGML, který je určený pro popis dat [76]. Podobá se jazyku HTML s rozdílem, že nemá předdefinované tagy. Tyto tagy si může uživatel nadefinovat sám, aby tak mohl popsat strukturu dat, která chce uchovávat [49].

7.2.2 JSON

Obdobně jako XML, je i JSON textový formát pro uchování a přenášení dat [22]. Jeho výhodou je jednoduchost a čitelnost. Soubory psané v JSON bývají ve většině případů menší než souboru v XML a přenosy jsou tedy rychlejší [57].

7.3 Databáze

Pro velké množství dat je vhodné použít systém určený k ukládání takových dat. V tomto případě se nabízí využití databáze.

Databázi můžeme mít jak interní (např SQLite) tak i externí (např MySQL). Výhodou interní databáze je, že je uložena přímo na zařízení a není tak nutné připojení k internetu [62].

Externí databáze mohou uchovávat velké množství dat, ale pro přístup k nim je potřebné připojení k internetu.

7.3.1 MySQL

MySQL je open-source nástroj pro správu relačních databází na bázi klient-server architektury a jedním z nejpopulárnějších nástrojů pro práci s databázemi [27]. Původně byl vyvinutý švédskou společností MySQL AB v roce 1994. V roce 2008 byla firma odkoupena a po dobu dvou let vlastněna americkou společností Sun Microsystems. Společnost Oracle v roce 2010 odkoupila firmu Sun Microsystems a od té doby je MySQL ve vlastnictví Oracle [52].

7.3.2 PostgreSQL

PostgreSQL je open-source objektově-relační databázový systém, známý především pro svou spolehlivost a rychlost [34]. Obsahuje mnoho pokročilých funkcí a datových typů [80].

PostgreSQL, dříve známý pod názvem Postgres, byl vyvinutý na Kalifornské univerzitě v Berkeley profesorem Michaelem Stonebrakerem a jeho týmem [35]. Dnes má za sebou již 30 let aktivního vývoje a jeho popularita stále roste. Aktuálně je ve verzi 14 [34].

7.3.3 SQLite

Na rozdíl od předešlých relačních databází není SQLite založen na klient-server architektuře. Celý program je obsažen v knihovně psané v jazyce C a lze jej tak jednoduše implementovat do vlastního softwaru [85]. Díky

tomu není vyžadované připojení k internetu. Celý systém pro své fungování vyžaduje pouze jediný soubor, ve kterém jsou všechna data uložena [41].

Právě kvůli své jednoduché implementaci a přístupnosti se stal SQLite tak populárním [85].

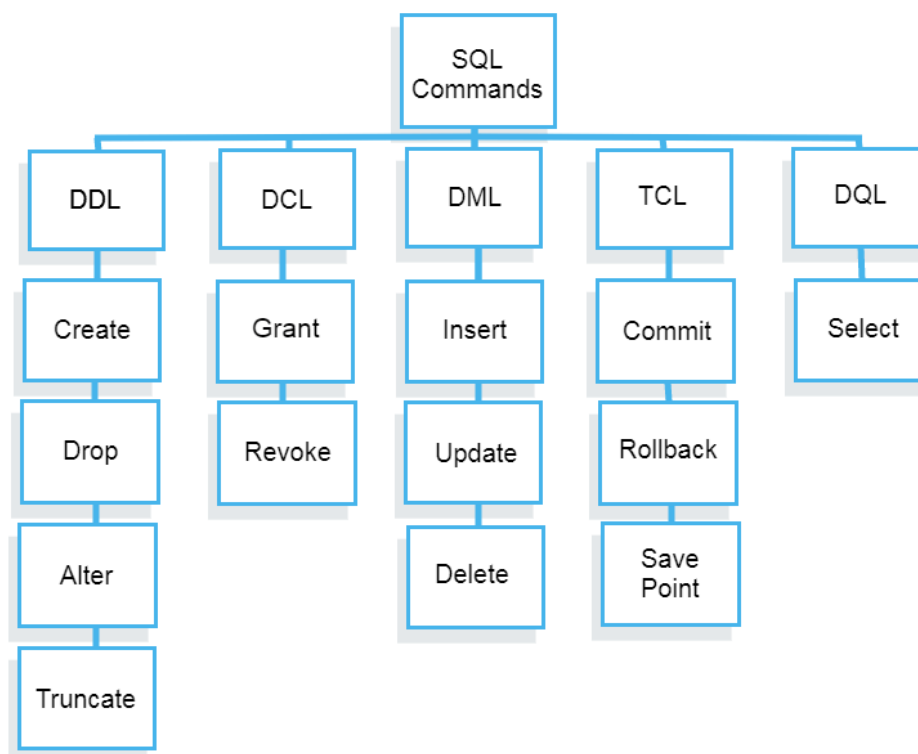
7.4 SQL

SQL je standardizovaný jazyk určený pro manipulaci s daty v relačních databázových systémech [75]. Všechny databázové systémy, které jsem popsal v předešlých třech kapitolách, využívají SQL pro práci s daty a liší se pouze některými menšími odlišnostmi související s konkrétním databázovým systémem.

SQL funguje na principu dotazů složených z příkazů, které se dají řetězit za sebou [75].

SQL příkazy jsou rozděleny do několika typů: Data Definition Language (DDL), Data Manipulation Language (DML), Data Query Language (DQL), Data Control Language (DCL) a Transaction Control Language (TCL) (viz obr. 7.1) [77].

DDL slouží pro práci se strukturou databáze a databázových objektů, pomocí DCL lze řídit práva a oprávnění, s DML lze manipulovat s daty a s DQL lze data číst. TCL se stará o transakce a jejich případné vrácení.



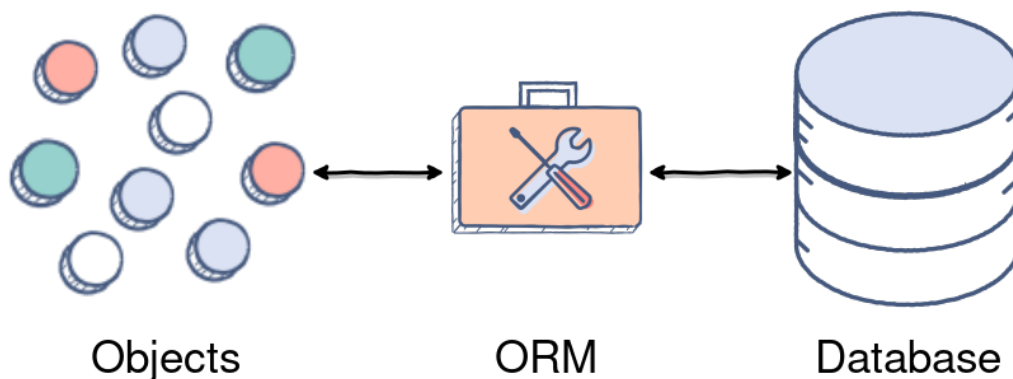
Obrázek 7.1: Rozdělení SQL příkazů [77]

Zdroj: <https://www.guru99.com/sql-commands-dbms-query.html>

7.5 ORM

Pro zjednodušení práce s databází využívám v projektu objektově relační mapování.

Objektově relační mapování poskytuje vrstvu mezi relačními databázemi a aplikací, která umožňuje získávat a manipulovat s daty v databázi objektově orientovaným přístupem [33, 70]. Můžeme si ORM představit jako most propojující databázi s naším softwarem (viz obr. 7.2) [30].



Obrázek 7.2: Schéma ORM [30]

Zdroj: <https://www.educative.io/answers/what-is-object-relational-mapping>

7.5.1 NHibernate

NHibernate je aktivně vyvíjená open-source knihovna pro objektově relační mapování pro jazyky z .NET rodiny [31, 71]. Je inspirována populární Java knihovnou Hibernate [31].

NHibernate podporuje databázové systémy Microsoft SQL, Windows Azure SQL, Oracle, MySQL a PostgreSQL [71]. Aktuálně je již ve verzi 5.x [32]. Pro podporu SQLite bylo nutné implementovat vlastní driver [29].

7.5.2 Fluent NHibernate

Fluent NHibernate je nadstavba nad knihovnou NHibernate, která umožňuje ukládání databázového modelu do jednotlivých tříd místo do XML souborů, jak to dělá NHibernate [14]. Dále obohacuje knihovnu o fluent interface, což je princip řetězení volání metod jedné instance třídy. Tím lze snížit duplicitu kódu a zároveň zvýšit jeho čitelnost [83].

7.6 Datové modely

Aplikace obsahuje několik datových modelů, které se vytváří buď při prvotním spuštění aplikace, nebo průběžně za běhu aplikace.

Databázový model popisuje strukturu prvku nějaké konkrétní databázové tabulky.

V následujících podkapitolách se vyjádřím k těm nejdůležitějším a nejobsáhlejšími datovým modelům. Zbytek databáze je popsán v databázovém diagramu viz obr. 7.3.

7.6.1 Room

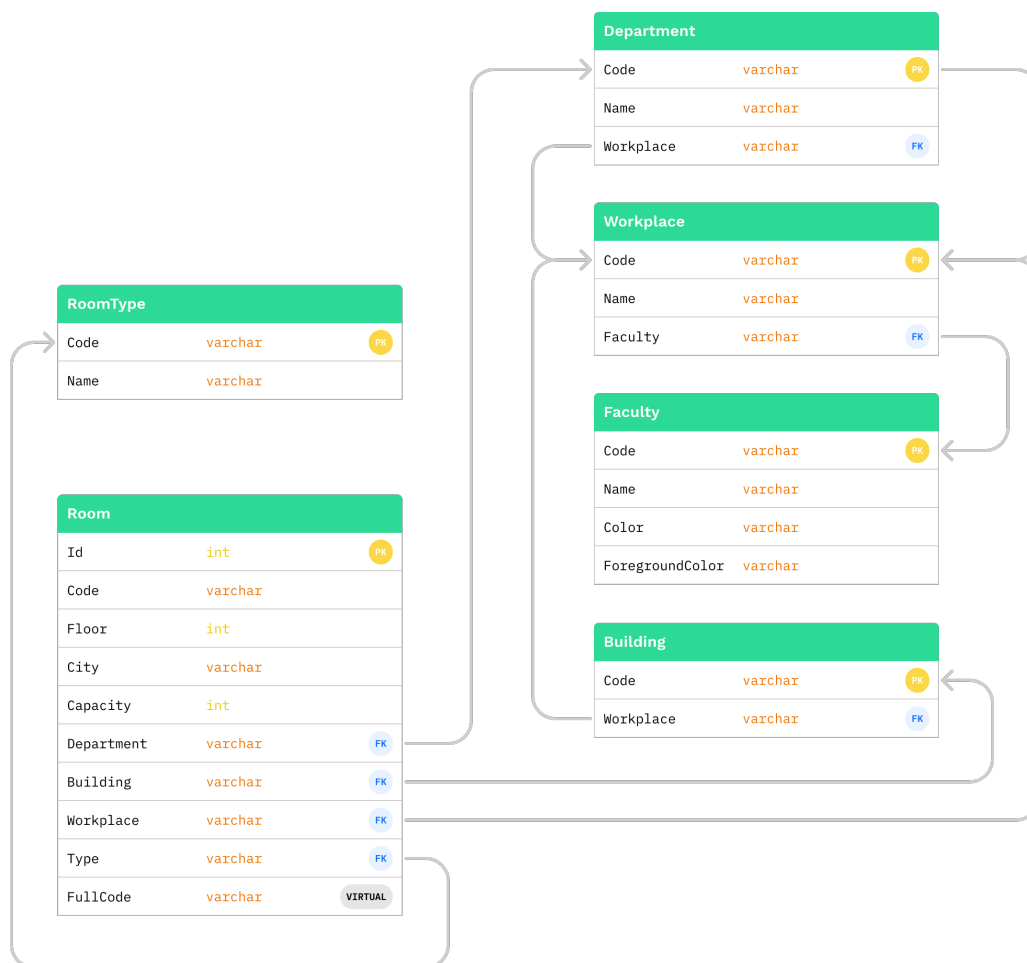
Hlavní datový model. Obsahuje informace o místnostech - kód, budovu, patro a podobně. S tímto modelem také souvisí model *RoomType*, který obsahuje kód a příslušný název typu místnosti. Například posluchárna, kancelář, laboratoř, atd.

7.6.2 Faculty

Obsahuje informace o fakultě. Kromě názvu a kódu obsahuje také barvu fakulty pro obarvení budov na mapě a v uživatelském rozhraní (tlačítka, značky, apod.). V neposlední řadě obsahuje barvu pro texty pro případ, kdy se vykreslí texty přes grafický prvek s barvou fakulty.

7.6.3 Ostatní modely

- Department
- Building
- Workplace



Obrázek 7.3: diagram databáze

7.7 Migrace

Implementoval jsem jednoduchý migrační systém, který slouží pro práci s databází. Jedná se o třídy ze jmenného prostoru *Campus.Database.Migrations*, které dědí abstraktní třídu *Migration* a implementují jedinou funkci *Run()* (viz ukázka kódu 7.1).

Migrace mají několik úkolů. Prvním úkolem migrace je separovat databázový kód od logiky aplikace pro zlepšení čitelnosti kódu.

Dalším úkolem migrace je plnění databáze daty. Například pro vytvoření prvků datového modelu *Building* slouží migrace *BuildingMigration*.

V neposlední řadě umožňují migrace aktualizovat data aplikace. Bude-li nutné v budoucí verzi aplikace přidat například další místnosti, vytvoří se pro tento účel nová migrace.

Aplikace si pamatuje, které migrace byly již provedeny a při každém

spuštění prohledá jmenný prostor, seřadí všechny migrace podle priority a následně spustí ty migrace, které ještě nebyly provedeny.

Priorita je statická vlastnost třídy *Migration*, kterou lze v dědící třídě přepsat. Priorita určuje pořadí migrace v případě, pokud je nutné, aby byla migrace spuštěna dříve než jiná (například jsou-li na sobě závislé). Ve výchozím nastavení je nastavena na 0. Čím **vyšší číslo je nastaveno, tím vyšší prioritu migrace má** a tím dříve bude provedena

Práce s objekty při běhu aplikace je možná díky reflexi (viz kapitola 7.7.1).

```
1 namespace Campus.Database {
2     public abstract class Migration {
3
4         public static int Priority => 0;
5         public virtual bool AutoTidyUp => true;
6
7         public static bool Done{ get; set; }
8
9         public ISession Session { get; private set; }
10
11        public ITransaction Transaction { get; private set; }
12
13        public Migration(ISession session) { ... }
14
15        public void TryRun() { ... }
16
17        private bool ShouldRun() { ... }
18
19        protected abstract void Run();
20
21        protected void TidyUp() { ... }
22
23        public static bool IsDone() { ... }
24    }
25 }
```

Ukázka kódu 7.1: třída Migrations.cs

Díky tomu, že se spustí migrace vždy jen jednou, nedojde k vytvoření duplicitních záznamů v databázi a umožňuje flexibilně upravovat databázovou strukturu i v případě aktualizace aplikace.

7.7.1 Reflexe

Reflexe poskytuje informace o typech objektů při běhu aplikaci. Je to komplexní a silný nástroj, díky kterému lze dynamicky vytvářet instance typu, získávat informace o typu z již existujícího objektu nebo dokonce i zavádět nové metody a volat již existující metody [12, 50, 64].

7.8 Shrnutí

V této kapitole jsem popsal jaké různé metody jsou k dispozici pro uchování dat na platformě Android. Zmínil jsem, které z metod využívám v aplikaci já a za jakým účelem.

Popsal jsem co je to ORM, proč jej využívám, jaké datové modely aplikace obsahuje a jak funguje mnou vytvořená migrační utilita.

8 Webové služby IS-STAG

Jako zdroj dat o budovách a místnostech na ZČU využívám univerzitní informační systém STAG. Všechny koncové body, které jsou v aplikaci využity jsou veřejně dostupné a tak nevyžadují žádnou autentizaci.

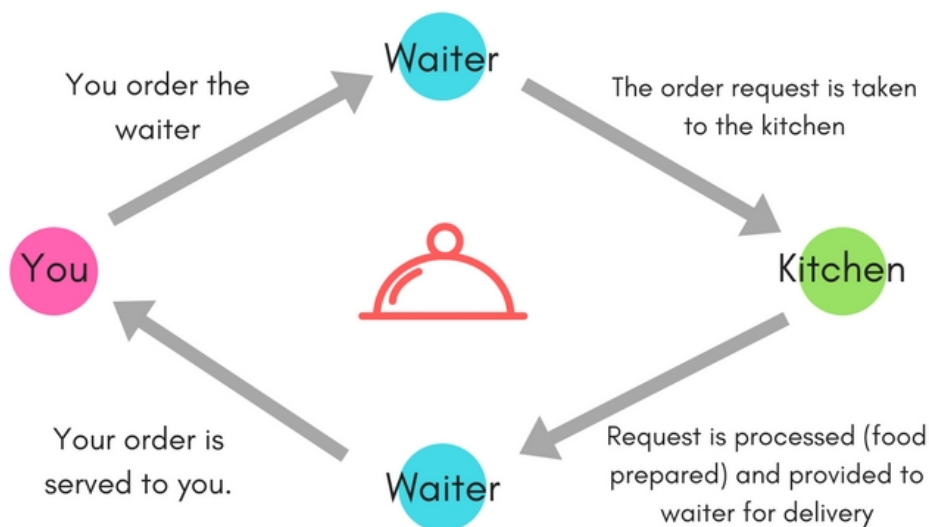
K webovým službám STAGu lze přistupovat dvěma způsoby: přes SOAP API a nebo přes REST API.

8.1 API

API je prostředník, který umožňuje softwaru komunikovat s jiným softwarem. Typické užití je například u aplikací typu klient-server, kdy klientem může být např. mobilní aplikace pro zobrazování aktuálního počasí a serverem nějaký webový systém, který poskytuje informace o počasí klientovi. Klient neví přesně jak data webový systém zpracovává, ale ani ho to nezajímá. Klienta zajímá jen výsledek, který od webového systému očekává [3].

Můžeme si princip fungování API ukázat i na analogickém příkladu z reálného světa (viz obr. 8.1):

Máme restauraci, ve které si host objednává své oblíbené jídlo. Číšníkovi sdělí, které jídlo by si přál a číšník požadavek předá do kuchyně, kde se jej kuchař ujme. Jakmile bude mít uvařeno, číšník jídlo převezme a předá hostovi. Host je v tomto případě klient, číšník představuje API a kuchař funguje jako server, který požadavek zpracovává. Hosta nezajímá jak přesně kuchař jídlo připravuje, zajímá ho jen výsledek [4].



Obrázek 8.1: Analogie API z reálného světa [4]

Zdroj: <https://www.studytonight.com/post/understanding-what-api-is-and-how-it-works>

8.2 SOAP

SOAP je komunikační protokol navržený v roce 1998. Dnes je hojně využíván v rámci webových služeb ale jeho užití není omezeno jen pro web. Jediným podporovaným formátem je XML [40].

SOAP je procedurálně orientovaný [60] (spouští vzdálené procedury) a byl navržen tak, aby byl nezávislý na platformě a operačním systému. Je založený na posílání zpráv. Zprávou je XML dokument, který se skládá z obálky, hlavičky a těla [68, 84].

Obálka je obalující kořenová značka označující XML dokument jako SOAP zprávu [68].

Hlavička je volitelná a obsahuje doplňující informace ke zprávě. Například identifikaci uživatele či autentizační informace [68].

Tělo je nejdůležitější částí zprávy, které obsahuje předávaná, resp. návratová data [68].

8.3 REST

REST je návrhový vzor pro webové API využívající HTTP/HTTPS protokol. Byl vyvinutý v roce 2000 Royem Fieldingem. Aplikacím, které návrhový vzor REST využívají, se říká RESTful aplikace [60].

REST je na rozdíl od SOAPu datově orientovaný. Pracuje tedy pouze s daty, která se nazývají zdroje [39]. Primárně pro přenos dat kvůli jednoduchosti a čitelnosti využívá formát JSON, ale podporuje i další formáty, mimo jiné také XML [38].

8.4 Postman

Před implementací webových služeb do aplikace je vhodné nejprve koncové body webových služeb otestovat. K tomu jsem využil software Postman (<https://www.postman.com>).

Postman je vývojářský nástroj pro vytváření, testování a modifikování API. Nástroj umožňuje vytváření různých HTTP (GET, POST, PUT, PATCH, DELETE a mnoho dalších) požadavků na různé koncové body API. Lze jednoduše přidat parametry a celý požadavek lze rovnou z nástroje odeslat a zobrazit odpověď v různých formátech jako jsou například JSON a XML. V programu lze také vytvářet proměnné a skripty, které se spouští před jednotlivými požadavky a lze tak docílit komplexních testovacích scénářů [36].

V Postmanu jsem si pro vývoj vytvořil kolekci, která slouží k seskupení souvisejících požadavků týkajících se jednoho projektu či API. Do kolekce jsem umístil všechny potřebné požadavky ke koncovým bodům (více ke koncovým bodům v následující kapitole). Kromě požadavků jsem ke kolekci také přidal proměnnou *URL*, která obsahuje adresu webové služby.

V poslední řadě jsem napsal jednoduchý skript (viz ukázka kódu 8.1), který před každým spuštěním požadavku upraví URL a hlavičku požadavku.

```
1 pm.request.url = pm.collectionVariables.get('URL') + pm.request.url;  
2 pm.request.headers.add({key: 'Accept', value: 'application/json'});
```

Ukázka kódu 8.1: Skript upravující URL a hlavičku požadavku v Postmanu

Tímto jsem docílil toho, že každý existující i budoucí požadavek bude automaticky využívat nastavenou adresu z proměnné *URL* a vždy se do hlavičky přidá parametr, který určí jaký formát odpovědi klient očekává. V případě, že by se v budoucnu změnila URL webových služeb STAG, bude stačit změnit jen proměnnou na jednom místě.

8.5 Koncové body

Všechny koncové body, které jsem v aplikaci použil, krátce popíšu v následujících podkapitolách.

Každý koncový bod je popsán tabulkou s typem HTTP požadavku, názvem/URL koncového bodu a seznamem parametrů. **Tučně** vyznačené parametry jsou povinné.

Popsány jsou pouze koncové body a parametry, které jsem pro vývoj využil. Úplný popis specifikace webových služeb STAGu je k dispozici na adrese <https://stag-ws.zcu.cz>.

8.5.1 Získání místností

get	/ws/services/rest2/mistnost/getMistnostiInfo
Parameter	
<i>Bez parametrů</i>	

Tento koncový bod slouží k získání seznamu místností na univerzitě. V závislosti na zadaných parametrech umí vyfiltrovat místnosti z jednotlivých budov na univerzitě. Aplikace ovšem využívá základní požadavek bez parametrů pro získání všech místností.

Aby bylo možné aplikaci používat i bez internetu, jsou již data stažená a přibalena v aplikaci v interní SQLite databázi. Aktuální data lze stáhnout v nastavení aplikace.

8.5.2 Získání seznamu domén

get	/ws/services/rest2/ciselniky/getSeznamDomen
Parameter	
<i>Bez parametrů</i>	

Tento koncový bod není přímo využit uvnitř aplikace, ale je využit jen jako pomůcka v Postmanu pro zjištění seznamu číselníků, které jsou k dispozici pro použití jako parametr pro následující koncový bod.

8.5.3 Získání číselníků

get	<code>/ws/services/rest2/ciselniky/getCiselnik</code>
Parameter	
domena	název domény číselníku

Tímto koncovým bodem lze získat obsahy různých číselníků. Pro získání obsahu je nutné znát název domény z předchozího koncového bodu.

V aplikaci jsem využil následující domény:

- TYP_UCEB: Obsahuje informace o typech učeben.
- PRACOVISTE: Obsahuje informace o pracovištích na univerzitě.
- FAKULTA: Obsahuje informace o fakultách na univerzitě.
- BUDOVA: Obsahuje informace o jednotlivých budovách univerzity.

8.5.4 Získání rozvrhů místností

get	<code>/ws/services/rest2/rozvrhy/getRozvrhByMistnost</code>
Parameter	
budova	Kód budovy, například UC
mistnost	Kód/číslo místnosti, například 326

Tímto koncovým bodem získávám v aplikaci rozvrh jednotlivých místností. Jedná se o jediný endpoint, který se volá opakovaně při běhu aplikace a jehož výsledek není nijak uchovávan pro offline použití.

Je to dáno tím, že pokud bych zabalil i rozvrhy do aplikace, byla by aplikace zbytečně velká a data by stejně brzy byla neplatná, protože se rozvrhy mění každý semestr.

8.6 Mapování na databázový model

Databázové modely byly navrženy tak, aby bylo možné výstupy z webových služeb STAGu přetransformovat na interní databázové modely .

K tomuto účelu využívám knihovnu JSON .NET pro Unity [73], která umožňuje jednoduchým způsobem přes atributy vlastností mapovat JSON objekty na instanci třídy a opačně (viz ukázka kódu 8.2).

```

1 namespace Campus.Models {
2     public class Lector {
3
4         ...
5
6         [JsonProperty("ucitIdno")]
7         public virtual string Id { get; set; }
8
9         ...
10    }
11 }
12 }

```

Ukázka kódu 8.2: Datový model Lector

Máme-li JSON objekt s daty o lektorovi, můžeme jej tak díky atributům (*JsonProperty*) deserializovat pomocí:

```
var lector = JsonConvert.DeserializeObject<Lector>(jsonString);
```

a opět zpět serializovat do JSON objektu pomocí:

```
var jsonString = JsonConvert.SerializeObject(lector);
```

8.7 Shrnutí

Cílem této kapitoly bylo prozkoumat co vše webové služby univerzity nabízí, jaké výhody a nevýhody různé přístupy k API mají a jak koncové body testovat bez nutnosti psaní kódu.

Kritériem, který jsem si stanovil bylo, aby se se získanými daty dobře pracovalo a aby se data jednoduchým způsobem dala přetransformovat na datové modely databáze.

Po uvážení těchto požadavků a mých dosavadních zkušeností s uvedenými technologiemi jsem se rozhodl využít REST API webových služeb a data získávat ve formátu JSON.

9 Struktura a řešení aplikace

V této kapitole je popsáno řešení aplikace a jakým způsobem je projekt a kód strukturován. Následně jsou popsány některé důležité třídy aplikace.

9.1 Struktura projektu

Aplikace je strukturována do několika složek, které pomáhají organizaci projektu. Část složek byla vytvořena automaticky engine Unity pro ukládání různých nastavení, instalovaných balíčků z Unity Asset Storu a podobně. Zbylé složky jsem vytvořil já.

Aby aplikace mohla uživateli nabídnout jednoduché, přehledné a přívětivé uživatelské rozhraní, je potřeba řada grafických prvků jako jsou obrázky, ikony, 3D modely a podobně. Tyto prvky jsou rozděleny pro přehlednost do různých složek, například modely jsou obsaženy ve složce **Models**, obrázky a ikony ve složce **Sprites** a podobně.

Důležitou složkou je složka **UI**, která obsahuje všechny konfigurační soubory a styly pro vykreslení uživatelského rozhraní.

Další velmi důležitou složkou je složka **Scripts**, která obsahuje veškerý kód aplikace.

9.2 Struktura kódu

Kód aplikace se skládá z mnoha tříd, které spolu různě interagují. Třídy jsou z organizačních důvodů rozděleny do několika jmenných prostorů dle funkce, kterou vykonávají.

Jmenný prostor **API** obsahuje třídy, které se starají o komunikaci s API webových služeb STAGu.

Camera obsahuje třídy starající se o funkcionalitu kamery.

Jmenný prostor **Database** obsahuje třídy týkající se práce s databází. Jedná se o řídicí třídy pro práci s databází a migrace, které se nacházejí ve vnořeném jmenném prostoru **Migrations**.

Entities obsahuje třídy, které reprezentují nějaký objekt na interaktivní mapě. Například budova, fakulta a nebo značka pro bod zájmu.

Enums je jmenný prostor obsahující veškeré číselníky aplikace.

Models obsahuje datové modely aplikace.

Jmenný prostor **UI** slouží pro třídy, které se starají o uživatelské rozhraní.

A poslední jmenný prostor **Utils** obsahuje pomocné třídy aplikace.

9.2.1 Třída App

Hlavní třídou aplikace je třída App (viz ukázka kódu 9.1). Tato třída uchovává informace o stavu aplikace, například zda se uživatel nachází na mapě či v UI, zda je aplikace připojena k internetu a nebo například v jakém stavu se nachází kamera.

```
1 namespace Campus
2 {
3     public class App : MonoBehaviour
4     {
5         public static App Instance{ get; private set; }
6
7         public State State;
8         public MouseState MouseState = MouseState.NONE;
9         public CameraState CameraState = CameraState.IDLE;
10
11        public Dictionary<string, Building> Buildings{ get; } = new();
12        public Dictionary<string, VectorImage> Icons{ get; } = new();
13        public Icon[] icons;
14
15        private NetworkReachability NetworkReachability;
16
17        public delegate void ConnectivityHandler(NetworkReachability
18        ↪ networkReachability);
19        public event ConnectivityHandler OnConnectivityChange;
20
21        private void Awake() { ... }
22
23        private void Update() { ... }
24
25        private void LoadBuildings() { ... }
26
27        private void LoadIcons() { ... }
28    }
29 }
```

Ukázka kódu 9.1: Třída App.cs

9.2.2 Třída CameraController

Další důležitou třídou je třída *CameraController* (viz ukázka kódu 9.2). Tato třída se stará o rotaci kamery kolem definované pozice na mapě. Ve výchozím stavu se jedná o střed mapy. Pozici lze také nastavit například na budovu a kamera se poté bude rotovat kolem této budovy.

```
1 namespace Campus.Camera
2 {
3
4     public class CameraController : MonoBehaviour {
5
6         ...
7
8         private void Update() {
9             if (App.Instance.CameraState == CameraState.IDLE &&
10                ⇨ App.Instance.MouseState != MouseState.DRAG) {
11                 // store mouse
12                 mouseReference = Input.mousePosition;
13             }
14
15             if (App.Instance.CameraState != CameraState.IDLE ||
16                ⇨ App.Instance.MouseState != MouseState.DRAG) return;
17
18             // offset
19             mouseOffset = (Input.mousePosition - mouseReference);
20
21             // apply rotation
22             var angle = (mouseOffset.x + mouseOffset.y) * sensitivity;
23
24             // rotate
25             transform.RotateAround(point, Vector3.up, angle *
26                ⇨ Time.deltaTime * speedMod);
27
28             // store mouse
29             mouseReference = Input.mousePosition;
30         }
31     }
32 }
```

Ukázka kódu 9.2: Třída CameraController.cs

9.2.3 API Manager

API manager je velmi jednoduchá, avšak poměrně významná třída, která se stará o získávání dat ze STAGu (viz ukázka kódu 9.3). Obsahuje jedinou metodu *Send*, která se stará o odesílání požadavku webové službě STAG.

Díky tomu, že metoda vrací jako návratovou hodnotu instanci své třídy je možné volání metody *Send* několikrát řetězit.

```
1 namespace Campus.API {
2
3     public class ApiManager : MonoBehaviour {
4         public bool flushPlayerPrefs = false;
5         protected internal static string URI =>
6             ↪ "https://stag-ws.zcu.cz";
7
8         public static ApiManager Instance{ get; private set; }
9
10        private void Awake() { ...}
11
12        public ApiManager Send(Request request) {
13            StartCoroutine(request.Send());
14
15            return this;
16        }
17    }
18 }
```

Ukázka kódu 9.3: Třída ApiManager.cs

Se třídou *API manager* souvisí také třída *Request* (viz ukázka kódu 9.4), která definuje požadavek pro API webových služeb STAGu.

```

1 namespace Campus.API {
2
3     public class Request {
4         private static string URI => "https://stag-ws.zcu.cz";
5
6         public string Endpoint{ get; private set; }
7         public Action<UnityWebRequest.Result, string, Action>
8         ↪ Callback{ get; set; }
9
10        protected UnityWebRequest R{ get; private set; }
11
12        protected Action End{ get; set; }
13        protected Request(string endpoint,
14        ↪ Action<UnityWebRequest.Result, string, Action>
15        ↪ callback, Action end = null) { ... }
16
17        public Request SetHeader(string key, string value) { ...
18        ↪ }
19
20        internal IEnumerator Send() {
21            yield return R.SendWebRequest();
22
23            if (R.result != UnityWebRequest.Result.Success)
24            ↪ {
25                Callback.Invoke(R.result, String.Empty,
26                ↪ End);
27            }
28            else {
29                Callback.Invoke(R.result,
30                ↪ R.downloadHandler.text, End);
31            }
32        }
33
34        public static Request Create(string endpoint,
35        ↪ Action<UnityWebRequest.Result, string, Action>
36        ↪ callback, Action end = null) {
37            return new Request(endpoint, callback, end);
38        }
39    }
40 }

```

Ukázka kódu 9.4: Třída Request.cs

9.3 Řešení problémů

Potíže při vývoji a hledání jejich řešení je nevyhnutelnou součástí vývoje každého softwaru. V této kapitole bych chtěl popsat několik řešených problémů, které mi přišly zvláště zajímavé nebo neobvyklé.

9.3.1 Potíže s otevřenou klávesnicí

Když je otevřená klávesnice, aplikace nezaregistrovává dotykový vstup na obrazovce.

Toto je bohužel problém, který v Unity existuje již roky a oficiálně zatím není známé řešení. Z diskuse, datující až do roku 2015 vyplývá, že do dnes neexistuje oficiální řešení problému a jediným řešením je využití komunitních pluginů či kódu [42]. Tato řešení jsou však využitelné jen pro staré nástroje pro práci s uživatelským rozhraním. Já využívám nový aktuální systém (UI Toolkit) pro který jsem zatím žádné řešení nenašel.

Problém však nezpůsobuje žádné výrazné potíže v aplikaci a ani jinak neomezuje funkcionalitu aplikace.

9.3.2 SQLite

Dalším z řešených problémů bylo zprovoznění SQLite databáze na mobilním zařízení. Unity Engine, jelikož se jedná o herní engine, sám o sobě nemá podporu SQL a tak bylo zapotřebí do aplikace zabalit spoustu knihoven, které funkcionalitu pro práci s SQL databází doplní.

Zajímavé bylo, že přestože Unity nemá podporu pro SQL, tak v sobě nese systémovou knihovnu pro práci s SQL. Díky online návodu [47] se mi podařilo SQLite zprovoznit tak, že jsem z instalačního adresáře Unity editoru zkopíroval do projektu soubory nutné pro fungování SQLite (System.Data, Mono.Data.Sqlite).

Tím se z počátku zdálo, že vše funguje. Ovšem později se ukázalo, že na systému Android práce s databází stále nefunguje. K tomu bylo nutné ještě navíc stáhnout knihovny určené pro konkrétní platformu a přibalit je do aplikace [45, 46, 51].

9.3.3 Stahování aktualizace dat

Stahování aktualizace dat v nastavení aplikace probíhá na starších zařízeních pomalu. Jelikož celá aplikace běží v jednom vlákne, po dobu několika vteřin se aplikace stane neovladatelnou.

Abych případ z uživatelského hlediska co nejlépe ošetřil, přepne se aplikace do stavu aktualizace a vypne funkcionalitu veškerých tlačítek na obrazovce, aby tak uživatel neprovedl nějakou akci, která by se po aktualizaci dat samovolně spustila.

Dále se po spuštění aktualizace vypíše informace o tom, že probíhá aktualizace a že má uživatel počkat, než se aktualizace dokončí.

V budoucí verzi by mělo stahování aktualizace probíhat v pozadí ve vlastním vlákně a uživatel tak mohl aplikaci dále využívat.

9.4 Distribuce

Aplikaci lze distribuovat prostřednictvím APK souboru, který získáme dle návodu v instalační příručce v příloze.

Chtěl bych také zkontaktovat Centrum informatizace a výpočetní techniky ZČU a případně se domluvit na distribuci do Google Play Storu pod univerzitním účtem.

9.5 Shrnutí

Aplikace se skládá z poměrně velkého množství složek a souborů, které společně definují celou aplikaci.

Během vývoje jsem narazil na různé potíže, které jsem zde zmínil a popsal, jakým způsobem jsem problém řešil.

10 Rozšíření

Aplikace je plně funkční a rozsahově odpovídá zadání práce. Existuje ovšem řada dalších funkcionalit nad rámec bakalářské práce, které by aplikace v budoucích verzích mohla obsahovat. Tyto funkcionality stručně popíšu v následujících podkapitolách.

10.1 Podrobnější informace

Aplikace aktuálně uživateli poskytuje pouze ty nejdůležitější informace o místnostech a budovách.

V budoucí verzi by měla uživateli poskytovat doplňující informace k budovám, místnostem a lektorům.

U budov by například mohla zobrazovat adresu a možnost zobrazit 2D půdorys jednotlivých pater. U místností by měla v závislosti na typu místnosti zobrazit dodatečné informace. Například u kanceláří by měla uživateli sdělit, kdo z vyučujících zde sídlí, jaké má konzultační hodiny a jakým způsobem lze vyučujícího kontaktovat.

10.2 Oblíbené

Užitečnou rozšiřující funkcionalitou by byla možnost přidat místnosti do oblíbených. Uživatel by tak měl všechny své často navštívené místnosti na jednom místě a mohl by k nim přistupovat rychleji.

10.3 Přidání vlastních bodů zájmů

Aplikace obsahuje různé body zájmů, které uživatel vidí na mapě. Pokud uživateli nějaký bod zájmu chybí, měl by v budoucí verzi mít možnost, si tento bod vytvořit a umístit na mapě sám.

10.4 Cachování rozvrhů

Po otevření obrazovky s informacemi o místnosti je odeslán požadavek pro stažení aktuálního rozvrhu místnosti z webových služeb univerzity. V budoucí verzi by se měla odpověď po určitou dobu uložit do mezipaměti, aby

bylo možné rozvrh zobrazit i v případě krátkodobého výpadku internetového připojení.

10.5 Půdorysy budov

Aktuálně se po rozkliknutí budovy na interaktivní mapě zobrazí seznam všech místností v budově. V budoucí verzi by měla zobrazit 3D model budovy s možností zobrazit půdorys jednotlivých pater budovy.

V každém patře by měly být zakresleny místnosti, které by měly být také interaktivní a po rozkliknutí zobrazit informace o místnosti.

10.6 Rozšíření mapy

V aktuální verzi se jedná o mapu kampusu ZČU na Borech. V dalších verzích by bylo možné mapu rozšířit i o další budovy, které se nachází mimo kampus na Borech.

10.7 Vnitřní navigace

A v neposlední řadě by se aplikace mohla rozšířit o vnitřní navigaci, která by uživatele zavedla od aktuální pozice až k hledané místnosti. Tato funkcionality by ovšem vyžadovala mnohem více úsilí a spolupráci s univerzitou pro zjištění pozice uživatele uvnitř budov.

Vnitřní navigace se dá realizovat několika způsoby. Jednou z nejlevnějších, ale také z pohledu uživatele nejméně přívětivých, variant by bylo rozmístění QR kódů po celém kampusu a uvnitř budov. Tyto QR kódy by uživatel naskenoval a tím aktualizoval svou pozici v budově. Aplikace by poté uživateli uměla vykreslit nejkratší cestu do požadované místnosti. V případě, že by uživatel naskenoval další QR kód, tak by se cesta od nové pozice znovu přepočítala.

Další, finančně nákladnější variantou na realizaci je rozmístění velkého počtu Bluetooth beaconů ve všech budovách ZČU. Tyto beacony periodicky rozesílají signál, které okolní zařízení s podporou Bluetooth mohou zaznamenat. Změřením síly signálu by následně bylo možné určit polohu uživatele uvnitř budovy. Tímto způsobem by bylo možné realizovat plně funkční navigační systém uvnitř budov.

10.8 Shrnutí

Aplikaci lze rozšířit o spoustu dalších užitečných rozšíření. Kromě rozšíření aktuálních funkcionalit lze přidat řadu zcela nových funkcí, které uživatelům poskytnou další informace a možnosti, jak s aplikací interagovat.

Myslím si, že má aplikace potenciál stát se užitečným pomocníkem mnoha studentů, kteří se zejména na začátku nového semestru na kampusu ztrácejí.

11 Testování

Implementované funkcionality a očekávané chování aplikace je nutné otestovat. V následujících kapitolách popíšu různé testovací scénáře, kterými jsem aplikaci testoval.

11.1 Hledání místnosti

Ústřední funkcionalitou aplikace je vyhledávání místností na univerzitě. Hledání by mělo být možné z kterékoliv obrazovky stisknutím tlačítka pro hledání v navigaci aplikace. Tlačítko se nachází uprostřed navigace v dolní části obrazovky, je označeno ikonou lupy a textem Hledat.

Testoval jsem otevření vyhledávacího dialogu ve všech obrazovkách aplikace.

Ve všech případech došlo k očekávanému chování a to otevřením dialogu v horní části obrazovky, otevřením klávesnice pro zadání vstupu a v případě, kdy se uživatel nachází v nějaké jiné obrazovce než té hlavní (na mapě), tak se obrazovka skryje.

Dále ve všech těchto případech taktéž fungovalo zadávání vstupu od uživatele a otevření místnosti, kterou uživatel zvolil.

Nepříjemná situace ovšem nastává při zvolení místnosti. Pokud má uživatel otevřenou klávesnici a pokusí se zvolit nějakou z místností, očekával by, že se provede nějaká akce. Ovšem jak již bylo zmíněno v kapitole 9.3.1, pouze se zavře klávesnice a až po následujícím kliknutí se provede očekávaná akce. Bohužel se jedná o vlastnost prostředí a aktuálně nelze odstranit.

11.2 Manipulace s body zájmů

Uživatel má mít možnost si zvolit, které body zájmu uživatele zajímají a které si přeje vidět na mapě ve formě značek. Dále má možnost všechny body skrýt a zobrazit.

Standardně jsou zobrazeny všechny body zájmů a jsou viditelné.

Zkusil jsem si některé body odznačit, vyhledat si místnost, zobrazit si informace o místnosti a následně znovu zobrazit mapu. Stále se zobrazovaly na mapě jen ty značky, které jsem měl označené.

V dalším kroku jsem značky skryl, aplikaci vypnul a znovu zapnul s cílem zjistit, zda si aplikace zapamatuje mé preference. Po spuštění aplikace byly

dle očekávání značky skryté a po zobrazení značek byly vidět jen ty značky, které jsem v předešlém kroku označil jako aktivní.

11.3 Otočení obrazovky

Aplikace byla vyvinuta pouze pro režim na výšku. Ve výchozím nastavení Unity je ovšem nastavení konfigurováno na automatický režim. Aplikaci šlo tedy otáčet jak na výšku, tak i na šířku.

Automatický režim otáčení však není žádoucí a tak bylo nutné nastavení v Unity pozměnit.

Při opakovaném pokusu v další verzi aplikace je již orientaci uzamknuta na režim na výšku.

Při přechodu do mé aplikace z jiné aplikace, která byla v režimu na šířku, se aplikace zachová správně a stále se zobrazí na výšku, jak je očekáváno.

11.4 Připojení k internetu

Ve výchozím stavu má aplikace, s výjimkou informací o rozvrzích a zkouškách, plně fungovat bez internetu, jelikož jsou data obsažena již ve zveřejněné aplikaci.

Aplikaci jsem se pokoušel spustit s vypnutou Wi-Fi a vypnutými daty a aplikace se úspěšně spustila. Všechny funkcionality aplikace fungovaly tak, jak jsem očekával.

Mapa se zobrazila, vyhledávání místností fungovalo, informační obrazovky k budovám a místnostem zobrazovala správná data. Dle očekávání se nezobrazily informace o rozvrhu a zkouškách, protože nebylo k dispozici internetové připojení a tak nebylo možné odeslat požadavek do webových služeb STAG. Místo toho se zobrazí v dolní části notifikace, že není k dispozici internetové připojení a některé funkce nejsou k dispozici. Aplikace však stále fungovala jak má a nepřestala pracovat.

Při vypnutí Wi-Fi a mobilních dat v průběhu aplikace se aplikace zachovala taktéž správně. Vypne-li se internetové připojení, zatímco je uživatel již na obrazovce s rozvrhy, data zůstanou zobrazena, jelikož byla již stažena. Pokud z obrazovky odejdu nebo vypnu připojení na jiné obrazovce a následně se pokusím otevřít obrazovku s informacemi o místnosti, rozvrh nepůjde stáhnout a aplikace zobrazí informaci o nedostupnosti internetu.

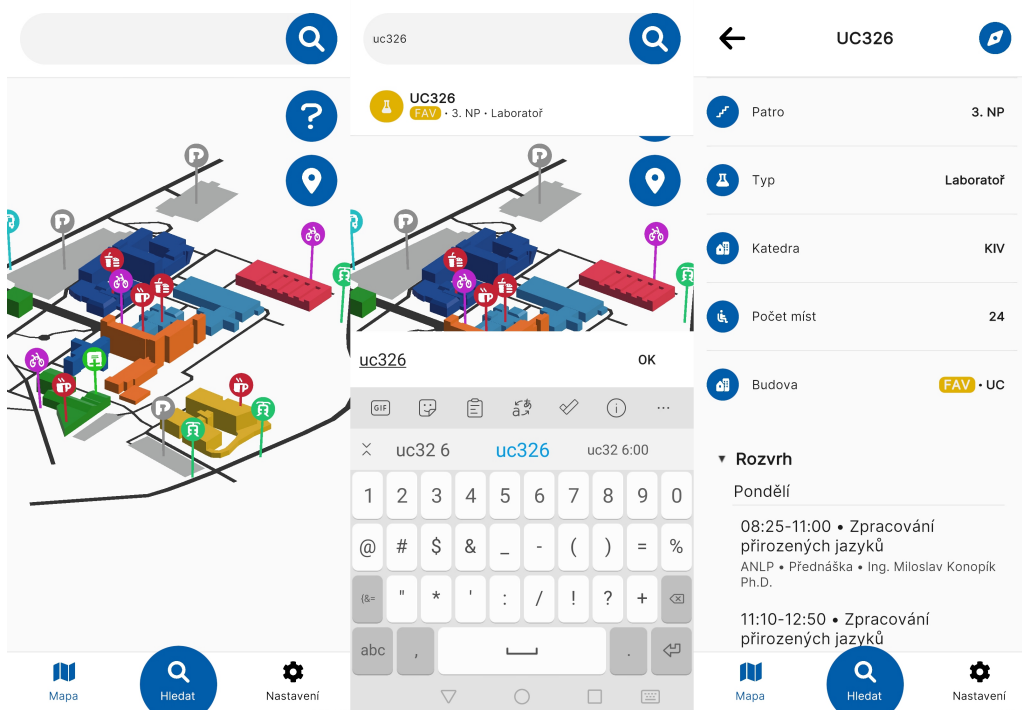
11.5 Různá zařízení

Android má řadu různých zařízení s různými verzemi Androidu a odlišnými velikostmi obrazovek. Proto jsem aplikaci testoval na několika různých zařízeních, abych ověřil, zda aplikace funguje tak, jak má.

11.5.1 Huawei nova 3

Primárně jsem aplikaci testoval na zařízení Huawei nova 3 s rozlišením 2340x1080 pixelů s operačním systémem Android 9.

Na tomto zařízení je aplikace plně funkční a veškerá funkcionalita se chová dle očekávání, viz snímky z obrazovky na obr. 11.1.

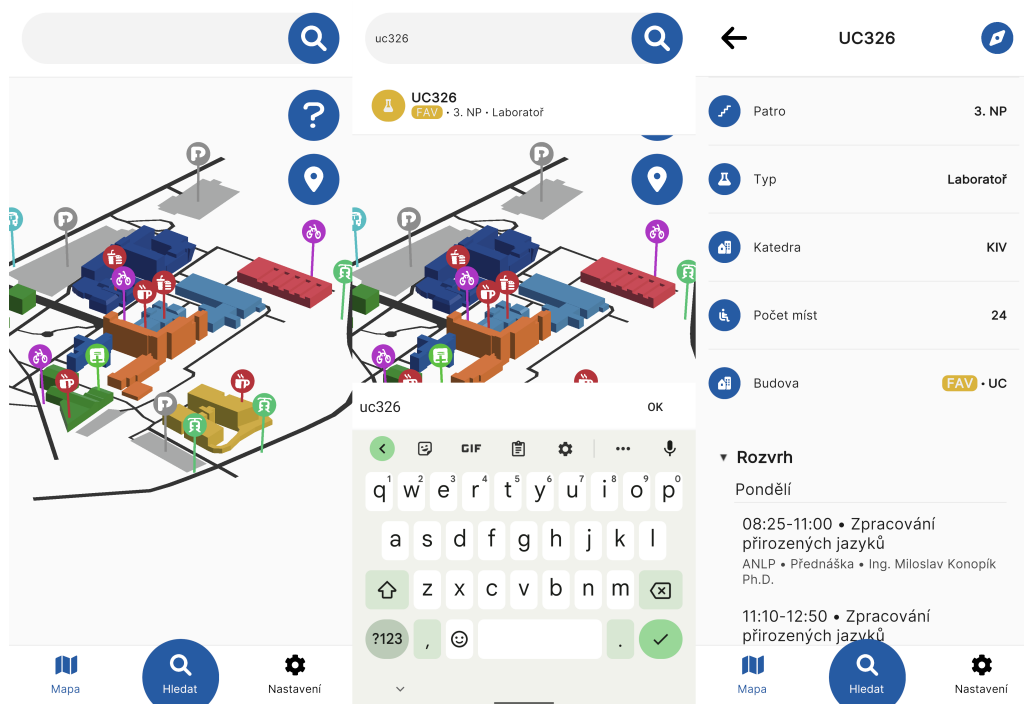


Obrázek 11.1: Snímky z obrazovky Samsung Galaxy S7

11.5.2 Google Pixel 6 Pro

Dalším zařízením, na kterém jsem aplikaci testoval je Google Pixel 6 Pro s rozlišením 3120 x 1440 pixelů a operačním systémem Android 12.

Aplikace se na zařízení chová dle očekávání, viz snímky z obrazovky na obr. 11.2.



Obrázek 11.2: Snímky obrazovky ze zařízení Google Pixel 6 Pro

11.5.3 Samsung Galaxy S7

Aplikaci jsem dále testoval na zařízení Samsung Galaxy S7 s rozlišením 1440x2560 pixelů a Androidem ve verzi 8.0.

Aplikace se vykreslila správně a veškerá funkcionalita aplikace fungovala dle očekávání, viz snímky z obrazovky na obr. 11.3.



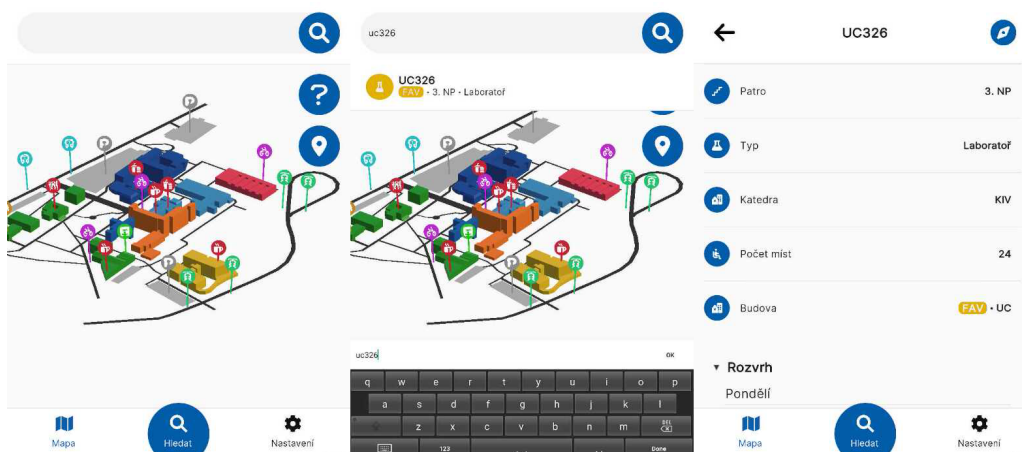
Obrázek 11.3: Snímky obrazovky ze zařízení Samsung Galaxy S7

11.5.4 Samsung Galaxy Tab S3 9.7

Otestoval jsem také extrémní případ, kdy by uživatel aplikaci mohl otevřít na tabletu, který má velkou obrazovku.

Tablet Samsung Galaxy S3 9.7 měl rozlišení 1536x2048 pixelů a verzi Androidu 9.0.

I zde se aplikace vykreslila správně a fungovala tak, jak má, viz snímky z obrazovky na obr. 11.4.



Obrázek 11.4: Snímky obrazovky ze zařízení Samsung Galaxy Tab S3

11.6 Shrnutí

Aplikaci jsem se snažil pečlivě testovat jak po funkční stránce tak i hardwarové stránce. Otestoval jsem hlavní funkcionality aplikace a obvyklé potíže, které mohou nastat jako nesprávné otáčení obrazovky či potíže s internetovým připojením.

Testování dopadlo kladně a až na menší potíže s klávesnicí (viz kapitola 9.3.1). Aplikace je dle mého názoru stabilní.

12 Závěr

Prozkoumal jsem aplikace různých univerzit, které se zabývají orientací v daném kampusu univerzity.

Na základě těchto poznatků jsem zvolil funkcionality, které by interaktivní mapa měla uživatelům nabízet. Zejména se jedná o rychlé hledání místností a poskytování informací o místnostech a bodech zájmu v univerzitním kampusu.

Navrhl jsem a realizoval aplikaci pro platformu Android, která má za cíl pomoci obzvláště studentům prvních ročníků, zahraničním studentům a univerzitním hostům orientovat se na poměrně rozsáhlém univerzitním kampusu v Plzni na Borech.

Funkcionalita vytvořené aplikace byla ověřena na celé řadě testovacích scénářů včetně běhu na zařízeních s různými verzemi Androidu a různými rozlišeními obrazovek.

V neposlední řadě jsem se zaměřil na možné budoucí rozšíření. Aplikace má dle mého názoru potenciál se dále rozvíjet a přidat se tak k již existujícím rozvrhovým aplikacím jako nepostradatelný pomocník při studiu na Západočeské univerzitě v Plzni.

Literatura

- [1] *Uses SDK element* [online]. www.android.com. [cit. 21. 6. 2022].
Dostupné z: <https://developer.android.com/guide/topics/manifest/uses-sdk-element>.
- [2] *Mobile Tablet Android Version Market Share Worldwide* [online].
gs.statcounter.com. [cit. 21. 6. 2022]. Aktualizováno 21. 6. 2022.
Dostupné z: <https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide/#yearly-2021-2021-bar>.
- [3] *What is an API?* [online]. www.redhat.com. [cit. 15. 6. 2022].
Aktualizováno 2. 6. 2022. Dostupné z: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>.
- [4] *Understanding What API is and How it works - not just its definition!*
[online]. www.studytonight.com. [cit. 15. 6. 2022]. Aktualizováno 16. 10.
2021. Dostupné z: <https://www.studytonight.com/post/understanding-what-api-is-and-how-it-works>.
- [5] *Introduction to Apple platform security* [online]. www.apple.com. [cit. 13. 6.
2022]. Aktualizováno 13. 5. 2022. Dostupné z:
<https://support.apple.com/en-gb/guide/security/seccd5016d31/web>.
- [6] *Berkeley Mobile App* [online]. www.berkeley.edu. [cit. 20. 6. 2022].
Dostupné z: <https://technology.berkeley.edu/mobile-app>.
- [7] *UC Berkeley Mobile* [online]. www.google.com. [cit. 20. 6. 2022].
Dostupné z: https://play.google.com/store/apps/details?id=edu.ucberkeley.com&hl=en_US.
- [8] *Blender Download Page* [online]. www.blender.org. [cit. 22. 6. 2022].
Dostupné z: <https://www.blender.org/download/>.
- [9] *Browser Market Share Czechia* [online]. gs.statcounter.com. [cit. 14. 6.
2022]. Aktualizováno 14. 6. 2022. Dostupné z:
<https://gs.statcounter.com/browser-market-share/all/czech-republic/#yearly-2021-2021-bar>.
- [10] *Browser Market Share Worldwide* [online]. gs.statcounter.com. [cit. 14. 6.
2022]. Aktualizováno 14. 6. 2022. Dostupné z: <https://gs.statcounter.com/browser-market-share#yearly-2021-2021-bar>.

- [11] *C Programming Language* [online]. www.geeksforgeeks.org. [cit. 13. 6. 2022]. Aktualizováno 20. 5. 2022. Dostupné z: <https://www.geeksforgeeks.org/csharp-programming-language/>.
- [12] *Reflexe (C#)* [online]. docs.microsoft.com. [cit. 14. 6. 2022]. Aktualizováno 10. 5. 2022. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/programming-guide/concepts/reflection>.
- [13] *What is Cross-Platform App Development?* [online]. www.fullscale.io. [cit. 19. 6. 2022]. Aktualizováno 23. 6. 2020. Dostupné z: <https://fullscale.io/blog/what-is-cross-platform-app-development/>.
- [14] *Repozitář softwaru Fluent NHibernate* [online]. www.github.com. [cit. 16. 6. 2022]. Dostupné z: <https://github.com/nhibernate/fluent-nhibernate>.
- [15] *A Brief History of Flutter* [online]. www.medium.com. [cit. 14. 6. 2022]. Aktualizováno 29. 1. 2021. Dostupné z: <https://medium.com/pragmatic-programmers/a-brief-history-of-flutter-939645f93255>.
- [16] *What is the Flutter Framework?* [online]. www.perfecto.io. [cit. 14. 6. 2022]. Aktualizováno 18. 5. 2021. Dostupné z: <https://www.perfecto.io/blog/what-is-flutter-framework>.
- [17] *Game Engines Aren't Just for Gaming Anymore* [online]. www.hackernoon.com. [cit. 14. 6. 2022]. Aktualizováno 20. 5. 2021. Dostupné z: <https://hackernoon.com/game-engines-arent-just-for-gaming-anymore-nb2a34k0>.
- [18] *Apple unveils an all-new Lock Screen experience and new ways to share and communicate in iOS 16* [online]. www.apple.com. [cit. 12. 6. 2022]. Aktualizováno 6. 6. 2022. Dostupné z: <https://www.apple.com/newsroom/2022/06/apple-unveils-new-ways-to-share-and-communicate-in-ios-16/>.
- [19] *List Of All Apple iOS Version History* [online]. www.gkgigs.com. [cit. 12. 6. 2022]. Aktualizováno 7. 6. 2022. Dostupné z: <https://www.gkgigs.com/list-apple-ios-version-history/>.
- [20] *Apple Support Page* [online]. www.apple.com. [cit. 12. 6. 2022]. Aktualizováno 9. 6. 2022. Dostupné z: <https://support.apple.com/en-in/HT201222>.
- [21] *Apple iOS Operating System: History, Origin, and More* [online]. www.history-computer.com. [cit. 12. 6. 2022]. Aktualizováno 1. 5. 2022. Dostupné z: <https://history-computer.com/apple-ios-operating-system-guide/>.

- [22] *JSON - Introduction* [online]. www.w3schools.com. [cit. 16. 6. 2022].
Dostupné z: https://www.w3schools.com/js/js_json_intro.asp.
- [23] *Mazemap* [online]. www.mazemap.com, Aktualizováno 19. 1. 2021. [cit. 16. 6. 2022]. Dostupné z: <https://www.mazemap.com/universities>.
- [24] *Mobile Operating System Market Share Czech Republic in 2021* [online].
gs.statcounter.com. [cit. 12. 6. 2022]. Aktualizováno 12. 6. 2022.
Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/czech-republic/#yearly-2021-2021-bar>.
- [25] *Mobile Operating System Market Share Worldwide in 2021* [online].
gs.statcounter.com. [cit. 12. 6. 2022]. Aktualizováno 12. 6. 2022.
Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/worldwide/#yearly-2021-2021-bar>.
- [26] *Mobile Operating System Market Share United States Of America in 2021*
[online]. gs.statcounter.com. [cit. 12. 6. 2022]. Aktualizováno 12. 6. 2022.
Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/united-states-of-america/#yearly-2021-2021-bar>.
- [27] *What Is MySQL? A Beginner-Friendly Explanation* [online].
www.hostinger.com. [cit. 16. 6. 2022]. Aktualizováno 11. 4. 2022.
Dostupné z: <https://kinsta.com/knowledgebase/what-is-mysql/>.
- [28] *Native Mobile App* [online]. www.techopedia.com. [cit. 13. 6. 2022].
Aktualizováno 5. 9. 2018. Dostupné z:
<https://www.techopedia.com/definition/27568/native-mobile-app>.
- [29] *Trying to using Nhibernate with Mono SQLite - can't find
System.Data.SQLite* [online]. www.stackoverflow.com. [cit. 22. 6. 2022].
Dostupné z: <https://stackoverflow.com/a/6203812/>.
- [30] *What is Object Relational Mapping?* [online]. www.educative.io. [cit. 16. 6.
2022]. Dostupné z: <https://www.educative.io/answers/what-is-object-relational-mapping>.
- [31] *NHibernate - Overview* [online]. www.tutorialspoint.com. [cit. 16. 6. 2022].
Dostupné z: https://www.tutorialspoint.com/nhibernate/nhibernate_overview.htm.
- [32] *NHibernate Homepage* [online]. www.nhibernate.info. [cit. 16. 6. 2022].
Dostupné z: <https://nhibernate.info/>.
- [33] *What is an ORM, how does it work, and how should I use one?* [online].
www.stackoverflow.com. [cit. 16. 6. 2022]. Aktualizováno 19. 2. 2018.
Dostupné z: <https://stackoverflow.com/a/1279678/>.

- [34] *PostgreSQL: The World's Most Advanced Open Source Relational Database* [online]. www.postgresql.org. [cit. 16. 6. 2022]. Dostupné z: <https://www.postgresql.org>.
- [35] *PostgreSQL Documentation: A Brief History of PostgreSQL* [online]. www.postgresql.org. [cit. 16. 6. 2022]. Dostupné z: <https://www.postgresql.org/docs/current/history.html>.
- [36] *Introduction to Postman for API Development* [online]. www.geeksforgeeks.org. [cit. 15. 6. 2022]. Aktualizováno 6. 8. 2021. Dostupné z: <https://www.geeksforgeeks.org/introduction-postman-api-development/>.
- [37] *What Is React Native?* [online]. www.oreilly.com. [cit. 19. 6. 2022]. Dostupné z: <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>.
- [38] *What is a REST API* [online]. www.redhat.com. [cit. 18. 6. 2022]. Dostupné z: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.
- [39] *Co je to REST API* [online]. www.parse-error.cz. [cit. 18. 6. 2022]. Dostupné z: <https://www.parse-error.cz/nodejs-tutorial/4/co-je-to-rest-api>.
- [40] *What is SOAP?* [online]. www.altexsoft.com. [cit. 18. 6. 2022]. Aktualizováno 20. 8. 2020. Dostupné z: <https://www.altexsoft.com/blog/engineering/what-is-soap-formats-protocols-message-structure-and-how-soap-is-different-from-rest/>.
- [41] *What is SQLite?* [online]. www.codecademy.com. [cit. 16. 6. 2022]. Dostupné z: <https://www.codecademy.com/article/what-is-sqlite>.
- [42] *Input problem when keyboard is opened on Android* [online]. forum.unity.com. [cit. 19. 6. 2022]. Dostupné z: <https://forum.unity.com/threads/input-problem-when-keyboard-is-opened-on-android.354864/>.
- [43] *Unity (game engine)* [online]. fandom.com. [cit. 18. 6. 2022]. Dostupné z: [https://ultimatepopculture.fandom.com/wiki/Unity_\(game_engine\)](https://ultimatepopculture.fandom.com/wiki/Unity_(game_engine)).
- [44] *Unity Documentation: PlayerPrefs* [online]. docs.unity3d.com. [cit. 16. 6. 2022]. Dostupné z: <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>.

- [45] *Where can I find a working libsqlite3.so library to run SQLite on Android 64-bit?* [online]. www.dev-qa.com. [cit. 21. 6. 2022]. Dostupné z: <https://dev-qa.com/1826943/where-can-find-working-libsqlite3-library-run-sqlite-android>.
- [46] *Github Repozitář: sqlite-unity-plugin* [online]. www.github.com. [cit. 21. 6. 2022]. Dostupné z: <https://github.com/rizasif/sqlite-unity-plugin/tree/master/Plugins>.
- [47] *Database (SQLite) Setup for Unity* [online]. www.unity.com. [cit. 21. 6. 2022]. Dostupné z: <https://answers.unity.com/questions/743400/database-sqlite-setup-for-unity.html>.
- [48] *Unity Download Page* [online]. www.unity3d.com. [cit. 14. 6. 2022]. Dostupné z: <https://unity3d.com/get-unity/download>.
- [49] *XML introduction* [online]. developer.mozilla.org. [cit. 16. 6. 2022]. Aktualizováno 15. 5. 2022. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction.
- [50] ALLEN JONES, A. F. *C# for Java Developers*. Microsoft Press, 2003. ISBN 0-7356-1779-1.
- [51] ASIF, R. *SQLite and Unity: How to do it right*. [online]. www.medium.com. [cit. 21. 6. 2022]. Aktualizováno 17. 7. 2018. Dostupné z: <https://medium.com/@rizasif92/sqlite-and-unity-how-to-do-it-right-31991712190>.
- [52] BOYETT, R. *What is MySQL: MySQL Explained For Beginners* [online]. www.hostinger.com. [cit. 16. 6. 2022]. Aktualizováno 13. 5. 2022. Dostupné z: <https://www.hostinger.com/tutorials/what-is-mysql>.
- [53] BUDZIŃSKI, M. *What Is React Native? Complex Guide for 2022* [online]. www.netguru.com. [cit. 19. 6. 2022]. Dostupné z: <https://www.netguru.com/glossary/react-native>.
- [54] CALLAHAM, J. *The history of Android: The evolution of the biggest mobile OS in the world* [online]. www.androidauthority.com. [cit. 12. 6. 2022]. Aktualizováno 25. 12. 2021. Dostupné z: <https://www.androidauthority.com/history-android-os-name-789433/>.
- [55] CLEMENT, J. *Percentage of mobile device website traffic worldwide from 1st quarter 2015 to 4th quarter 2021* [online]. www.statista.com. [cit. 18. 6. 2022]. Aktualizováno 18. 2. 2022. Dostupné z: <https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/>.

- [56] COSTELLO, S. *The History of iOS, from Version 1.0 to 16.0* [online].
www.lifewire.com. [cit. 12. 6. 2022]. Aktualizováno 19. 5. 2022. Dostupné z:
<https://www.lifewire.com/ios-versions-4147730>.
- [57] FREEMAN, J. *What is JSON? A better format for data exchange* [online].
www.infoworld.com. [cit. 16. 6. 2022]. Dostupné z:
<https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html>.
- [58] GAVIN, B. *What is Sketchup (and How Do I Use It)?* [online].
www.howtogeek.com. [cit. 15. 6. 2022]. Aktualizováno 4. 9. 2018.
Dostupné z: <https://www.howtogeek.com/364232/what-is-sketchup/>.
- [59] GRIFFITH, C. *What is Hybrid Mobile App Development: Hybrid vs Native vs Web* [online].
www.ionic.io. [cit. 19. 6. 2022]. Dostupné z: <https://ionic.io/resources/articles/what-is-hybrid-app-development>.
- [60] GUPTA, L. *What is REST* [online]. www.restfulapi.net. [cit. 18. 6. 2022].
Aktualizováno 7. 4. 2022. Dostupné z: <https://restfulapi.net>.
- [61] HAAS, J. K. *A History of the Unity Game Engine* [online].
www.semanticscholar.org. [cit. 14. 6. 2022]. Aktualizováno 20. 11. 2019.
Dostupné z: <https://www.semanticscholar.org/paper/A-History-of-the-Unity-Game-Engine-Haas/5e6b2255d5b7565d11e71e980b1ca141aeb3391d#paper-header>.
- [62] HAINES, R. *How to store data locally in an Android app* [online].
www.androidauthority.com. [cit. 16. 6. 2022]. Aktualizováno 4. 5. 2022.
Dostupné z: <https://www.androidauthority.com/how-to-store-data-locally-in-android-app-717190/>.
- [63] HARTMANN, T. *What Is Blender (Software)? – Simply Explained* [online].
www.all3dp.com. [cit. 19. 6. 2022]. Aktualizováno 21. 5. 2020. Dostupné z:
<https://all3dp.com/2/blender-simply-explained/>.
- [64] HOCKING, J. *Unity in Action: Multiplatform game development in C#*. 2. vydání, Manning Publications Co., 2018. ISBN 1-61729-496-9.
- [65] KARIM, E. *Apple Ecosystem Explained* [online]. www.techjourneyman.com.
[cit. 13. 6. 2022]. Aktualizováno 24. 4. 2022. Dostupné z:
<https://techjourneyman.com/blog/apple-ecosystem-explained/>.
- [66] KENNEDY, K. *What Is 3ds Max? – Simply Explained* [online].
www.all3dp.com. [cit. 15. 6. 2022]. Aktualizováno 22. 3. 2020. Dostupné z:
<https://all3dp.com/2/what-is-3ds-max-simply-explained/>.

- [67] KENTON, W. *Apple iOS* [online]. www.investopedia.com. [cit. 12. 6. 2022]. Aktualizováno 25. 10. 2021. Dostupné z: <https://www.investopedia.com/terms/a/apple-ios.asp>.
- [68] KOSEK, J. *Využití webových služeb a protokolu SOAP při komunikaci* [online]. www.kosek.cz. [cit. 18. 6. 2022]. Dostupné z: <https://www.kosek.cz/diplomka/html/websluzby.html>.
- [69] LACKO, L. *Android: Kompletní průvodce vývojáře*. 1. vydání, Computer Press Publications Co., 2017. ISBN 978-80-251-4875-4.
- [70] LIANG, M. *Understanding Object-Relational Mapping: Pros, Cons, and Types* [online]. www.altexsoft.com. [cit. 16. 6. 2022]. Aktualizováno 11. 3. 2021. Dostupné z: <https://www.altexsoft.com/blog/object-relational-mapping/>.
- [71] MANDALIYA, P. *What Is NHibernate?* [online]. www.thecodehubs.com. [cit. 16. 6. 2022]. Aktualizováno 9. 4. 2022. Dostupné z: <https://www.thecodehubs.com/what-is-nhibernate/>.
- [72] NTNU. *Případová studie na NTNU* [online]. www.mazemap.com. [cit. 16. 6. 2022]. Aktualizováno 15. 2. 2017. Dostupné z: <https://www.jottacloud.com/p/mazemap/e9dd2117a84a4d059a0391ef4a0aac74>.
- [73] PARENTELEMENT, L. *JSON .NET For Unity* [online]. assetstore.unity.com. [cit. 16. 6. 2022]. Aktualizováno 8. 11. 2017. Dostupné z: <https://assetstore.unity.com/packages/tools/input-management/json-net-for-unity-11347>.
- [74] PARSONS, A. *Introducing Unity 2021 LTS* [online]. blog.unity.com. [cit. 14. 6. 2022]. Aktualizováno 12. 4. 2022. Dostupné z: <https://blog.unity.com/news/introducing-unity-2021-lts>.
- [75] PETER LOSHIN, J. S. *Structured Query Language (SQL)* [online]. www.techtarget.com. [cit. 16. 6. 2022]. Dostupné z: <https://www.techtarget.com/searchdatamanagement/definition/SQL>.
- [76] PETER LOSHIN, M. G. D. L. *XML (Extensible Markup Language)* [online]. www.techtarget.com. [cit. 16. 6. 2022]. Dostupné z: <https://www.techtarget.com/whatis/definition/XML-Extensible-Markup-Language>.
- [77] PETERSON, R. *SQL Commands: DML, DDL, DCL, TCL, DQL with Query Example* [online]. www.guru99.com. [cit. 16. 6. 2022]. Aktualizováno 23. 4. 2022. Dostupné z: <https://www.guru99.com/sql-commands-dbms-query.html>.

- [78] PETTY, J. *What is 3ds Max What is it Used For?* [online]. www.conceptartempire.com. [cit. 15. 6. 2022]. Dostupné z: <https://conceptartempire.com/what-is-3ds-max/>.
- [79] SHARMA, S. *The Growth of Flutter Development— 3 Years After The Birth of Alpha* [online]. medium.flutterdevs.com. [cit. 14. 6. 2022]. Aktualizováno 16. 8. 2020. Dostupné z: <https://medium.flutterdevs.com/the-growth-of-flutter-development-3years-after-the-birth-of-alpha-78baee809dff>.
- [80] SHURMA, K. *What is PostgreSQL and why do enterprise developers and start-ups love it?* [online]. www.ubuntu.com. [cit. 16. 6. 2022]. Aktualizováno 19. 1. 2021. Dostupné z: <https://ubuntu.com/blog/what-is-postgresql>.
- [81] STEFANIDIS, M. *Exploring the Growth of Game Engines* [online]. www.roundhillinvestments.com. [cit. 14. 6. 2022]. Aktualizováno 28. 4. 2022. Dostupné z: <https://www.roundhillinvestments.com/research/metaverse/exploring-the-growth-of-game-engines>.
- [82] SUMMERSON, C. *What's the Latest Version of Android?* [online]. www.howtogeek.com. [cit. 12. 6. 2022]. Aktualizováno 18. 11. 2021. Dostupné z: <https://www.howtogeek.com/345250/whats-the-latest-version-of-android/>.
- [83] TATARU, D. *How to Design and Implement the Fluent Interface Pattern in C#* [online]. www.assist-software.net. [cit. 16. 6. 2022]. Aktualizováno 14. 2. 2019. Dostupné z: <https://assist-software.net/blog/how-design-and-implement-fluent-interface-pattern-c>.
- [84] WALKER, A. *SOAP Web Services Tutorial: What is SOAP Protocol?* [online]. www.guru99.com. [cit. 18. 6. 2022]. Aktualizováno 16. 4. 2022. Dostupné z: <https://www.guru99.com/soap-simple-object-access-protocol.html>.
- [85] WALKER, J. *What Is SQLite and Why Is It So Popular?* [online]. www.howtogeek.com. [cit. 16. 6. 2022]. Aktualizováno 4. 8. 2021. Dostupné z: <https://www.howtogeek.com/devops/what-is-sqlite-and-why-is-it-so-popular/>.

Přílohy

- Uživatelská příručka
- Instalační příručka
- Struktura přiloženého ZIP souboru

Uživatelská příručka

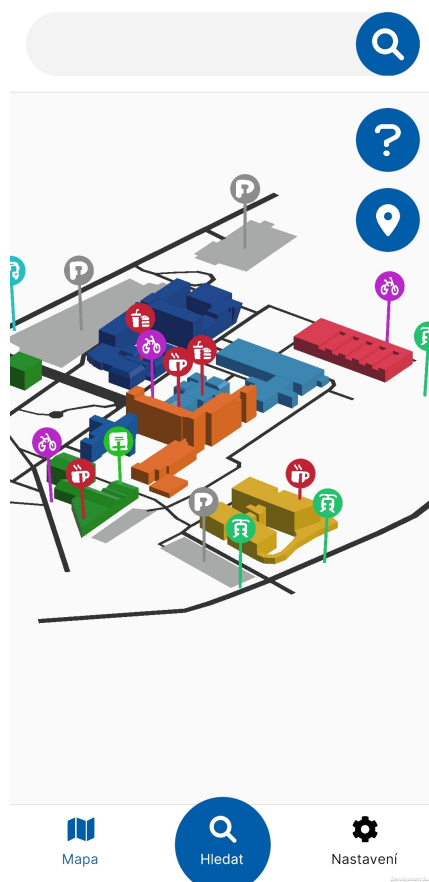
Aplikace nabízí užitečné funkcionality, které mají pomoci uživateli s orientací v kampusu a poskytování informací.

Mapa kampusu

Jádrem aplikace je interaktivní vizualizace plzeňského kampusu na Borech viz obr. 1. Zde je možné rotovat s mapou, která zobrazuje budovy kampusu a značky k bodům zájmů. Budovy lze rozkliknout pro zobrazení místností v dané budově.

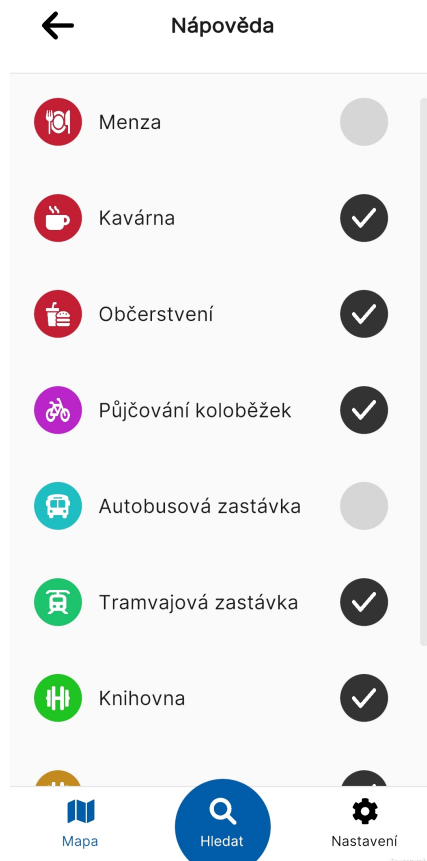
V horní části se nachází vyhledávač pro rychlé vyhledání konkrétní hledané místnosti. Pod vyhledávačem se nachází nápověda ke značkám na mapě a tlačítko pro zobrazení a skrytí značek.

V dolní části obrazovky se nachází navigace. Ta je součástí každé obrazovky aplikace. Obsahuje tlačítko pro zobrazení mapy, tlačítko pro rychlé hledání a tlačítko pro zobrazení nastavení aplikace.



Obrázek 1: Snímek interaktivní mapy

Po rozkliknutí tlačítka nápovědy (ikona s otazníkem v horním pravém rohu) se zobrazí další obrazovka s vysvětlením jednotlivých značek a možností si značky aktivovat či deaktivovat, viz obrázek 2.



Obrázek 2: Snímek obrazovky s nápovědou ke značkám

Kliknutím na druhé z tlačítek (ikona se značkou) se zviditelní či skryjí značky na mapě.

Aplikace si preference značek a jejich viditelnosti pamatuje, i v případě vypnutí a zapnutí aplikace.

Vyhledávání

Aplikace nabízí možnost rychlého hledání místností na základě jejich kódu, který se skládá z označení budovy a čísla místnosti, např. UC326.

Na obrázku 3 je vidět vyhledávací dialog aplikace. Výsledky obsahují název (kód) místnosti, fakultu či budovu do které místnost spadá, podlaží a typ místnosti. Výsledek se pro přehlednost barevně kóduje na základě fakulty, ve které se místnost nachází. Dále obsahuje také ikonu, které se mění v závislosti na typu místnosti.

Po rozkliknutí výsledku se otevře obrazovka s podrobnějšími informacemi o vybrané místnosti.



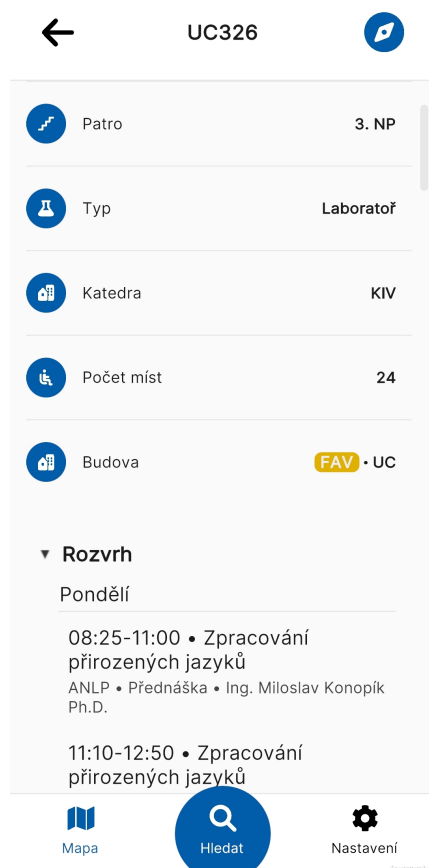
Obrázek 3: Snímek hledání místnosti

Informace o místnostech

Na této obrazovce jsou uživatelům zobrazeny podrobné informace o místnosti. Obrazovku lze zobrazit po rozkliknutí místnosti v seznamu místností v budově nebo po výběru místnosti ve vyhledávací.

Na obrázku 4 je vidět informační obrazovka o místnosti UC326. Je zde vidět patro ve kterém se místnost nachází, typ místnosti, katedra a budova, do které místnost patří a případně počet míst, je-li u místnosti určená kapacita.

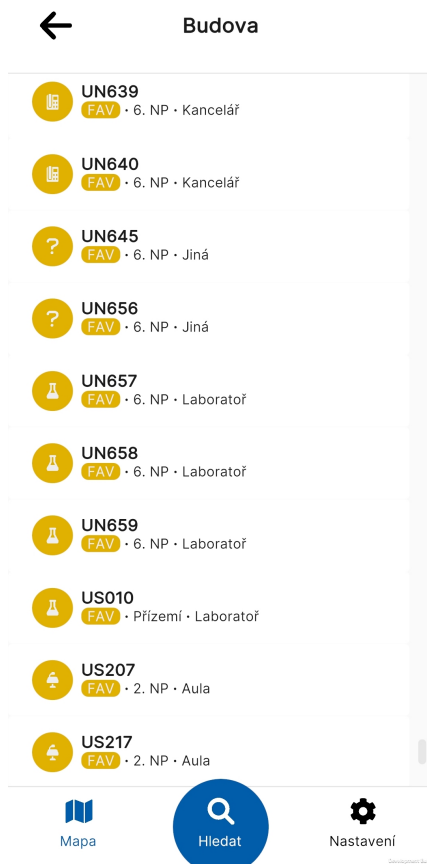
Pokud má uživatel aktivní připojení k internetu, zobrazí se také rozvrh místnosti a zkuškové termíny, které se v místnosti konají. Pokud nejsou tyto údaje o místnosti dostupné či připojení k internetu není k dispozici, zůstanou položky rozvrh a zkoušky prázdné.



Obrázek 4: Snímek informační obrazovky místnosti

Informace o budově

Budova na mapě je interaktivní. Po rozkliknutí se zobrazí obrazovka se seznamem místností v dané budově, viz obr. 5. Místnost je možné rozkliknout pro otevření obrazovky s informacemi o místnosti (viz předchozí kapitola).



Obrázek 5: Snímek obrazovky budovy

Instalační příručka

Proces instalace na mobilním zařízení je jednoduchý. Aplikaci lze nainstalovat buď z přiloženého APK souboru nebo lze vyexportovat aplikaci ručně a nainstalovat tuto verzi.

Požadavky

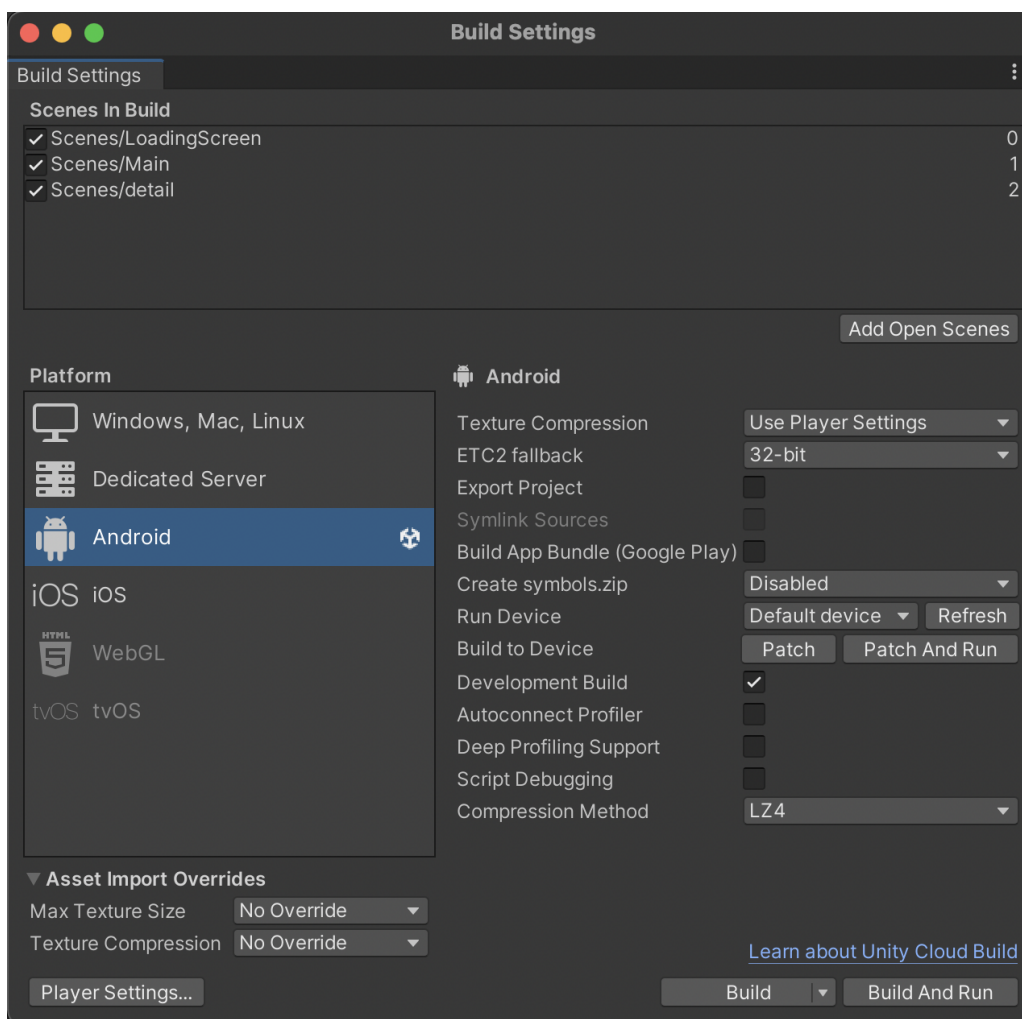
Pokud chceme vyexportovat aplikaci, je vyžadován počítač s operačním systémem Windows, Mac nebo Linux a nainstalované Unity ve verzi 2021.2.

Dále je potřeba mobilní zařízení se systémem Android v minimální verzi 5.1.

Export aplikace

Po otevření projektu v Unity je nejprve vhodné ujistit se, že je nastavená správná platforma pro export v nastavení exportu (viz obr. 6).

Nastavení exportu najdeme v záložce **File** → **Build Settings**. Stisknutím tlačítka **Build** získáme aplikaci zabalenou do souboru APK.



Obrázek 6: Nastavení exportu

Exportovat aplikaci ale není nutné, protože součástí přílohy je i APK soubor s již vyexportovanou verzí.

Instalace aplikace

Pro nainstalování aplikace musíme mít k dispozici APK soubor, který získáme buď z přílohy nebo si aplikaci vyexportujeme sami viz předchozí kapitola.

Soubor přetáhneme do mobilního zařízení, v prohlížeči souborů na mobilním zařízení soubor najdeme a kliknutím spustíme instalaci.

Struktura přiloženého ZIP souboru

Přiložený ZIP soubor obsahuje:

- Soubor README
- PDF verzi bakalářské práce
- Zdrojové kódy bakalářské práce
- Zdrojové kódy aplikace
- Vyexportovaná aplikace ve formátu APK